

Documented Code For glossaries v4.6

Nicola L.C. Talbot

Dickimaw Books

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

2025-04-14

Contents

1	Introduction	3
2	Main Package Code (glossaries.sty)	4
2.1	Package Definition	4
2.2	Package Options	7
2.3	Predefined Text	49
2.4	Xindy	59
2.5	Loops and conditionals	69
2.6	Defining new glossaries	77
2.7	Defining new entries	81
2.8	Resetting and unsetting entry flags	109
2.9	Keeping Track of How Many Times an Entry Has Been Unset . .	114
2.10	Loading files containing glossary entries	119
2.11	Using glossary entries in the text	120
	2.11.1 Links to glossary entries	125
	2.11.2 Displaying entry details without adding information to the glossary	172
2.12	Adding an entry to the glossary without generating text	183
2.13	Creating associated files	185
2.14	Writing information to associated files	207
2.15	Glossary Entry Cross-References	218
2.16	Displaying the glossary	220
2.17	Acronyms	265
2.18	Predefined acronym styles	270
2.19	Predefined Glossary Styles	288
2.20	Debugging Commands	289
2.21	Compatibility with version 2.07 and below	294

3	Glossary Styles	295
3.1	Glossary hyper-navigation definitions (<code>glossary-hypernav.sty</code>) . . .	295
3.2	In-line Style (<code>glossary-inline.sty</code>)	299
3.3	List Style (<code>glossary-list.sty</code>)	301
3.4	Glossary Styles using <code>longtable</code> (<code>glossary-long.sty</code>)	305
3.5	Glossary Styles using <code>longtable</code> and <code>booktabs</code> (<code>glossary-longbooktabs.sty</code>)	312
3.6	Glossary Styles using <code>longtable</code> (<code>glossary-longragged.sty</code>)	317
3.7	Glossary Styles using <code>multicol</code> (<code>glossary-mcols.sty</code>)	322
3.8	Glossary Styles using <code>supertabular</code> environment (<code>glossary-super.sty</code>)	330
3.9	Glossary Styles using <code>supertabular</code> environment (<code>glossary-superragged.sty</code>)	337
3.10	Tree Styles (<code>glossary-tree.sty</code>)	344
4	Multi-Lingual Support	415
4.1	Babel Captions (<code>glossaries-babel.sty</code>)	415
4.2	Polyglossia Captions (<code>glossaries-polyglossia.sty</code>)	416
5	Prefix Support (<code>glossaries-prefix Code</code>)	417
6	Accessibility Support (<code>glossaries-accsupp Code</code>)	424
6.1	Defining Replacement Text	426
6.2	Accessing Replacement Text	430
6.3	Displaying the Glossary	450
6.4	Acronyms	452
6.5	Debugging Commands	462
7	Backwards Compatibility	463
7.1	<code>glossaries-compatible-207</code>	463
7.2	<code>glossaries-compatible-307</code>	463
8	Rollback	464
8.1	Rollback v4.46 (<code>glossaries-2020-03-19.sty</code>)	464
8.2	Rollback v4.46 (<code>glossaries-accsupp-2020-03-19.sty</code>)	627
8.3	Rollback v4.46 (<code>glossaries-babel-2020-03-19.sty</code>)	662
8.4	Rollback v4.46 (<code>glossaries-compatible-207-2020-03-19.sty</code>)	663
8.5	Rollback v4.46 (<code>glossaries-compatible-307-2020-03-19.sty</code>)	668
8.6	Rollback v4.46 (<code>glossaries-polyglossia-2020-03-19.sty</code>)	679
8.7	Rollback v4.46 (<code>glossaries-prefix-2020-03-19.sty</code>)	679
8.8	Rollback v4.46 (<code>glossary-hypernav-2020-03-19.sty</code>)	684
8.9	Rollback v4.46 (<code>glossary-inline-2020-03-19.sty</code>)	685
8.10	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	686
8.11	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	688
8.12	Rollback v4.46 (<code>glossary-longbooktabs-2020-03-19.sty</code>)	691
8.13	Rollback v4.46 (<code>glossary-longragged-2020-03-19.sty</code>)	694
8.14	Rollback v4.46 (<code>glossary-mcols-2020-03-19.sty</code>)	697
8.15	Rollback v4.46 (<code>glossary-super-2020-03-19.sty</code>)	701
8.16	Rollback v4.46 (<code>glossary-superragged-2020-03-19.sty</code>)	705

8.17	Rollback v4.46 (glossary-tree-2020-03-19.sty)	708
8.18	Rollback v4.49 (glossaries-2021-11-01.sty)	714
8.19	Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)	880
8.20	Rollback v4.49 (glossaries-babel-2021-11-01.sty)	915
8.21	Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)	916
8.22	Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)	922
8.23	Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)	932
8.24	Rollback v4.49 (glossaries-prefix-2021-11-01.sty)	932
8.25	Rollback v4.49 (glossary-hypernav-2021-11-01.sty)	937
8.26	Rollback v4.49 (glossary-inline-2021-11-01.sty)	938
8.27	Rollback v4.49 (glossary-list-2021-11-01.sty)	940
8.28	Rollback v4.49 (glossary-long-2021-11-01.sty)	942
8.29	Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)	945
8.30	Rollback v4.49 (glossary-longragged-2021-11-01.sty)	948
8.31	Rollback v4.49 (glossary-mcols-2021-11-01.sty)	951
8.32	Rollback v4.49 (glossary-super-2021-11-01.sty)	954
8.33	Rollback v4.49 (glossary-superragged-2021-11-01.sty)	958
8.34	Rollback v4.49 (glossary-tree-2021-11-01.sty)	962
8.35	Rollback v4.52 (glossaries-2022-11-03.sty)	967
8.36	Rollback v4.52 (glossary-hypernav-2022-11-03.sty)	1121
9	Rollback	1122
9.1	Rollback v4.54 (glossaries-2024-04-03.sty)	1122
9.2	Rollback v4.58 (glossaries-2025-03-19.sty)	1276
9.3	Rollback v4.58 (glossary-mcols-2025-03-19.sty)	1442
9.4	Rollback v4.58 (glossary-tree-2025-03-19.sty)	1445

1 Introduction

This is the documented code for the `glossaries` package. This bundle comes with the following documentation:

`glossariesbegin.pdf` If you are a complete beginner, start with “The glossaries package: a guide for beginners”.

`glossary2glossaries.pdf` If you are moving over from the obsolete `glossary` package, read “Upgrading from the `glossary` package to the `glossaries` package”.

`glossaries-user.pdf` For the main user guide, read “`glossaries.sty` v4.6: L^AT_EX₂ε 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{v4.6}{2025-04-14}
```

Declare package:

```
\ProvidesPackage{glossaries}[2025/04/14 v4.6 (NLCT)]
```

Required packages:

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

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

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

```
\ExplSyntaxOn
```

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

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

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

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

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

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

Automatically identify exclusions, blockers and mappings.

`\glsmfuexcl`

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

`\glsmfublocker`

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

`\glsmfuaddmap`

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

```
}  
{
```

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

```
\RequirePackage{textcase}
```

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

`\MFUsentencecase`

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

Provide support for exclusions with `\MFUsentencecase`.

`\glsmfuexcl`

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

`\mfirstucMakeUppercase` We have an old version of `mfirstuc` that is still using this command in the definition of `\makefirstuc`, in which case, continue to use `\MakeTextUppercase` as in previous versions, in case there is also an old version of `textcase` installed. However, note that the all-caps commands like `\GLS` now use `\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. If you want some other way to start the glossary section (e.g. a numbered section) you will have to redefined `\glossarysection`.

```

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

```

Determine whether or not to use numbered sections.

`\@glossarysecstar`

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

```

`\@glossaryseclabel`

```
\newcommand*\@glossaryseclabel{}

```

`\glsautoprefix` Prefix to add before label if automatically generated:

```
\newcommand*\glsautoprefix{}

```

numberedsection

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

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

`\@glossary@default@style`

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

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

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

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]{}
\glsesclocationsfalse
```

`\@gls@loadlong`

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

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

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

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

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

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

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

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

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

`\@gls@loadlist`

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

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

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

```

    {\let\@glossary@default@style\relax}%
    }%
  }%
}

```

`\@gls@loadtree`

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

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

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

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

```

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

```

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

```

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

```

`nopostdot` Boolean option to suppress post description dot

```

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

```

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

```

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

```

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

```

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

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

```

```
{%
  \glsucmarkfalse
}
```

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

`\define@glossaryentrycounter`

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

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

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

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

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

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

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

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

`\define@glossarysubentrycounter`

```
\newcommand*{\@gls@define@glossarysubentrycounter}{%
Check if counter already defined.
  \ifundef\c@glossarysubentry
  {%
```

```

\ifglssubentrycounter
\ifglseentrycounter
\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

```

`\glssindexingsetting` 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{\glssindexingsetting}{\ifglsxindy xindy\else makeindex\fi}

```

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

```

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

```

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

```

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

```

```

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

```

`\glsprestandardsort`

Allow user to hook into sort mechanism. The first argument `<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_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@setupsort@standard
```

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

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

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

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

Store entry information when it's defined.

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

Defined count register associated with the glossary. 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_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_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[glg2]{main}{gls2}{glo2}{\glossaryname}%
  \else
    \newglossary{main}{gls}{glo}{\glossaryname}%
  \fi
}

```

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

```

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

```

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

`\glsdefaulttype`

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

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

`\acronymtype`

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

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

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

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

\newcommand*{\@glsacronymlists}{-}

```

`\@addtoacronymlists`

```

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

```

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

```

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

```

```

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

```

`\glsIfListOfAcronyms`

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

```

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

```

Internal command requires label and list to be expanded:

```

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

```

Not found

```
#4%
```

```
\else
```

Found

```
#3%
```

```
\fi
```

```
}
```

`\if@glsisacronymlist` Convenient boolean.

```
\newif\if@glsisacronymlist
```

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

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

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

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

`acronymlists`

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

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

`\glscounter`

```
\newcommand{\glscounter}{page}
```

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

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

`\@gls@nohyperlist`

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

`\GlsDeclareNoHyperList`

```
\newcommand*\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
```

```

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

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

\GlossariesWarning Prints a warning message.
\newcommand*\GlossariesWarning[1]{%
    \PackageWarning{glossaries}{#1}%
}

\GlossariesWarningNoLine Prints a warning message without the line number.
\newcommand*\GlossariesWarningNoLine[1]{%
    \PackageWarningNoLine{glossaries}{#1}%
}

\glosortentrieswarning Warn user that sorting may take a long time. This is actually an informational
message rather than a warning so just use \typeout.
\newcommand*\glosortentrieswarning{%
    \typeout{Using TeX to sort glossary entries---this may
    take a while}%
}

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

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

```


nolangwarn Suppress warning if language support not found.

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

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

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

\@gls@warnontheGLOSSdefined Issue a warning if overriding theGLOSSary

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

noredefwarn Suppress warning on redefinition of \printglossary

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

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

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

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

\glssetexpandfield

Sets field to always expand.

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

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

\glssetnoexpandfield

Sets field to never expand.

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

```

\gls@assign@type@field The type must always be expandable.
                        \glssetexpandfield{type}

\gls@assign@desc@field The description is not expanded by default:
                        \glssetnoexpandfield{desc}

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

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

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

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

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

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

                        Sanitizing stuff:

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

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

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

\@@gls@noidx@sanitizesort Remove braces around first character (if present) before sanitizing.
                        \newcommand*\@@gls@noidx@sanitizesort{%
                        \ifdefvoid\@glo@sort
                        {}%
                        {%

```

```

        \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
    }%
}
\def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
    \def\@glo@sort{#1#2}%
    \@onelevel@sanitize\@glo@sort
}

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

  \let\a@secondoftwo
  \def\AE{AE}%
  \def\ae{ae}%
  \def\OE{OE}%
  \def\oe{oe}%
  \def\AA{AA}%
  \def\aa{aa}%
  \def\L{L}%
}

```

```

\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%

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

```

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

```

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

```

`\@gls@removedoption`

```

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

```

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

```

\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\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
  \glSsetnoexpandfield { 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
    \glSsetnoexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glssanitizesorttrue
      \glSsetnoexpandfield{sortvalue}%
    }%
  \else
    \glSsetexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glssanitizesortfalse
      \glSsetexpandfield{sortvalue}%
    }%
  \fi
}

```

Default setting:

```

\glssanitizesorttrue
\glSsetnoexpandfield{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
depreciated. Use sanitizesort instead}%
}

```

sanitize

```

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

```

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

```
\newif\ifglstranslate
```

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

`\@glS@usetranslator`

```

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

```

```

        {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
        }%
        {}%
    }%
    {}%
}%
}

```

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

```

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

```

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

```

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

```

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

```

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

```



```

    }%
  \or
    \glstranslatefalse
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
  \or
    \glstranslatetrue
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
  \fi
}

```

Set the default value:

```

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

```

`\@gls@requested@languages`

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

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

```

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

```

locales Provide synonym.

```

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

```

```

        \let\glsifusetranslator\@secondoftwo
    }

indexonlyfirst Set whether to only index on first use.
    \define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
    \glsindexonlyfirstfalse

hyperfirst Set whether or not terms should have a hyperlink on first use.
    \define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
    \glshyperfirsttrue

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

footnote Set the long form of the acronym in footnote on first use.
    \define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
    \ifglsacrfootnote
        \@gls@removedoption{footnote}%
    \fi
}

description Allow acronyms to have a description (needs to be set using the description key
in the optional argument of \newacronym).
    \define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
    \ifglsacrdescription
        \@gls@removedoption{description}%
    \fi
}

smallcaps Define \newacronym to set the short form in small capitals.
    \define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
    \ifglsacrsmallcaps
        \@gls@removedoption{smallcaps}%
    \fi
}

smaller Define \newacronym to set the short form using \smaller which obviously needs
to be defined by loading the appropriate package.
    \define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
    \ifglsacrsmaller
        \@gls@removedoption{smaller}%
    \fi
}

dua Define \newacronym to always use the long forms (i.e. don't use acronyms)
    \define@boolkey{glossaries.sty}[glsacr]{dua}[true]{}
    \ifglsacrdua

```

```

    \@gls@removedoption{dua}%
  \fi
}

```

shortcuts Define acronym shortcuts.

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

```

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

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

```

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

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

```

order

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

```

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

```
\newif\ifglxindy

```

The default is `makeindex`:

```
\glxindyfalse

```

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

```
\@gls@declareoption{makeindex}{\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{\languagename}%

```

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

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

```

```

        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@\glsdefaulttype @language}}%
    }%
    {\letcs\@gls@langmod{\@xdy@#1@language}}%
}

```

Define key to set the language

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

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

```

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

```

Define a key to set the code page.

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

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

```

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

```

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

```

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

```

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

```

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

```

`\@domakeglossaries`

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

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

```

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

```

```

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

```

restoremakegloss Cancel the effect of disablemakegloss.

```

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

```

\@do@write@glslabels

```

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

```

\@@do@write@glslabels

```

\newcommand*\@@do@write@glslabels}{%
  \newwrite\@gls@labelsfile
  \immediate\openout\@gls@labelsfile=\jobname.glslabels
  \protected@edef\@gls@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{%
    {\protected@eappto\@gls@fulllist{,\@ignored@glossaries}}%
  }%
  \forallglsentries[\@gls@fulllist]{\@glsentry}%
}

```

```

{%
  \ifdefempty{\@glsentry}{}%
  {%
    \ifx\do@write@glslabelsname\@gobble
      \def\@gls@thisentryname{%
    \else
      \letcs{\@gls@thisentryname}{glo@\glsdetoklabel{\@glsentry}@name}%
      \@onelevel@sanitize\@gls@thisentryname
    \fi
    \immediate\write\@gls@labelsfile
      {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
  }%
}%
\immediate\closeout\@gls@labelsfile
}

```

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

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

`\@write@glslabelsname`

```

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

```

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

```

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

```

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

```

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

```

`\ifglsautomake`

```
\newif\ifglsautomake
```

`\gls@automake@nr`

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

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

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

```

`\gls@automake@option`

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

```

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

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

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

```
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
  \ifcase\gls@automake@nr\relax
    \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
case 0: true (same as delayed)
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{}%
  \or
case 1: false
  \glsautomakefalse
  \renewcommand{\gls@automake@makegloss}{}%
  \or
case 2: immediate
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{}%
  \or
case 3: makegloss
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{makeglossaries \@gls@run@output@dir{-d}}%
  \or
case 4: lite
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
  \or
case 5: delayed
  \def\gls@automake@nr{0}%
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{}%
  \fi
  \renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
  \ifglsautomake
```

```

        \let\gls@doautomake\gls@do@automake@err
    \else
        \renewcommand*\gls@doautomake-{}%
    \fi
}
\glsautomakefalse

```

`\gls@doautomake`

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Set default:

```

\gls@savewritesfalse
\let\gls@writefiles\@empty

```

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

```

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

```



```

\ifbool{glscompatible-3.07}%
{%
  \@gls@removedoption{compatible-3.07}%
}%
{}%
}
\boolfalse{glscompatible-3.07}

```

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

```

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

\ifbool{glscompatible-2.07}%
{%
  \@gls@removedoption{compatible-2.07}%
}%
{}%
}
\boolfalse{glscompatible-2.07}

```

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

```
\let\gls@original@makeglossary\makeglossary
```

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

```
\let\gls@original@glossary\glossary
```

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

`\@gls@override@glossary`

```

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

```

In v4.10, the redefinition of `\glossary` was removed since it was never intended as a user level command (and wasn't documented in the user manual), however it seems there are packages that have hacked the internal macros used by `glossaries` and no longer work with this redefinition removed, so it's been restored in v4.11 but is not used at all by `glossaries`. (This may be removed or

moved to a compatibility mode in future.) As from v4.41, the use of `\glossary` now triggers a warning. The package option `kernelglossredefs=nowarn` may be used to remove the warning, but it's better not to use `\glossary`.

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

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

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

```
\define@choicekey{glossaries.sty}{kernelglossredefs}%
  [\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
  {%
    \ifcase\gls@kernelglossredefs@nr\relax
      \def\glossary{\@gls@override@glossary}%
      \def\makeglossary{%
        \GlossariesWarning{Use of \string\makeglossary\space with
          glossaries.sty is deprecated. Use \string\makeglossaries\space
          instead. If you need the original definition of
          \string\makeglossary\space use the package options
          kernelglossredefs=false (to prevent redefinition of
          \string\makeglossary) and nomain (if the file extensions cause a
          conflict)}%
        \makeglossaries
      }%
    \or
      \let\glossary\gls@original@glossary
      \let\makeglossary\gls@original@makeglossary
    \or
      \def\makeglossary{\makeglossaries}%
      \renewcommand*{\@gls@override@glossary}[1][main]{%
        \gls@glossary{##1}%
      }%
    \fi
  }
```

`symbols` Create a “symbols” glossary type

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

Default is not to define the symbols glossary:

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

```
\@gls@symbolsdef
```

```

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

```

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

```

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

```

numbers Create a “symbols” glossary type

```

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

```

Default is not to define the numbers glossary:

```

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

```

`\@gls@numbersdef`

```

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

```

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

```

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

```

index Create an “index” glossary type

```

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

```

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

```

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

```

Default is not to define index glossary:

```

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

```

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

```

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

```

```

        {type={index},name={##2},description={\nopostdesc},##1}}
    \let\@gls@do@indexdef\relax
}%

```

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

```

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

```

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

```

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

```

Now process options passed to the package:

```
\ProcessOptionsX
```

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

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

This should expand to nothing if no output directory or to $\langle switch \rangle \langle dir \rangle$ otherwise, where $\langle switch \rangle$ is the application's switch that specifies the output directory and $\langle dir \rangle$ 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{\langle filename \rangle}
```

This should expand to $\langle filename \rangle$ or $\langle dir \rangle \langle sep \rangle \langle filename \rangle$ where $\langle dir \rangle$ is the output directory and $\langle sep \rangle$ is the directory separator. This is provided for direct calls to `makeindex` or `xindy`.

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

```
\ifglsautomake
```

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

```
\newcommand*\@gls@loadshellesc{\ifdef\directlua{\RequirePackage{shellesc}}{}}%
```

```
\ExplSyntaxOn
```

```
\sys_if_shell:TF
```

```
{
```

```
  \sys_if_shell_unrestricted:F
```

```
{
```

```
  \cs_set:Npn \@gls@run@unrestricted@shell #1
```

```
{
```

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

```
\GlossariesWarningNoLine
```

```
{
```

```
  Unrestricted ~ shell ~ required ~ for ~ current ~  
  automake ~ setting. ~
```

```
  Either ~ remove ~ automake ~ option ~ and ~ run ~ the ~  
  system ~ command ~ outside ~ of ~ TeX ~ or ~ enable ~  
  unrestricted ~ mode ~ (but ~ be ~ aware ~ of ~ the ~  
  security ~ risks ~ in ~ doing ~ so)
```

```
}
```

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

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

```
}
```

```
}
```

Try to determine if the output directory has been set. This requires at least TeX Live 2024 and is only applicable for `makeglossaries` and `bib2gls`. Unsupported

with `makeglossaries-lite` and for direct calls to `makeindex` and `xindy`. In the case of `makeindex`, the directory can be included in the reference for the input, output and style files, but the OS directory divider needs to be determined. This doesn't seem to work for `xindy` as it can't find the style module.

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

```

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

```

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

```

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

```

```

    }
    \int_if_zero:nTF { \gls@automake@nr }
    {
Delayed write
    \cs_set:Npn \@gls@runshell #1 { \exp_args:Ne \write18 { #1 } }
    }
    {
Immediate write
    \cs_set:Npn \@gls@runshell #1 { \sys_shell_now:e { #1 } }
    \cs_set_eq:NN \@gls@loadshellesc \relax
    }
    }
    {
    \cs_set_eq:NN \@gls@loadshellesc \relax
    \legacy_if_set_false:n { glsautomake }
    \PackageWarning { glossaries }
    {
    Can't ~ implement ~ automake: ~ no ~ shell
    }
    }
    }
\ExplSyntaxOff
\@gls@loadshellesc
\fi

```

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

```

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

```

Now define `\setupglossaries`:

```

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

```

```

        \let\@gls@doentrycounterdef\relax
    \else
        \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
    \fi
\ifgls@subentrycounter
    \let\@gls@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}

```

`\glssettoctitle` Sets the TOC title for the given glossary.

```

\newcommand*\glssettoctitle}[1]{%
\def\glossarytoctitle{\csname @gls@#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.

```
\ifvoid\@gls@requested@languages{%  
  {%  
    \for\@gls@thislang:=\@gls@requested@languages\do{%  
      \TrackIfKnownLanguage{\@gls@thislang}{%  
        {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%  
      }%  
    }%  
  }%
```

Load translator if required.

```
\@gls@usetranslator
```

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

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

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

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

Check for tracked languages

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

if using translator use translator interface.

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

Just in case `\istfilename` has been modified.

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

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

```
\@onlypremakeg\setStyleFile
```

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

`\istfilename`

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

`\gl@istfilebase`

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

The `makeglossaries` Perl script picks up this name from the auxiliary file. If the name ends with `.xdy` it calls `xindy` otherwise it calls `makeindex`. Since its not required by L^AT_EX, `\@istfilename` ignores its argument.

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

```

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

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

```

```
\glsSetCompositor Sets the compositor.
  \newcommand*\glsSetCompositor}[1]{%
    \renewcommand*\glscompositor}{#1}}
Only use before \makeglossaries
\@onlypremakeg\glsSetCompositor

```

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

```
\@glsAlphacompositor This is only used by xindy. It specifies the compositor to use when loca-
tion numbers are in the form <letter><compositor><number>. For example, if
\@glsAlphacompositor is set to “.” then it allows locations such as A.1 whereas
if \@glsAlphacompositor is set to “-” then it allows locations such as A-1.
  \newcommand*\@glsAlphacompositor}{\glscompositor}

```

```
\glsSetAlphaCompositor Sets the alpha compositor.
  \ifglsxindy
    \newcommand*\glsSetAlphaCompositor[1]{%
      \renewcommand*\@glsAlphacompositor}{#1}}
  \else
    \newcommand*\glsSetAlphaCompositor[1]{%
      \glsnoxindywarning\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[<type>]{<text>}
```

```
\setglossarypreamble
```

Code provided by Michael Pock.

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

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

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

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

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

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



```

    {%
      \@glossarysection{#1}{#2}%
    }%
    {%
      \@p@glossarysection{#1}{#2}%
    }%

    \glsglossarymark{\glossarytoctitle}%
  }

```

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

```

\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#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. If you don't want any sectional command, you will need to redefine `\glossarysection`. The sectional unit can be changed, if different sectional units are required.

`\setglossarysection`

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

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

`\@glossarysection`

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

Do automatic labelling if required

```
\@glossaryseclabel
}
```

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

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

```

\newcommand*\glsclearpage}{\gls@doclearpage}

```

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

```

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

```

2.4 Xindy

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

\ifglsxindy
  \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.

```

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

```

Check if already in list before adding.

```

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

```

Only has an effect before `\writeist`:

```

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

```

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

```

\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
    can't be used after \string\GlsAddXdyAttribute}{Move all
    occurrences of \string\GlsAddXdyCounters\space before the first
    instance of \string\GlsAddXdyAttribute}%
}

```

`\GlsAddXdyAttribute` Adds an attribute.

```
\ifglsxindy
```

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

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

Add to xindy attribute list

```

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

```

Add to xindy markup location.

```

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

```

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

```

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

```

High-level command:

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

Add to comma-separated attribute list

```

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

```

First add the attribute without the counter name:

```

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

```

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

```

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

```

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

```

\let\GlsAddXdyCounters\@disabled@glsaddxdycounters
}

```

Only has an effect before `\writeist`:

```

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

```

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

```

\ifglsxindy
\newcommand*\@gls@addpredefinedattributes{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
}

```

```

\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}

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

```

`\@xdyuseralphabets` List of additional alphabets

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

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

```

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

```

This code is only required for xindy:

```
\ifglxindy
```

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

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```



```

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

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

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

    End of xindy-only block:
\fi

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

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

\GlsSetXdyStyles Reset the list of required styles
    \ifglxindy
        \newcommand*\GlsSetXdyStyles[1]{%
            \edef\@xdyrequiredstyles{#1}}
    \else
        \newcommand*\GlsSetXdyStyles[1]{%
            \glsnoxywarning\GlsSetXdyStyles}
    \fi

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

\@xdylanguage The xindy language setting is required by makeglossaries, so provide a com-
mand for makeglossaries to pick up the information from the auxiliary file.
This command is not needed by the glossaries package, so define it to ignore its
arguments.
    \def\@xdylanguage#1#2{}

\GlsSetXdyLanguage Define a command that allows the user to set the language for a given glossary
type. The first argument indicates the glossary type. If omitted the main glos-
sary is assumed. This uses the unstarred form of \ifglossaryexists because
ignored glossaries can't be used with xindy.
    \ifglxindy
        \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
            \ifglossaryexists{#1}{%
                \expandafter\def\csname @xdy@#1@language\endcsname{#2}}%
            }{%
                \PackageError{glossaries}{Can't set language type for
                    glossary type '#1' --- no such glossary}{%
                    You have specified a glossary type that doesn't exist}}
    \fi

```

```

\else
  \newcommand*\GlsSetXdyLanguage[2][1]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi

```

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

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

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

```

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

```

Suggested by egreg:

```

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

```

`\@xdylettergroups` Store letter group definitions.

```

\ifglxindy
  \ifglxindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\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

```

`\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}%
}
```

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

```
\forallsentries[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*\forallsentries}[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@}%
  {%
    \forallsentries[\@this@glo@]{#2}{#3}%
  }%
}
```

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

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

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

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

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

```
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#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]{%
  \ifglentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{-}{#2}%
  }

```

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

`\glsdoifexistsorwarn`

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

```

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

```

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

`\glsdoifexistsordo`

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

```

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

```

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

`\doifglossarynoexistsordo`

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

```

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

```



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

\ifglshaschildren

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

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

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

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

\ifglshasparent

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

```
\ifglshasdesc \ifglshasdesc{<label>}{<true part>}{<>false part>}
\newcommand*\ifglshasdesc[3]{%
  \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
```

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

```

\newcommand*{\ifglstdescsuppressed}[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}%
    {%
      \ifdequal\@glo@short\@gls@default@value
        {#3}%
      {#2}%
    }%
  }
}

```

```

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

```

\ifgl\$fieldvoid

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

```

\newcommand{\ifgl$fieldvoid}[4]{\ifcvoid{glo@glsdetoklabel{#2}@#1}{#3}{#4}}

```

```

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

```

\ifgl\$hasfield

```

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

```

First check supplied field label is defined.

```

\ifdef\@glo@thisvalue
  {%

```

Is defined, so now check if empty.

```

\ifdefempty\@glo@thisvalue
  {%

```

Is empty, so doesn't have field set.

```

#4%
}%
{%

```

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

```

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

```

Value is set to the default value.

```

#4%
}%
{%

```

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

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

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

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

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

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

Is defined, so now check if empty.

```
\letcs{\@glo@thisvalue}{glo@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`.

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

Value is set to the default value.

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

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

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

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

```

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

```

`\gls@gobbleopt` Discard optional argument.

```

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

```

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

```

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

```

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

`\newglossary`

```

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

```

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

```

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

```

`\ns@newglossary` Define the unstarred version.

```

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

```

Check if default has been set

```

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

```

Add this to the list of glossary types:

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

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

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

Store the file extensions:

```
\expandafter\edef\csname @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 @glstype@#1@counter\endcsname{#2}%
```

Add counter to xindy list, if not already added:

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

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

```
\@gls@getcounter
```

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

```

`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}}

```

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

```

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

```

```

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

```

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

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

```

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

```

`\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
```

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

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

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

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

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

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

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

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

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

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

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

```

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

```

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

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

This entry has already been defined in the document environment.

```

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

```

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

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

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

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

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

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

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

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

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

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

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

```
\newcommand*{\gls@begindocdefs}{%
\@gls@enablesavenonumberlist
\edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{\@gls@restoreat}{\@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{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\protected@csxdef{glo@\glo@label @#2}{#3}%
}%
{%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}

```

\glsfielddedef

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

```

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

```

\glsfieldgdef

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

```

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

```

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

\glsfielddef

```

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

\glsfieldfetch

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.

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

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

`\ifglsfieldef`

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

```
\newrobustcmd{\ifglsfieldef}[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}{}%
}%
}

```

\gls.writedefhook

```

\newcommand*{\gls.writedefhook}{}

```

\gls@assign@desc

```

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

```

\longnewglossaryentry

```

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

```

```

        \gls@defglossaryentry{#1}{#2}%
    \egroup
}
}

```

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

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

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

```

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

```

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

`\gls@defglossaryentry`

Defines a new entry without checking if it already exists.

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

Prevent any further use of `\GlsSetQuote`:

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

Store label

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Set the default sort:

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

Set the default counter:

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

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

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

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

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

```
\@gls@initnonnumberlist
```

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

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

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

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

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

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

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

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

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

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

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

```
\@newglossaryentryprehook
```

Extract key-val information from third parameter:

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

Check there is a default glossary.

```
\ifundef\glsdefaulttype
```

```
{%
```

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

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

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

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

```
}%
```

```
{ }%
```

Assign type. This must be fully expandable

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

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

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

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

```
{%
```

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

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

```

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

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

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

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

```

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

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

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

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

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

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

Set the level for this entry

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

Define commands associated with this entry:

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

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

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



```

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

\ifcsundef{@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@see}%
```

```

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

\@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@glseentries[\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{gls@glsdetoklabel{##1}@currcount}}%
{0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
\ifcsundef{gls@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{gls@glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*\@glslocalreset}[1]{%
\@glslocalreset{##1}%
\ifglsresetcurrcount\csdef{gls@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\cglsp1@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cglsp1format{##2}{##3}%
    \glunset{##2}%
  \else
    \@glsp1@{##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@glstoklabel{#1}@currcount}{%
    \number\numexpr\glsentrycurrcount{#1}+1}%
  }

```

`\@gls@write@entrycounts` Write the entry counts to the aux file. Use `\immediate` since this occurs right at the end of the document. Only write information for entries that have been used. (Some users have a file containing vast numbers of entries, many of which may not be used. There's no point writing information about the entries that haven't been used and it will only slow things down.)

```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*{\string@gls@entry@count}[2]{}}%
  \forallglsentries{\@glsentry}{%
    \ifglsused{\@glsentry}%
    {\immediate\write\@auxout
      {\string@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
    }%
  }

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

\newcommand*{\cglformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
  }

```

`\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{\cGls}{\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][{}]{%
  \ifglsashaslong{#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}{\glstentrylongpl{#1}}{\glstentryfirstplural{#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}{\Glstentrylongpl{#1}}{\Glstentryfirstplural{#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

```
\glscaps case  
{%
```

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  
}%  
}
```

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

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

Subsequent use:

```
\glsifplural  
{%
```

Subsequent plural form:

```
\glsupercase  
{%
```

Subsequent plural form, don't adjust case:

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

Subsequent plural form, make first letter upper case:

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

Subsequent plural form, all caps:

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

Subsequent singular form

```
\glsupercase  
{%
```

Subsequent singular form, don't adjust case:

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

Subsequent singular form, make first letter upper case:

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

Subsequent singular form, all caps:

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

First use:

```
\glsifplural  
{%
```

First use plural form:

```
\glsupercase  
{%
```

First use plural form, don't adjust case:

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

First use plural form, make first letter upper case:

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

First use plural form, all caps:

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

First use singular form

```
\glscapscase  
{%
```

First use singular form, don't adjust case:

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

First use singular form, make first letter upper case:

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

First use singular form, all caps:

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

User supplied text.

```
\glscustomtext  
}%  
}
```

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

\genacrfullformat

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

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

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

`\Genacrfullformat`

```
\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\@gls@link
}
```

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

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

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

```
  \glstextformat{#3}%
  }%

  \glspostlinkhook
}
```

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

```
\newrobustcmd*\Glslink{%
  \@glshyp@opt\@Gls@link
}
```



```

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

\glspostlinkhook
\newcommand*{\glspostlinkhook}{}

\@gls@link@checkfirsthyper Check for first use and switch off hyper key if hyperlink not wanted. (Should be
off if first use and hyper=false is on or if first use and both the entry is in an
acronym list and the acrfootnote setting is on.) This assumes the glossary type
is stored in \glsstype and the label is stored in \glslabel.
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glsstype
    \ifglsyperfirst
      \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@glsdisablehyperinlist Disable hyperlink if in the “nohyper” list.
\newcommand*{\do@glsdisablehyperinlist}{%
  \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
\@gls@restore@glslocal
}
```

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

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

`\glolinkprefix`

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

`\glsentrycounter` Set default value of entry counter

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

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

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

Are we using equation counter?

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

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

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

Check if hyperref version of this counter

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

```

    }%
  }%
  {%

```

Not using equation counter so no special measures:

```

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

```

Check if hyperref version of this counter

```

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

```

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

```

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

```

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

```

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

```

```

\if#1)\relax
  \def\@glo@prefix{)}%
  \if\relax#2\relax
    \def\@glo@suffi{x{glsnumberformat}}%
  \else
    \def\@glo@suffi{x{#2}}%
  \fi
\else
  \def\@glo@prefix{}\def\@glo@suffi{x{#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@dostubst{%
      \noexpand\DTLsubstituteall\noexpand\gls@xdystring
      {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
    }%
    \gls@dostubst
  }%
\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
\@gls@checkescactual
}

```

\@gls@checkescbar Similarly for \|:

```

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

```

\@gls@checkesclevel Similarly for \!:


```

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

```

\@gls@checkbar and for |:

```

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

```

\@gls@checklevel and for !:

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

\@gls@checkactual and for ?:

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

\@gls@checkesc and for \:

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

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

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

```

    \fi
    \@gls@xdycheckquote
}

```

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

```

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

```

Now go ahead and define `\@gls@xdycheckbackslash`

```

\def@gls@xdycheckbackslash

```

`\glsdohypertarget`

```

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

```

```

        \glsmeasureheight{\gls@tmplen}{#2}%
        \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\@gls@target\glsdohypertarget
}
```

Provide some convenience commands if not already defined:

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

Syntax:

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

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

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

First determine which version is being used:

`\gls`

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

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

`\@gls`

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

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

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

    \let\glsifplural\@secondoftwo
    \let\gls@scaps@case\@firstofthree
    \let\gls@customtext\@empty
  }
  \def\glsinsert{#3}%
}
```

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

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

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

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

Indicate that this entry has now been used

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

\glspostlinkhook
}
```

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

`\Gls`

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

Add sentence case mapping.

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

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

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

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

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

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

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

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

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

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

Indicate that this entry has now been used

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

```
\glspostlinkhook
}
```

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

\glspl

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

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

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

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

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

    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%

```

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

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

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

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

Indicate that this entry has now been used

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

```
\glspostlinkhook
}
```

`\Glspl` behaves in the same way as `\glspl`, except that the first letter of the link text is converted to uppercase (as with `\Gls`, if the first letter has an accent, it will need to be grouped).

`\Glspl`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glspl}{\Glspl}
```

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

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

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

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

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

Determine what the link text should be (this is stored in `\@glo@text`). This needs to be expanded so that the `\@glo@text` can be passed to `\xmakefirstuc`. Note that `\@gls@link` sets `\glstype`.

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

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

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

Indicate that this entry has now been used

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

\glspostlinkhook
}
```

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

`\GLSp1`

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

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

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

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

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

    \let\glsifplural\@firstoftwo
    \let\glsapspace\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
```

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

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

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

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

Indicate that this entry has now been used

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

```
\glspostlinkhook
}
```

`\glsdisp` `\glsdisp[options]{label}{text}` This is like `\gls` except that the link text is provided. This differs from `\glslink` in that it uses `\glsdisplay` or `\glsdisplayfirst` and unsets the first use flag.

First determine if we are using the starred form:

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

Defined the un-starred form.

```
\@glsdisp
```

```
\newcommand*{\@glsdisp}[3] []{%
\glsdoifexists{#2}{%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
\let\glsifplural\@secondoftwo
```

```
\let\glsifscapscase\@firstofthree
```

```
\def\glscustomtext{#3}%
```

```
\def\glsinsert{}%
```

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

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

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

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

Indicate that this entry has now been used

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

```
\glspostlinkhook
}
```

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

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

`\@Glsdisp`

```
\newcommand*{\@Glsdisp}[3] [] {%
  \@glsdisp[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glsdisp}{\@Glsdisp}
```

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

```
\newcommand*{\@gls@link@nocheckfirsthyper}{}
```

```
\@gls@field@link{<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
}
```

`\gls`text behaves like `\gls` except it always uses the value given by the text key and it doesn't mark the entry as used.

`\gls`text

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

Read in the final optional argument:

```
\def\@gls@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
```

`\GLStext` behaves like `\gls`text except the text is converted to uppercase.

`\GLStext`

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

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

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

Read in the final optional argument:

```
\def\@GLStext@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrytext{#2}#3}}%
}
```

`\GLstext` behaves like `\glsstext` except that the first letter of the text is converted to uppercase.

`\GLstext`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsstext}{\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}{\@gls@entryname{#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}{\@gls@supercase{\@gls@entryname{#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}{\@gls@entrydesc{#2}#3}%
}
```

`\Glsdesc` behaves like `\glsdesc` except that the first letter is converted to uppercase.

`\Glsdesc`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsdesc}{\Glsdesc}
```

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

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

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
```

\GLSdesc behaves like \glsdesc except that the link text is converted to uppercase.

\GLSdesc

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

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

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

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydesc{#2}#3}}%
}
```

\glsdescplural behaves like \gls except it always uses the value given by the descriptionplural key and it doesn't mark the entry as used.

\glsdescplural

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

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

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

Read in the final optional argument:

```
\def\@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
```

\Glsdescplural behaves like \glsdescplural except that the first letter is converted to uppercase.

\Glsdescplural

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
```

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

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

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
```

`\GLSdescplural` behaves like `\glsdescplural` except that the link text is converted to uppercase.

`\GLSdescplural`

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

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

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

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentrydescplural{#2}#3}%
}
```

`\glsymbol` behaves like `\gls` except it always uses the value given by the symbol key and it doesn't mark the entry as used.

`\glsymbol`

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

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

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

Read in the final optional argument:

```
\def\@glsymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
```

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

`\Glsymbol`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsymbol}{\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}{\glsentrysymbol{#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}{\glsuppercase{\glsentrysymbol{#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}{\glsentrysymbolplural{#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*{\@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}{\glsentrysymbolplural{#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}{\glsuppercase{\glsentrysymbolplural{#2}#3}}%
}
```

`\glsuseri` behaves like `\gls` except it always uses the value given by the `user1` key and it doesn't mark the entry as used.

`\glsuseri`

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

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

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

Read in the final optional argument:

```
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
```

`\Glsuseri` behaves like `\glsuseri` except that the first letter is converted to uppercase.

`\Glsuseri`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseri}{\Glsuseri}
```

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

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

Read in the final optional argument:

```
\def\@Glsuseri@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentryuseri{#2}#3}%
}
```

\GLSuseri behaves like \glsuseri except that the link text is converted to uppercase.

\GLSuseri

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

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

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

Read in the final optional argument:

```
\def\@GLSuseri@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@Glsentryuseri{#2}#3}%
}
```

\glsuserii behaves like \gls except it always uses the value given by the user2 key and it doesn't mark the entry as used.

\glsuserii

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

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

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

Read in the final optional argument:

```
\def\@glsuserii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentryuserii{#2}#3}%
}
```

\Glsuserii behaves like \glsuserii except that the first letter is converted to uppercase.

\Glsuserii

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserii}{\Glsuserii}
```

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

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

Read in the final optional argument:

```
\def\@Glsuserii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryuserii{#2}#3}%  
}
```

\Glsuserii behaves like \glsuserii except that the link text is converted to uppercase.

\Glsuserii

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

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

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

Read in the final optional argument:

```
\def\@GLSuserii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuserii{#2}#3}%  
}
```

\glsuseriii behaves like \gls except it always uses the value given by the user3 key and it doesn't mark the entry as used.

\glsuseriii

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

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

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

Read in the final optional argument:

```
\def\@glsuseriii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryuseriii{#2}#3}%  
}
```

\Glsuseriii behaves like \glsuseriii except that the first letter is converted to uppercase.

\Glsuseriii

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
```


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

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

Read in the final optional argument:

```
\def\@Glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
```

\Glsuseriii behaves like \glsuseriii except that the link text is converted to uppercase.

\Glsuseriii

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

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

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

Read in the final optional argument:

```
\def\@GLSuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryuseriii{#2}#3}}%
}
```

\glsuseriv behaves like \gls except it always uses the value given by the user4 key and it doesn't mark the entry as used.

\glsuseriv

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

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

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

Read in the final optional argument:

```
\def\@glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
```

\Glsuseriv behaves like \glsuseriv except that the first letter is converted to uppercase.

\Glsuseriv

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
```

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

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

Read in the final optional argument:

```
\def\@Glsuseriv@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryuseriv{#2}#3}%  
}
```

\GLSuseriv behaves like \glsuseriv except that the link text is converted to uppercase.

\GLSuseriv

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

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

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

Read in the final optional argument:

```
\def\@GLSuseriv@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuseriv{#2}#3}%  
}
```

\glsuserv behaves like \gls except it always uses the value given by the user5 key and it doesn't mark the entry as used.

\glsuserv

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

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

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

Read in the final optional argument:

```
\def\@glsuserv@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryuserv{#2}#3}%  
}
```

\Glsuserv behaves like \glsuserv except that the first letter is converted to uppercase.

\Glsuserv

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserv}{\Glsuserv}
```

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

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

Read in the final optional argument:

```
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentryuserv{#2}#3}%
}
```

\GLSuserv behaves like \glsuserv except that the link text is converted to uppercase.

\GLSuserv

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

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

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

Read in the final optional argument:

```
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuserv{#2}#3}}%
}
```

\glsuservi behaves like \gls except it always uses the value given by the user6 key and it doesn't mark the entry as used.

\glsuservi

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

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

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

Read in the final optional argument:

```
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentryuservi{#2}#3}%
}
```

\Glsuservi behaves like \glsuservi except that the first letter is converted to uppercase.

\Glsuservi

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuservi}{\Glsuservi}
```

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

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

Read in the final optional argument:

```
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentryuservi{#2}#3}%
}
```

\GLSuservi behaves like \glsuservi except that the link text is converted to uppercase.

\GLSuservi

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

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

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

Read in the final optional argument:

```
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuservi{#2}#3}}%
}
```

Now deal with acronym related keys. First the short form:

\acrshort

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

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

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

Read in the final optional argument:

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

    \let\glsifplural\@secondoftwo
    \let\glsifscaps\@firstofthree
    \let\glsinsert\@empty
    \def\glsinserttext{%
      \acronymfont{\@glsentryshort{#2}}#3%
    }%
  }
```

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

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

\glspostlinkhook
}
```

`\Acrshort`

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

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\glsifscaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
  }%
}
```

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

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

\glspostlinkhook
}
```

`\ACRshort`

```
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
```

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

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

Read in the final optional argument:

```
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glentryshort{#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{\glentryshortpl{#2}}#3%
    }%
  }
```

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

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

```

    \glspostlinkhook
  }

\Acrshortpl
  \newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}

Add sentence case mapping.
  \glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
Define the un-starred form. Need to determine if there is a final optional argument
  \newcommand*{\ns@Acrshortpl}[2][{}]{%
    \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2}[]}]%
  }
Read in the final optional argument:
  \def\@Acrshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

      \def\glslabel{#2}%
      \let\glsifplural\@firstoftwo
      \let\gls caps case\@secondofthree
      \let\glsinsert\@empty
      \def\gls custom text{%
        \acronymfont{\Glsentryshortpl{#2}}#3%
      }%
    }%
  }

Call \@gls@link Note that \@gls@link sets \gls type.
  \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\ACRshortpl
  \newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
  \glsmfublocker{\ACRshortpl}

Define the un-starred form. Need to determine if there is a final optional argument
  \newcommand*{\ns@ACRshortpl}[2][{}]{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2}[]}]%
  }
Read in the final optional argument:
  \def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%

```

```

\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsescapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
}%

```

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

```

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

```

```

\glspostlinkhook
}

```

`\acrlong`

```

\newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}

```

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

```

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

```

Read in the final optional argument:

```

\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glsinsert\@empty

```

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

```

\def\glscustomtext{%
  \glsentrylong{#2}#3%
}%

```

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

```

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

```

```

\glspostlinkhook
}

```


`\AcrLong`

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

Add sentence case mapping.

```
\glsmfuaddmap{\AcrLong}{\AcrLong}
```

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

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

Read in the final optional argument:

```
\def\@AcrLong#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%  
  
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper  
  
    \def\glslabel{#2}%  
    \let\glsifplural\@secondoftwo  
    \let\glscapscase\@secondofthree  
    \let\glsinsert\@empty
```

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

```
\def\glscustomtext{%  
  \Glsentrylong{#2}#3%  
}%
```

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

```
\@gls@link[#1]{#2}{\csname gls@\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] []{%  
  \new@ifnextchar[{\@ACRLong{#1}{#2}}{\@ACRLong{#1}{#2} []}%  
}
```

Read in the final optional argument:

```
\def\@ACRLong#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

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

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

```
\def\gls customtext{%  
  \gls upper case{\gls entry long{#2}#3}%  
}%
```

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

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

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

Short plural:

`\acrlongpl`

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

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

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

Read in the final optional argument:

```
\def\@acrlongpl#1#2[#3]{%  
  \gls do if exists{#2}%  
  {%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

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

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

```
\def\gls customtext{%  
  \gls entry long pl{#2}#3%  
}%
```

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

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

```

    \glspostlinkhook
  }

\Acrlongpl
  \newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}

Add sentence case mapping.
  \glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
Define the un-starred form. Need to determine if there is a final optional argument
  \newcommand*{\ns@Acrlongpl}[2][ ]{%
    \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2} [ ]}%
  }
Read in the final optional argument:
  \def\@Acrlongpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

      \def\glslabel{#2}%
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@secondofthree
      \let\glsinsert\@empty

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont
only designed for short form).
      \def\glscustomtext{%
        \Glsentrylongpl{#2}#3%
      }%
Call \@gls@link. Note that \@gls@link sets \glstype.
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
  }

\ACRlongpl
  \newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
  \glsmfublocker{\ACRlongpl}

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

```

Read in the final optional argument:

```
\def\ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
```

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

```
\def\glscustomtext{%
  \glsuppercase{\glsentrylongpl{#2}#3}%
}%
```

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

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

\glspostlinkhook
}
```

2.11.2 Displaying entry details without adding information to the glossary

These commands merely display entry information without adding entries in the associated file or having hyperlinks.

`\@gls@entry@field` Generic version.

```
\@gls@entry@field{<label>}{<field>}
```

Now using `\csuse` instead of `\csname` to expand to nothing rather than `\relax` if undefined.

```
\newcommand*\@gls@entry@field[2]{%
  \csuse{gls@glstoklabel{#1}@#2}%
}
```

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

`\glsletentryfield`

```
\newcommand*\glsletentryfield[3]{%
  \letcs{#1}{gls@glstoklabel{#2}@#3}%
}
```

`\@Gls@entry@field` Generic first letter uppercase version.

```
\@Gls@entry@field{<label>}{<field>}
```

Check for PDF bookmarks performed by `\@Glsentryfield`.

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

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

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

`\@Glsentryfield`

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

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

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

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

`\glsentryname`

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

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

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

  \glsmfuexcl{\Glsentryname}
```

`\@Gls@entryname` This is a workaround in the event that the user defies the warning in the manual about not using `\Glsname` or `\Glsentryname` with acronyms. The check for PDF bookmarks is in `\Glsentryname`. First the default behaviour:

```
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
```

`\@Gls@acentryname` Now the behaviour when `\setacronymstyle` is used. The original definition of this command used `mfirstuc` internal commands that are now being deprecated.

```
\newrobustcmd*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
```

Just need to ensure that `\acrnameformat` is expanded, which may cause problems as non-robust semantic commands may expand to content that interferes with sentence-casing.

```
  \protected@edef\@glo@text{\@gls@entry@field{#1}{name}}%
  \expandafter\glssentencecase\expandafter{\@glo@text}%
  }%
  {%
```

Not an acronym

```
  \@Gls@entry@field{#1}{name}%
  }%
}
```

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

```
\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
  }
```

`\@gls@getbody` is defined by `mfirstuc` (which used to be part of glossaries).

```

\expandafter\@gls@getbody\@glo@text{}\@nil
\expandafter\ifx\@gls@body\glsentrylong\relax
\expandafter\Glsentrylong\@gls@rest
\else
\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax

```

Temporarily make `\glsentryshort` behave like `\Glsentryshort`. (This is on the assumption that the argument of `\acronymfont` is `\glsentryshort` `{\label}`}, as that's the behaviour of the predefined acronym styles.) This is scoped to localise the effect of the assignment.

```

{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\glsentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%

```

Not an acronym

```

\@Gls@entry@field{#1}{name}%
}%
}

```

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

```

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

```

```

\Glsentrydesc
\newcommand*\Glsentrydesc[1]{\@Gls@entry@field{#1}{desc}}

```

Version 4.54: sentence case mapping removed as `\glsentrydesc` is expandable. This ensures that `\glsentrydesc` can be converted to all-caps if required. However, `\Glsentrydesc` needs to be added as an exclusion. (Similarly for the other `\Glsentry<field>` commands)

```
\glsmfuexcl{\Glsentrydesc}
```

Plural form:

```
\glsentrydescplural
```

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

```
\Glsentrydescplural
```

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

```
\glsmfuexcl{\Glsentrydescplural}
```

Get the entry text, as specified by the `text` key when the entry was defined. The argument is the label associated with the entry:

```
\glsentrytext
```

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

```
\Glsentrytext
```

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

```
\glsmfuexcl{\Glsentrytext}
```

Get the plural form:

```
\glsentryplural
```

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

```
\Glsentryplural
```

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

```
\glsmfuexcl{\Glsentryplural}
```

Get the symbol associated with this entry. The argument is the label associated with the entry.


```
\glsentrysymbol
  \newcommand*\glsentrysymbol[1]{%
    \@gls@entry@field{#1}{symbol}%
  }
```

```
\Glsentrysymbol
  \newcommand*\Glsentrysymbol[1]{%
    \@Glsentryfield{#1}{symbol}%
  }
```

```
\glsmfuexcl{\Glsentrysymbol}
```

Plural form:

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

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

```
\glsmfuexcl{\Glsentrysymbolplural}
```

Get the entry text to be used when the entry is first used in the document (as specified by the first key when the entry was defined).

```
\glsentryfirst
  \newcommand*\glsentryfirst[1]{%
    \@gls@entry@field{#1}{first}%
  }
```

```
\Glsentryfirst
  \newcommand*\Glsentryfirst[1]{%
    \@Glsentryfield{#1}{first}%
  }
```

```
\glsmfuexcl{\Glsentryfirst}
```

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

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

```

\Glsentryfirstplural
    \newcommand*{\Glsentryfirstplural}[1]{%
        \@Glsentryfield{#1}{firstpl}%
    }

    \glsmfuexcl{\Glsentryfirstplural}

\glscapitalisewords Make it easier to switch from \capitalisewords to \capitalisefmtwords.
    \newcommand{\glscapitalisewords}[1]{%
        \capitalisewords{#1}%
    }

\glsentrytitlecase Now uses sentence case in PDF bookmark.
    \newcommand*{\glsentrytitlecase}[2]{%
        \glstexorpdfstring
            {@glsentrytitlecase{#1}{#2}}%
            {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
    }

\@glsentrytitlecase
    \newrobustcmd*{\@glsentrytitlecase}[2]{%
        \glsdoifexists{#1}%
        {%
            \glsfieldfetch{#1}{#2}{\@gls@value}%
            \expandafter\glscapitalisewords\expandafter{\@gls@value}%
        }%
    }

    Display the glossary type with which this entry is associated (as specified by
    the type key used when the entry was defined). This expands to a label so isn't
    intended to produce text in the document.

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

    Display the sort text used for this entry. Note that the sort key is sanitize,
    so unexpected results may occur if the sort key contained commands. This is
    likewise not intended to produce text in the document, other than for debugging
    purposes.

\glsentrysort
    \newcommand*{\glsentrysort}[1]{%
        \@gls@entry@field{#1}{sort}%
    }

\glsentryparent Expands to the label of the entry's parent. This expands to a label so isn't
    intended to produce text in the document.
    \newcommand*{\glsentryparent}[1]{%
        \@gls@entry@field{#1}{parent}%
    }

```

```

\glsentryuseri Get the first user key (as specified by the user1 when the entry was defined).
The argument is the label associated with the entry.
  \newcommand*\glsentryuseri[1]{%
    \@gls@entry@field{#1}{useri}%
  }

\Glsentryuseri
  \newcommand*\Glsentryuseri[1]{%
    \@Glsentryfield{#1}{useri}%
  }

  \glsmfuexcl{\Glsentryuseri}

\glsentryuserii Get the second user key (as specified by the user2 when the entry was defined).
The argument is the label associated with the entry.
  \newcommand*\glsentryuserii[1]{%
    \@gls@entry@field{#1}{userii}%
  }

\Glsentryuserii
  \newcommand*\Glsentryuserii[1]{%
    \@Glsentryfield{#1}{userii}%
  }

  \glsmfuexcl{\Glsentryuserii}

\glsentryuseriii Get the third user key (as specified by the user3 when the entry was defined).
The argument is the label associated with the entry.
  \newcommand*\glsentryuseriii[1]{%
    \@gls@entry@field{#1}{useriii}%
  }

\Glsentryuseriii
  \newcommand*\Glsentryuseriii[1]{%
    \@Glsentryfield{#1}{useriii}%
  }

  Add sentence case mapping.
  \glsmfuexcl{\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}

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

\Glsentryuseriv
    \newcommand*{\Glsentryuseriv}[1]{%
        \@Glsentryfield{#1}{useriv}%
    }

    \glsmfuexcl{\Glsentryuseriv}

\glsentryuserivi 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*{\glsentryuserivi}[1]{%
        \@gls@entry@field{#1}{userivi}%
    }

\Glsentryuserivi
    \newcommand*{\Glsentryuserivi}[1]{%
        \@Glsentryfield{#1}{userivi}%
    }

    \glsmfuexcl{\Glsentryuserivi}

\glsentryshort Get the short key (as specified by the short the entry was defined). The argument
is the label associated with the entry.
    \newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}

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

    \glsmfuexcl{\Glsentryshort}

\glsentryshortpl Get the short plural key (as specified by the shortplural the entry was defined).
The argument is the label associated with the entry.
    \newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}

```

```

\Glsentryshortpl
    \newcommand*\Glsentryshortpl}[1]{%
        \@Glsentryfield{#1}{shortpl}%
    }

    \glsmfuexcl{\Glsentryshortpl}

\glsentrylong Get the long key (as specified by the long the entry was defined). The argument
is the label associated with the entry.
    \newcommand*\glsentrylong}[1]{\@gls@entry@field{#1}{long}}

\Glsentrylong
    \newcommand*\Glsentrylong}[1]{%
        \@Glsentryfield{#1}{long}%
    }

    \glsmfuexcl{\Glsentrylong}

\glsentrylongpl Get the long plural key (as specified by the longplural the entry was defined).
The argument is the label associated with the entry.
    \newcommand*\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}

\Glsentrylongpl
    \newcommand*\Glsentrylongpl}[1]{%
        \@Glsentryfield{#1}{longpl}%
    }

    \glsmfuexcl{\Glsentrylongpl}
    Short cut macros to access full form:

\glsentryfull This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*\glsentryfull}[1]{%
        \glsentrylong{#1} (\glsentryshort{#1})%
    }

\Glsentryfull This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*\Glsentryfull}[1]{%
        \Glsentrylong{#1} (\glsentryshort{#1})%
    }

    \glsmfuexcl{\Glsentryfull}

\glsentryfullpl This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*\glsentryfullpl}[1]{%
        \acrffullformat{\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}{ \& }
```

`\gls hyperlink` Provide a hyperlink to a glossary entry without adding information to the glossary file. The entry needs to be added using a command like `\gls link` or `\gls add` 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*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@gls link{\glo link prefix\glsdetoklabel{#2}}{#1}}

```

2.12 Adding an entry to the glossary without generating text

The following keys are provided for `\gls add` and `\gls add all`:

```
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
```

```
\define@key{glossadd}{format}{\def\@gls@numberformat{#1}}
```

This key is only used by `\gls add all`:

```
\define@key{glossadd}{types}{\def\@glo@type{#1}}
```

```
\gls add[options]{label}
```

Add a term to the glossary without generating any link text. The optional argument indicates which counter to use, and how to format it (using a key-value list) the second argument is the entry label. Note that *options* only has two keys: `counter` and `format` (the `types` key will be ignored).

`\glsadd`

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

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

```
\@gls@adjustmode  
\glsdoifexists{#2}%  
{%
```

```
\def\@glsnumberformat{glsnumberformat}%
```

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

Store the entry's counter in `\theglsentrycounter`

```
\@gls@saveentrycounter
```

Define sort key if necessary:

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

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

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

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

```
\glsmfuexcl{\glsadd}
```

`\@gls@adjustmode`

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

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

```
\glsaddall[option list]
```

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

`\glsaddall`

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



```
\glsaddallunused[<glossary types>]
```

`\glsaddallunused`

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

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

`\glsignore`

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

2.13 Creating associated files

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

The special characters are stored in `\@gls@actualchar`, `\@gls@encapchar`, `\@gls@levelchar` and `\@gls@quotechar` to make them easier to use later, but don't change these values, because the characters are encoded in the command definitions that are used to escape the special characters (which means that the user no longer needs to worry about `makeindex` special characters).

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

`\glsopenbrace` Define `\glsopenbrace` to make it easier to write an opening brace to a file.

```
\edef\glsopenbrace{\expandafter@gobble\string\{}
```

`\glsclosebrace` Define `\glsclosebrace` to make it easier to write an opening brace to a file.

```
\edef\glsclosebrace{\expandafter@gobble\string\}}
```

`\glsbackslash` Define `\glsbackslash` to make it easier to write a backslash to a file.

```
\edef\glsbackslash{\expandafter@gobble\string\}}
```

`\glsquote` Define command that makes it easier to write quote marks to a file in the event that the double quote character has been made active.

```
\edef\glsquote#1{\string"#1\string"}
```

`\glspercentchar` Define `\glspercentchar` to make it easier to write a percent character to a file.

```
\edef\glspercentchar{\expandafter\@gobble\string\%}
```

`\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]{%
    \glsnoxywarning\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]{%
    \glsnoxywarning\GlsSetXdyNumberGroupOrder}

```

```

        \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
        \@GlsSetXdyNumberGroupOrder}
    \fi

\@glsminrange Define the minimum number of successive location references to merge into a
range.
    \newcommand*{\@glsminrange}{2}

\GlsSetXdyMinRangeLength Set the minimum range length. The value must either be none or a positive
integer. The glossaries package doesn't check if the argument is valid, that is
left to xindy.
    \ifglxindy
        \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \renewcommand*{\@glsminrange}{#1}}
    \else
        \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \glsnxindywarning\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
            }%
    \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@suffiF
\@onelevel@sanitize\gls@suffiFF

\ifx\gls@suffiF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffiFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiFF" :length 2 :ignore-end)}%
\fi
```

Specify how to format locations.

```
\write\glswrite{^^J; define format to use for locations^^J}%  
\write\glswrite{\@xdylocref}%
```

Specify how to separate letter groups.

```
\write\glswrite{^^J; define letter group list format^^J}%  
\write\glswrite{(markup-letter-group-list  
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
```

Define letter group headings.

```
\write\glswrite{^^J; letter group headings^^J}%  
\write\glswrite{(markup-letter-group  
:open-head \string"\string\glsgroupheading  
\glsopenbrace\string"^^J\space\space\space  
:close-head \string"\glsclosebrace\string")}%
```

Define additional letter groups.

```
\write\glswrite{^^J; additional letter groups^^J}%  
\write\glswrite{\@xdylettergroups}%
```

Define additional sort rules

```
\write\glswrite{^^J; additional sort rules^^J}  
\write\glswrite{\@xdysortrules}%
```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the style file

```
\closeout\glswrite
```

Suppress any further calls.

```
\let\writeist\relax  
\let\noist\@gls@noist@toolate  
}
```

\else

Code to use if makeindex is required.

```
\edef\@gls@actualchar{\string?}  
\edef\@gls@encapchar{\string|}  
\edef\@gls@levelchar{\string!}  
\edef\@gls@quotechar{\string"}%  
\let\GlsSetQuote\gls@nosetquote  
\def\writeist{\relax  
\ifundef{\glswrite}{\newwrite\glswrite}{\relax  
\openout\glswrite=\istfilename  
\write\glswrite{\glspercentchar\space makeindex style file  
created by the glossaries package}  
\write\glswrite{\glspercentchar\space for document  
'\jobname' on \the\year-\the\month-\the\day}  
\write\glswrite{actual '\@gls@actualchar'}  
\write\glswrite{encap '\@gls@encapchar'}  
\write\glswrite{level '\@gls@levelchar'}  
\write\glswrite{quote '\@gls@quotechar'}}
```

```

\write\glswrite{keyword \string\string\glossaryentry\string"}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string"}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string\string%\string\n\string"}
\write\glswrite{item_1 \string\string%\string\n\string"}
\write\glswrite{item_2 \string\string%\string\n\string"}
\write\glswrite{item_01 \string\string%\string\n\string"}
\write\glswrite{item_x1
  \string\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string\string%\string\n\string"}
\write\glswrite{item_x2
  \string\string\relax \string\glsresetentrylist\string\n
  \string"}

\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string\string\}\string}\string"}
\write\glswrite{delim_n \string\string\delimN \string"}
\write\glswrite{delim_r \string\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi

```


Hook for any additional information:

```
\@gls@writeisthook
```

Close the file and disable `\writeist`.

```
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi
```

`\GlsSetWriteIstHook` Allow user to append information to the style file.

```
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
```

`\@gls@writeisthook`

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

`\GlsSetQuote` Allow user to set the `makeindex` quote character. This is primarily for ngerman users who want to use `makeindex`'s `-g` option.

```
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}}%
```

If German is in use, set the extra `makeindex` option so `makeglossaries` can pick it up.

```
\@ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}}%
}%
}%
}
```

Need to redefine `\@gls@checkquote`

```
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#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{\cname#1\endcname}\expandonce{\cname#1\endcname}%
\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@checkescllevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkesquote####1%
\expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkesquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkesquote{%
\noexpand\@gls@checkesquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%

```

```

\newcommand\ifx\noexpand\null####3\noexpand\null
\newcommand\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 T_EX 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\@glsoopenfile\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}%
}%
}
```

`\glsoopenfile` Open write file associated with the given glossary.

```
\newcommand*\glsoopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glstype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file
    \jobname.\csname @glstype@#2@out\endcsname}%
}
```

`\@closegls`

```
\newcommand*\@closegls}[1]{%
  \closeout\csname glo@#1@file\endcsname
}
```

`\gls@automake` Unstarred form of `\ifglossaryexists` is used as `\gls@automake` can't be used with an ignored glossary.

```
\ifglsxindy
\newcommand*\gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
  }
```

```

        {\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%

\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\@gls@run@unrestricted@shell{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@run@output@file{\gls@istfilebase}
\@gls@codepage
-t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
-o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
\@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
}%
}%
}%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-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

```

`\@gls@automake@immediate` Unstarred form of `\ifglossaryexists` is used as `\@gls@automake@immediate` can't be used with an ignored glossary. Note that “out” means the output file from this package's point of view. It will be the input file from the indexing application's point of view.

```

\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%

```

```

\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-M ord/letorder }}%
  {\let\@gls@order\@empty}%
Set \@gls@langmod:
  \gls@xdy@set@lang{#1}%

  \ifdefvoid\gls@codepage
  {\let\@@gls@codepage\@empty}%
  {\def\@@gls@codepage{-C \gls@codepage\space}}%
  \@gls@run@unrestricted@shell{xindy
  -I xindy
  \@gls@order
  -L \@gls@langmod\space
  -M \@gls@run@output@file{\gls@istfilebase}
  \@gls@codepage
  -t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
  -o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
  \@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
  }%
  }%
  {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
  doesn't exist. Rerun may be required}}%
  }%
  {\%
  \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
  }%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
  \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
  {%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-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

```

`\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
  \def\glsindexingsetting{makeindex}%

```



```

\fi
\domakeglossaries
{%

```

If automake=immediate setting is on, use the shell escape now.

```

\@gls@@automake@immediate

```

Define the write used for style file also used for all other output files if savewrites=true.

```

\ifundef{\glswrite}{\newwrite\glswrite}{}%

```

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

```

\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}

```

If \@gls@extramakeindexopts has been defined, write it:

```

\ifundef\@gls@extramakeindexopts
{%
\%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{}}
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%

```

Write the name of the style file to the aux file (needed by makeglossaries)

```

\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\@glsorder}}

```

Iterate through each glossary type and activate it.

```

\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{}}{}{}%
\@makeglossary{\@glo@type}}%
}%

```

New glossaries must be created before \makeglossaries so disable \newglossary.

```

\renewcommand*\newglossary[4][]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%

```

Any subsequent instances of this command should have no effect. The deprecated \makeglossary is not redefined here as it either implements \makeglossaries or has been restored to its original definition (in which case it shouldn't be changed).

```

\let\@makeglossary\@gobble
\let\makeglossaries\relax

```

Disable all commands that have no effect after \makeglossaries

```

\@disable@onlypremakeg

```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Suppress warning about no \makeglossaries

```
\let\warn@nomakeglossaries\relax
```

Activate warning about missing \printglossary

```
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
```

Declare list parser for \glsdisplaynumberlist

```
\ifglsavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@dodolistparser
\fi
```

Prevent user from also using \makenoidxglossaries

```
\let\makenoidxglossaries\@no@makeglossaries
```

Prohibit sort key in printgloss family:

```
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
```

Check the automake setting:

```
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
```

Check the sort setting:

```
\@glo@check@sortallowed\makeglossaries
}%
}
```

Must occur in the preamble:

```
\@onlypreamble{\makeglossaries}
```

saries@warn@noprintglossary

```
\newcommand\@makeglossaries@warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
{%
\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
```

```

        found. ^^J(Remove \string\makeglossaries\space if you
        don't want any glossaries.) ^^JThis document will not
        have a glossary}%
    }%
}%

```

`\glswrite` The definition of `\glswrite` has now been moved to `\makeglossaries` so that it's only defined if needed.

If `\makeglossaries` hasn't been used, issue a warning. Also issue a warning if neither `\printglossaries` nor `\printglossary` have been used.

```

\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}

```

`\makenoidxglossaries` Analogous to `\makeglossaries` this activates the commands needed for `\printnoidxglossary`

```

\newcommand*\makenoidxglossaries{%
  \def\glsindexingsetting{noidx}%
  \@domakeglossaries
  {%

```

Redefine empty glossary warning:

```

  \renewcommand{\@gls@noref@warn}[1]{%
    \GlossariesWarning{Empty glossary for
      \string\printnoidxglossary[type={##1}].
    Rerun may be required (or you may have forgotten to use
    commands like \string\gls)}%
  }%

```

Don't escape makeindex/xindy characters:

```

\let\@gls@checkmkidxchars\@gobble

```

Don't escape locations:

```

\glsesclocationsfalse

```

Write glossary information to aux instead of glossary files

```

\let\@do@wrglossary\gls@noidxglossary

```

Switch on group headings that use the character code:

```

\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle

```

Allow see key:

```

\let\gls@checkseeallowed\relax

```

Redefine cross-referencing macro:

```

\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef\@gls@label{\glsdetoklabel{##1}}%
  \protected@write\@auxout{}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
  }%
}

```

```

    {%
      \string\glsseeformat##2}%
    }%
  }%
  \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@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%

```

Suppress warning about no \makeglossaries

```

\let\warn@nomakeglossaries\relax

```

Prevent user from also using \makeglossaries

```

\let\makeglossaries\@no@makeglossaries

```

Allow sort key in printgloss family:

```

\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
}

```

Initialise default sort order:

```

\def\@glo@sorttype{\@glo@default@sorttype}%
}%

```

All entries must be defined in the preamble:

```

\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%

```

Redefine \glsentrynumberlist

```

\renewcommand*\glsentrynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
}

```

```

    {%
      ??\glsdoifexists{##1}%
      {%
        \GlossariesWarning{Missing location list for ‘##1’. Either
          a rerun is required or you haven’t referenced the entry}%
      }%
    }%
  }%
}

Redefine \glsdisplaynumberlist
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}
}

```

Provide a generic way of iterating through the number list:

```

\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%

```

```

        \GlossariesWarning{Missing location list for ‘##1’. Either
          a rerun is required or you haven’t referenced the entry}%
      }%
    }%
    \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
    \let\glsseeformat\@gls@org@glsseeformat
  }%
  Modify sanitize sort function
  \let\@gls@sanitizesort\@gls@noidx@sanitizesort
  \let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
  \@gls@noidx@setsanitizesort
  Check sort option allowed.
  \@gls@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{<label>}{<handler>}{<xr handler cs>}
```

`\glsnumberlistloop`

```

\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}

```

`\glsnoidxnumberlistloophandler` Handler macro for `\glsnumberlistloop`. (The argument should be in the form `\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<n>}`)

```

\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}

```

`\@no@makeglossaries` Can’t use both `\makeglossaries` and `\makenoidxglossaries`

```

\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}

```

```

\@gls@noref@warn Warning when no instances of \@gls@reference found.
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}

```

2.14 Writing information to associated files

`\gls@noidxglossary` Write the glossary information to the aux file (for the ‘noidx’ method).

```

\newcommand*{\gls@noidxglossary}{%
  \protected@write\auxout{\glswrglossdisablelocationcmds}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@glslocref}}%
      }%
    }%
  \@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.

```

  \forallglossaries{\@glo@type}{%

```

Check for empty glossaries (patch provided by Patrick Häcker)

```

    \ifcsundef{glo@\@glo@type @filetok}%

```

```

    {%

```

```

      \def\gls@tmp{}%

```

```

    }%

```

```

    {%

```

```

      \edef\gls@tmp{\expandafter\the

```

```

        \csname glo@\@glo@type @filetok\endcsname}%

```

```

    }%

```

```

  \ifx\gls@tmp\@empty

```

```

    \ifx\@glo@type\glsdefaulttype

```

```

      \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
        entries.^^JRemember to use package option ‘nomain’ if

```

you

```

        don’t want to^^Juse the main glossary}%

```

```

    \else

```

```

      \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
        entries}%

```

```

    \fi

```

```

  \else

```

```

    \@glsopenfile{\glswrite}{\@glo@type}%

```

```

    \immediate\write\glswrite{%

```

```

      \expandafter\the

```

```

        \csname glo@\@glo@type @filetok\endcsname}%

```

```

    \immediate\closeout\glswrite

```

```

  \fi

```

```

}%

```

```

}

```

As from v4.10, the `\glossary` command isn’t used by the `glossaries` package. Since the user isn’t expected to use this command (as `glossaries` takes care of

the particular format required for `makeindex/xindy`) there's no need for a user level command. Using a custom internal command prevents any conflict with other packages (and with the `\mark` mechanism).

The associated number should be stored in `\theglsentrycounter` before using `\gls@glossary`.

```
\gls@glossary
  \newcommand*\gls@glossary}[1]{%
    \gls@glossary{#1}%
  }
```

```
\gls@glossary{<type>}{<indexing info>}
```

```
\@gls@glossary
```

(In v4.10, `\@glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\@makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as `memoir` changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\@makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within `<indexing info>` (so you can do, for example, `\index{%\@%}`), and the original design of `\@glossary` here was actually a legacy from the old `glossary` package. With the `glossaries` package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the sort key), the actual value (given by `\glosentry{<label>}` or `\subglosentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glspercentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```
\newcommand*\@gls@glossary}[2]{%
  \if@gls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}
```

This is a convenience command to set `\@gls@glossary`. It's used by `\@makeglossary` and then redefined to do nothing, as it only needs to be done once.

```
\@gls@renewglossary
```

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

```

\gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
\let\@gls@renewglossary\@empty
}

```

The `\gls@wrglossary` command is defined to have two arguments. The first argument is the glossary type, the second argument is the glossary entry (the format of which is set in `\glslink`).

```

\gls@wrglossary
\newcommand*\@gls@wrglossary}[2]{%
\ifgls@save@writes
\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@disable@pageref@expansion}{#2}%
}%
{%
\ifignoredglossary{#1}{%
{%
\GlossariesWarning{No file defined for glossary ‘#1’}%
}%
}%
\fi
\endgroup\@esphack
}

```

```

\@do@wrglossary
\newcommand*\@do@wrglossary}[1]{%
\gls@ifnotmeasuring{\gls@writeentry{#1}{\@do@wrglossary{#1}}}%
}

```

`\gls@writeentry` 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*\@gls@writeentry}[2]{%
\ifgls@index@only@first
\ifgls@used{#1}{#2}%
\else
#2%
\fi
}

```

`\gls@protected@pagefmts` List of page formats to be protected against expansion.

```

\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphapage,%
\gls@Alphapage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

```

`\gls@disablepagerefexpansion` As from v4.50, this will only do the page protection formats for `esclocations=true`.

```
\newcommand*{\gls@disablepagerefexpansion}{%
  \glswrglossdisablelocationcmds
  \ifglseclocations
    \@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 (`\<csname>\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 orgls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
      }%
    }%
  }
```

```

        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gl#1\endcsname}%
    }%
  }%
\else
  \GlsWarnAddProtectedPageFmt
\fi
}

\GlsWarnAddProtectedPageFmt
\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}

\wrglossarynumberhook Hook used by \@@do@wrglossary
\newcommand*\@wrglossarynumberhook{}

\@wrglossary@pageformat
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}

\@@do@wrglossary Write the glossary entry in the appropriate format.
\newcommand*\@@do@wrglossary[1]{%
  \ifglseclocations
    \@@do@esc@wrglossary{#1}%
  \else
    \@@do@noesc@wrglossary{#1}%
  \fi
}

glswrglossdisableanchorcmds
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}

\glswrglosslocationtarget Must expand. May be used to alter the location suffix in the \glshypernumber
anchor.
\newcommand{\glswrglosslocationtarget}[1]{#1}

\glswrglosslocationtextfmt Used to encapsulate the hyperlink text in \glshypernumber.
\newcommand{\glswrglosslocationtextfmt}[1]{#1}

\@@do@noesc@wrglossary Write the glossary entry in the appropriate format. The locations don't need
to be pre-processed before writing the information to the glossary file, but the
prefix still needs to be found.
\newcommand*\@@do@noesc@wrglossary[1]{%

```

Don't fully expand yet.

```
\expandafter\def\expandafter\@glslocref\expandafter{\theglsentrycounter}%  
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglsentrycounter}%
```

Find the prefix if `\@glsHlocref` and `\@glslocref` aren't the same.

```
\ifx\@glsHlocref\@glslocref  
  \def\@glo@counterprefix{}%  
\else
```

The value of the counter isn't important here as it's the prefix that's of interest. (`\c@page` will have the same value in both `\theglsentrycounter` and `\theHglsentrycounter` 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\c@#1\endcsname}}  
%
```

where `\tallynum` is a robust command that takes a number as its argument. With all indexing methods other than `xindy`, a deferred write with `\thepage` as the location will expand to `\tallynum{<n>}` where `<n>` is the page number. Since the write is deferred, the page number is correct. (`makeindex` won't accept this location format, but `\makenoidxglossaries` and `bib2gls` are quite happy with

it.) Unfortunately, this fails with `xindy` because `xindy` interprets this location as `\tallynum{<n>}` because `\t` represents a the character “t”. The location must be written as `\\tallynum{<n>}`.

This means that the location `\tally{page}` must be expanded and then the backslashes must be doubled. Unfortunately `\c@page` mustn’t be expanded until the deferred write is performed, so the location actually needs to be expanded to `\tallynum{\the\c@page}` but the backslashes in `\the\c@page` mustn’t be escaped. All other backslashes must be escaped. (In this case, only the backslash in `\tallynum` but the location format may include other control sequences.) The code below works on the assumption that commands like `\tally` are defined in the form

```
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
%
```

(note the use of `\expandafter` and `\name`) or in the form

```
%\newcommand{\tally}[1]{\tallynum{\arabic{#1}}}
%
```

In the second case, `\arabic` is one of the known commands that’s temporarily adjusted to prevent `\c@page` from being prematurely expanded. In the first case, `\the` is temporarily modified (unless `\glswrallowprimitivemodsfalse`) to check if it’s followed by `\c@page`. The `\expandafter` ensures that it is. If `\tally` is defined in another way that hides `\c@page` for example using `\the\value{#1}` then the process fails.

With `makeindex`, `\tallynum` needs to expand to just the decimal number while writing the location to the glossary file, otherwise `makeindex` will reject it. This can be done by defining `\glstallypage` so that `\tally` can locally be set to `\arabic` while expansion is occurring. Again, `\c@page` must be protected from expansion until the deferred write occurs.

The expansion before the write occurs also allows the hyper prefix to be determined where `\theH{<counter>}` is defined in the form `<prefix>.\the{<counter>}`. It’s possible (although again unlikely) that a `makeindex` character might occur in the prefix, which therefore needs escaping. The prefix is passed as the optional argument of `\setentrycounter` which is needed by commands like `\glshypernumber` to create a hyperlink for a given counter (like `\hyperpage` but for an arbitrary counter).

```
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
```

First a bit of hackery to prevent premature expansion of `\c@page`. Store original definitions (scoped):

```
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
```

```

\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman

```

Redefine:

```

\ifglswrallowprimitivemods

```

The redefinition of `\the` to use `\expandafter` solves the problem of `\the\csname c@{counter}\endcsname` but is only a partial solution to the problem of `\the\value`. With `\value`, `\c@page` is too deeply hidden and will be expanded too soon, but at least there won't be an error.

```

\def\gls@the##1{%
  \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
  \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%

\def\romannumeral##1{%
  \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
  \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
  \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
  \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
  \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%

```

Add hook to allow for other number formats:

```

\@wrglossarynumberhook

```

Prevent expansion:

```

\gls@disablepagerefexpansion

```

Now store location in `\@glslocref`:

```

\protected@xdef\@glslocref{\theHglentrycounter}%
\endgroup

```

Escape any special characters. It's possible that with `makeindex` the separator might be a `makeindex` special character. Although not likely, it still needs to be taken into account.

```

\@gls@checkmkidxchars\@glslocref

```

Check if the hyper-location is the same as the location and set the hyper prefix.

```

\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@glsHlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi

```


De-tok label if required

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

Write the information to file:

```
\@@do@@wrglossary  
}
```

```
\@@do@@wrglossary
```

```
\newcommand*\@@do@@wrglossary}{%
```

Determine whether to use xindy or makeindex syntax

```
\ifglsxindy
```

Need to determine if the formatting information starts with a (or) indicating a range.

```
\expandafter\glo@check@mkidxrangear\glsnumberformat\@nil  
\def\glo@range{}%  
\expandafter\if\glo@prefix\relax  
\def\glo@range{:open-range}%  
\else  
\expandafter\if\glo@prefix)\relax  
\def\glo@range{:close-range}%  
\fi  
\fi
```

Write to the glossary file using xindy syntax.

```
\gls@glossary{\csname glo@\gls@label @type\endcsname}{%  
(indexentry :tkey (\csname glo@\gls@label @index\endcsname)  
  
:locref \string"\@glo@counterprefix}{\@glslocref}\string" %  
:attr \string"\@gls@counter\@glo@suffix\string"  
\glo@range  
)  
}%  
\else
```

Convert the format information into the format required for makeindex

```
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%  
\@glo@counterprefix}%
```

Write to the glossary file using makeindex syntax.

```
\gls@glossary{\csname glo@\gls@label @type\endcsname}{%  
\string@glossaryentry{\csname glo@\gls@label @index\endcsname  
\@gls@encapchar\@glo@numfmt}{\@glslocref}}%  
\fi  
}
```

```
\@xp@gls@getcounterprefix
```

```
\newcommand*\@xp@gls@getcounterprefix[2]{%  
\bgroup  
\glswrglossdisableanchorcmds  
\protected@edef\do@gls@getcounterprefix{%
```

```

\noexpand\egroup
\noexpand\@gls@getcounterprefix{#1}{#2}%
}%
\do@gls@getcounterprefix
}

```

`\@gls@getcounterprefix` Get the prefix that needs to be prepended to counter in order to get the hyper counter. (For example, with the standard article class and `hyperref`, `\theequation` needs to be prefixed with `\langle section num \rangle|.` to get the equivalent `\theHequation`.) NB this assumes that the prefix ends with a dot, which is the standard. (Otherwise it makes the xindy location classes more complicated.)

The expansion has been moved to `\@xp@gls@getcounterprefix`.

```

\newcommand*\@gls@getcounterprefix[2]{%
\def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix

```

Warn if no prefix can be formed.

```

\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target '#2' can't be formed by
prefixing^^Jlocation '#1'. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you
will get the warning: "'name{\@gls@counter.#1}' has been^^J
referenced but does not exist"%
\fi
\fi
}

```

2.15 Glossary Entry Cross-References

`\do@seeglossary` Write the glossary entry with a cross reference. The first argument is the entry's label, the second must be in the form `[(tag)]{list}`, where `tag` is a tag such as "see" and `list` is a list of labels.

```

\newcommand{\do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%

```

```

(indexentry
 :tkey (\csname glo@#1@index\endcsname)
 :xref (\string"\@gls@xref\string")
 :attr \string"see\string"
 )
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}

```

`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```

\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{{#2#3}}%
\fi
}

```

`\@@gls@fixbraces`

```

\def\@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}

```

```

\glssee \glssee{<label>}{<cross-ref list>}
\newrobustcmd*{\glssee}[3][\seename]{%
\do@seeglossary{#2}{[#1]{#3}}}
\newcommand*{\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}

```

`\glsseeformat` The first argument specifies what tag to use (e.g. “see”), the second argument is a comma-separated list of labels. The final argument (the location) is ignored.

```

\newrobustcmd*{\glsseeformat}[3][\seename]{%
\emph{#1} \glsseelist{#2}}

```

`\glsseelist` `\glsseelist{<list>}` formats list of entry labels.

```

\newrobustcmd*{\glsseelist}[1]{%

```

If there is only one item in the list, set the last separator to do nothing.

```

\let\@gls@dolast\relax

```

Don’t display separator on the first iteration of the loop

```

\let\@gls@donext\relax

```

Iterate through the labels

```

\@for\@gls@thislabel:=#1\do{%

```

Check if on last iteration of loop

```
\ifx\@xfor@nextelement\@nnil
  \@gls@dolast
\else
  \@gls@donext
\fi
```

Display the entry for this label. (Expanding label as it's a temporary control sequence that's used elsewhere.)

```
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
```

Update separators

```
\let\@gls@dolast\glsseelastsep
\let\@gls@donext\glsseesep
}%
}
```

`\glsseelastsep` Separator to use between penultimate and ultimate entries in a cross-referencing list.

```
\newcommand*\glsseelastsep{\space\andname\space}
```

`\glsseesep` Separator to use between entries in a cross-referencing list.

```
\newcommand*\glsseesep{, }
```

`\glsseeitem` `\glsseeitem{<label>}` formats individual entry in a cross-referencing list.

```
\newrobustcmd*\glsseeitem[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
```

`\glsseeitemformat` Introduced in v3.0 to replace use of `\glsentryname` (originally in `\glsseeitem`) with `\glsentrytext` to avoid problems with the name key being sanitized. This ceased being a problem in v3.08a, but `\glsentrytext` was retained for backward-compatibility. As from v4.50, the default has reverted back to using `\glsentryname` unless the entry is an acronym (which could have a potentially long name).

```
\newcommand*\glsseeitemformat[1]{%
  \ifglshashshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
```

2.16 Displaying the glossary

An individual glossary is displayed in the text using `\printglossary[<key-val list>]`. If the type key is omitted, the default glossary is displayed. The optional argument can be used to specify an alternative glossary, and can also be used to set the style, title and entry in the table of contents. Available keys are defined below.

`\gls@save@numberlist` Provide command to store number list.

```
\newcommand*\gls@save@numberlist[1]{%
  \ifglssavenumberlist
    \toks@{#1}%
}
```

```

\protected@edef\do@writeaux@info{%
  \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\do@writeaux@info
\protected@write\@auxout{}{\do@writeaux@info}%
\fi
}

```

`\warn@noprintglossary` Warn the user if they have forgotten `\printglossaries` or `\printglossary`. (Will be suppressed if there is at least one occurrence of `\printglossary`. There is no check to ensure that there is a `\printglossary` for each defined glossary.)

```
\newcommand*{\warn@noprintglossary}{}%
```

`\printglossary` The TOC title needs to be processed in a different manner to the main title in case the translator and hyperref packages are both being used.

```
\ifcsundef{printglossary}{}%
{%

```

If `\printglossary` is already defined, issue a warning and undefine it.

```

\@gls@warnonglossdefined
\undef\printglossary
}

```

Neither `\printglossary` nor `\printnoidxglossary` can work with an ignored glossary (since ignored glossaries normally suppress indexing and there's no associated file for `makeindex/xindy` to process). However `\printunsrtglossary` can be used with an ignored glossary, so provide a command to warn if the glossary doesn't exist or is an ignored glossary.

`gloss@checkexists@noignored`

```

\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary '#1' is an ignored glossary}}%
    {\GlossariesWarning{Glossary '#1' doesn't exist}}%
  }%
}

```

`ss@checkexists@allowignored` For use with `\printunsrtglossary`.

```

\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary '#1' doesn't exist}}%
}

```

`\@printgloss@checkexists`

```
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
```

`\printglossary` has an optional argument. The default value is to set the glossary type to the main glossary.

```
\newcommand*\printglossary[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
```

The `\printglossaries` command will do `\printglossary` for each glossary type that has been defined. It is better to use `\printglossaries` rather than individual `\printglossary` commands to ensure that you don't forget any new glossaries you may have created. It also makes it easier to chop and change the value of the `acronym` package option. However, if you want to list the glossaries in a different order, or if you want to set the title or table of contents entry, or if you want to use different glossary styles for each glossary, you will need to use `\printglossary` explicitly for each glossary type.

`\printglossaries`

```
\newcommand*\printglossaries{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
```

`\printnoidxglossary` Provide an alternative to `\printglossary` that doesn't require an external indexing application.

```
\newcommand*\printnoidxglossary[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
```

`\printnoidxglossaries` Analogous to `\printglossaries`

```
\newcommand*\printnoidxglossaries{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
```

`\@printgloss@setsort` Initialise to do nothing.

```
\newcommand*\@printgloss@setsort{}
```

`\@gls@preglossaryhook`

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

`\@printglossary` Sets up the glossary for either `\printglossary` or `\printnoidxglossary`. The first argument is the options list, the second argument is the handler macro that deals with the actual glossary. This is also used by `glossaries-extra`'s `\printunsrtglossary` which may be used with an ignored glossary.

```
\newcommand{\@printglossary}[2]{%
```

Set up defaults.

```
\def\@glo@type{\glsdefaulttype}%
\def\glossarytitle{\csname @glo@type @title\endcsname}%
```

```

\def\glossarytoctitle{\glossarytitle}%
\let\org@glossarytitle\glossarytitle

\def\@glossarystyle{%
  \ifx\@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary ‘\@glo@type’. \MessageBreak
      Using fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def\gls@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@\@glo@type @title\endcsname
\glossarytitle
\fi

Allow a high-level user command to indicate the current glossary
\let\currentglossary\@glo@type

Enable individual number lists to be suppressed.
\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages

Enable individual number list to be activated:
\let\glsnextpages\@glsnextpages

Enable suppression of description terminators.
\let\nopostdesc\@nopostdesc

```

Set up the entry for the TOC

```
\gls@dotoctitle
```

Set the glossary style

```
\@glossarystyle
```

Added a way to fetch the current entry label (v3.08 updated for new `\glossentry` and `\subglossentry`, but this is now only needed for backward compatibility):

```
\let\gls@org@glossaryentryfield\glossentry
\let\gls@org@glossarysubentryfield\subglossentry
\renewcommand{\glossentry}[1]{%
  \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
  \gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
  \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
  \gls@org@glossarysubentryfield{##1}{##2}%
}%

\@gls@preglossaryhook
```

Now do the handler macro that deals with the actual glossary:

```
#2%
}%
```

End the current scope

```
\gls@printglossary@end
```

Reset `\glossaryentrynumbers`

```
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
```

Suppress warning about no `\printglossary`

```
\global\let\warn@noprntglossary\relax
}
```

```
\gls@printglossary@begin
```

```
\newcommand{\gls@printglossary@begin}{\begingroup}
```

```
\gls@printglossary@end See https://github.com/latex3/tagging-project/issues/48
```

```
\newcommand{\gls@printglossary@end}{%
\expandafter\endgroup\if@endpe\@doendpe\fi
}
```

```
\@print@glossary Internal workings of \printglossary dealing with reading the external file.
```

```
\newcommand{\@print@glossary}{%
```

Some macros may end up being expanded into internals in the glossary, so need to make `@` a letter. (Unlikely to be a problem since v3.08a but kept for backward compatibility.)

```
\makeatletter
```


Input the glossary file, if it exists.

```
\input@{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
```

If the glossary file doesn't exist, do \null. (This ensures that the page is shipped out and all write commands are done.) This might produce an empty page, but at this point the document isn't complete, so it shouldn't matter.

```
\IfFileExists{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
  {}%
  {\null}%
```

If xindy is being used, need to write the language dependent information to the .aux file for makeglossaries.

```
\ifglxindy
```

Set \gls@langmod:

```
\gls@xdy@set@lang{\glo@type}%
\edef\@do@auxoutstuff{%
  \noexpand\AtEndDocument{%
```

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

```
  \noexpand\immediate\noexpand\write\@auxout{%
    \string\providecommand\string\@xdy@language[2]{}%
  \noexpand\immediate\noexpand\write\@auxout{%
    \string\@xdy@language{\glo@type}{\gls@langmod}}%
  }%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
  \noexpand\AtEndDocument{%
```

If the user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
  \noexpand\immediate\noexpand\write\@auxout{%
    \string\providecommand\string\@gls@codepage[2]{}%
  \noexpand\immediate\noexpand\write\@auxout{%
    \string\@gls@codepage{\glo@type}{\gls@codepage}}%
  }%
}%
\@do@auxoutstuff
\fi
```

Activate warning if \makeglossaries hasn't been used.

```
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
```

The sort macros all have the syntax:

```
\@glo@sortmacro@⟨order⟩{⟨type⟩}
```

where $\langle order \rangle$ is the sort order as specified by the sort key and $\langle type \rangle$ 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:nm \__glossaries_actual_label:nm
    \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: $\langle handler-cs \rangle \{ \langle type \rangle \}$

```

\cs_new:Nn \__glossaries_do_sort:Nn
{
  \renewcommand \dtlnonlettergroup [ 1 ] { glssymbols }
  \renewcommand \dtlnumbergroup [ 1 ] { glnumbers }
  \__glossaries_set_sort_seq:n { #2 }
  \datatool_sortwordseq:NN

```

```

\l__glossaries_actual_sort_seq
#1
\tl_clear:c { @glsref@ #2 }
\seq_map_inline:Nn \l__glossaries_actual_sort_seq
{
\tl_set:Ne \l__glossaries_entry_letter_title_tl
{ \exp_args:No \exp_not:o { \DTLsortedletter { ##1 } } }
\tl_set:Ne \l__glossaries_entry_letter_group_tl
{ \DTLsortedletter { ##1 } }
\tl_set:Ne \l__glossaries_entry_label_tl { ##1 }
\group_begin:
\cs_set_eq:NN \__glossaries_actual:nm \__glossaries_actual_label:nm
\exp_args:NNNe
\group_end:
\tl_set:Nn \l__glossaries_entry_label_tl
{ \l__glossaries_entry_label_tl }
\exp_args:NnV \listcseadd { @glsref@ #2 } \l__glossaries_entry_label_tl

```

Save letter group information:

```

\tl_gset:ce
{ glo@ \l__glossaries_entry_label_tl @dtlsortgroup }
{
{ \l__glossaries_entry_letter_group_tl }
{ \exp_not:V \l__glossaries_entry_letter_title_tl }
}
}

```

Fetch the letter group from the sort value:

```

\cs_new:Npn \@gls@fetch@lettergrp
{
\tl_if_head_eq_meaning:VNTF \@gls@sort \__datatool_sorted:nm
{
\tl_set:Nx \@glo@thislettergrp
{ \exp_args:NV \DTLsortedletter \@gls@sort }
}
{
\exp_args:NVV \DTLassignlettergroup \@gls@sort \@gls@sort
\@glo@thislettergrp
}
}
}
{

```

Old version of datatool-base, so use old method.

```

\cs_new:Nn \__glossaries_sort_word:n
{
\@glo@sortentries{\@glo@sorthandler@word}{#1}
}
\cs_new:Nn \__glossaries_sort_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]{%
  \glo@sortentries@warning
  \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{\@glo@addchildren{<type>}{<parent>}}

```

\@glo@addchildren

```

\newcommand*{\@glo@addchildren}[2]{%

```

Scope to allow nesting.

```

  \bgroup
  \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%

```

Check this label hasn't already been added.

```

  \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
  {}%
  {%
    \listcsxadd{\@glsref@#1}{\@this@childlabel}%
  }%

```

Does this child have children?

```
\ifcsdef{@glo@sortingchildren@\@this@childlabel}%  
  {%  
    \glo@addchildren{#1}{\@this@childlabel}%  
  }%  
  {%  
  }%  
}%  
\egroup  
}
```

\@glo@do@sortentries

```
\newcommand*{\@glo@do@sortentries}[1]{%  
  \ifglshasparent{#1}%  
  {%
```

This entry has a parent, so add it to the child list

```
\protected@edef\@glo@parent{\csuse{glo@glsetoklabel{#1}@parent}}%  
\ifcsundef{@glo@sortingchildren@\@glo@parent}%  
  {%  
    \csdef{@glo@sortingchildren@\@glo@parent}{}%  
  }%  
  {}%  
\expandafter\@glo@sortedinsert  
  \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
```

Has the parent been added?

```
\xifinlistcs{\@glo@parent}{@gleref@\@glo@type}%  
  {%
```

Yes, it has so do nothing.

```
}%  
{%
```

No, it hasn't so add it now.

```
\expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%  
  }%  
}%  
  {%  
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%  
  }%  
}
```

\@glo@sortedinsert{<list>}{<entry label>}

\@glo@sortedinsert

Insert into list.

```
\newcommand*{\@glo@sortedinsert}[2]{%  
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%  
}%
```

The sort handlers need to be in the form required by datatool's `\dtl@sortlist` macro. These must set the count register `\dtl@sortresult` to either -1 ($\#1$ less than $\#2$), 0 ($\#1 = \#2$) or $+1$ ($\#1$ greater than $\#2$).

`\@glo@sorthandler@word`

```
\newcommand*\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{\#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{\#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
```

`\@glo@sorthandler@letter`

```
\newcommand*\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{\#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{\#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
```

`\@glo@sorthandler@case` Case-sensitive sort.

```
\newcommand*\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{\#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{\#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
```

`\@glo@sorthandler@nocase` Case-insensitive sort.

```
\newcommand*\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{\#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{\#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
}
```

```

\glo@do@compare
}

```

`\glo@sortmacro@word` Sort macro for ‘word’

```

\newcommand*{\@glo@sortmacro@word}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
\@glo@@sortentries{word}{#1}%
}%
{%
\PackageError{glossaries}{Conflicting sort options:^^J
\string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
\string\printnoidxglossary[sort=word]}{}}%
}%
}

```

`\@glo@sortmacro@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{}%
  \forglsentries[#1]{\@gls@thislabel}%
}

```

```

    {%
      \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
      {%
        \listeadadd{\@glo@sortinglist}{\@gls@thislabel}%
      }%
    }%
    Hasn't been referenced.
  }%
}%
\cslet{\@glsref@#1}{\@glo@sortinglist}%
}

```

`\@glo@sortmacro@def@do` This won't include parent entries that haven't been referenced.

```

\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}

```

`\@glo@sortmacro@use` Sort macro for 'use'. (No sorting is required, as the entries are already in order of use, so do nothing.)

```

\newcommand*{\@glo@sortmacro@use}[1]{

```

`\@print@noidx@glossary` Glossary handler for `\printnoidxglossary` which doesn't use an indexing application. Since `\printnoidxglossary` may occur at the start of the document, we can't just check if an entry has been used. Instead, the first pass needs to write information to the aux file every time an entry is referenced. This needs to be read in on the second run and stored in a list corresponding to the appropriate glossary.

Version 4.57: switch to constructing token list (as per `\printunsrtglossary`) and check for `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 redefining `\@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:V \@gls@loclist }
        }
      }
    }
  }
}
{

```

Doesn't have a parent

```

\tl_clear:N \l__glossaries_entry_letter_group_tl
\tl_clear:N \l__glossaries_entry_letter_title_tl
\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_grouptitle_tl }
    }
\l__glossaries_entry_letter_group_tl
\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
    }
\l__glossaries_entry_letter_group_tl
\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 \l__glossaries_lettergroup_title:nn
{
    \exp_not:n { #2 }
}
}

```

```
\ExplSyntaxOff
```

```
\gls@old@print@noidx@glossary Original definition (pending deletion):
```

```
\newcommand*{\@gls@old@print@noidx@glossary}{%  
  \ifcsdef{@glsref@ \@glo@type}%  
  {%
```

```
Sort the entries:
```

```
  \ifcsdef{@glo@sortmacro@ \@glo@sorttype}%  
  {%  
    \csuse{@glo@sortmacro@ \@glo@sorttype}{\@glo@type}%  
  }%  
  {%  
    \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%  
  }%
```

```
Do the glossary heading and preamble
```

```
  \glossarysection[\glossarytoctitle]{\glossarytitle}%  
  \glossarypreamble
```

The glossary style might use a tabular-like environment, which may cause scoping problems when setting the current letter group. The predefined tabular-like styles don't support letter group headings, but there's nothing to stop the user from defining their own custom style that might, so any redefinition of this command within theglossary will have to be done globally.

```
  \def\@gls@currentlettergroup{}%  
  \begin{theglossary}%  
  \glossaryheader  
  \glsresetentrylist
```

```
Iterate through the entries.
```

```
  \forlistcsloop{\@gls@noidx@do}{@glsref@ \@glo@type}%
```

```
Finally end the glossary and do the postamble:
```

```
  \end{theglossary}%  
  \glossarypostamble  
}%  
{%
```

Add section header if there are actually entries defined in this glossary as the document is likely pending a re-run.

```
  \@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@glstdetoklabel{#1}@loclist}%
```

Does this entry have a parent?

```
\ifglshasparent{#1}%
```

```
{%
```

Has a parent.

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

```

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
\glsgrpskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%

```

```
\global\let\@gls@currentlettergroup\@glo@thislettergrp
```

Do this entry:

```

\ifdefvoid{\@gls@loclist}
  {}%
  \glossentry{#1}{}%
  }%
  {%
    \glossentry{#1}%
    {%
      \glsnoidxprenumberlist{#1}%
      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
  }%
}%
}
}

```

```
\glsnoidxloclist{\list cs}
```

\glsnoidxloclist

Display location list.

```

\newcommand*\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}

```

`\glsnoidxloclisthandler` Handler for location list iterator.

```

\newcommand*\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
    Same as previous location so skip.
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

`\glsnoidxdisplayloclisthandler` Handler for location list iterator when used with `\glsdisplaynumberlist`.

```

\newcommand*\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
    Same as previous location so skip.
  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

`\glsnoidxdisplayloc`

```

\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}
{<location>}

```

Display a location in the location list.

```

\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}

```

`\@gls@reference`

```

\@gls@reference{<type>}{<label>}{<loc>}

```

Identifies that a reference has been used (for use in the aux file). All entries must be defined in the preamble.

```
\ExplSyntaxOn
\newcommand*{\@gls@reference}[3]{%
```

Add to label list

```
\glsdoifexistsorwarn{#2}%
{%
  \__glossaries_add_glsref:nnn { #1 } { #2 } { #3 }
}%
}
\cs_new:Nn \__glossaries_add_glsref:nnn
{
  \__glossaries_add_glsref:nn { #1 } { #2 }
```

Add to location list 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@dotoc@title\relax
}
```

The `toctitle` sets the text used for the relevant entry in the table of contents.

```
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotoc@title\relax
}
```

The `style` key sets the glossary style (but only for the given glossary).

```
\define@key{printgloss}{style}{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossarystyle{#1}}%
}%
}
```

The `numberedsection` key determines if this glossary should be in a numbered section.

```

\define@choicekey{printgloss}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]%
  {false,nolabel,autolabel,nameref}[nolabel]%
{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar{*}%
    \renewcommand*\@@glossaryseclabel{}%
  \or
    \renewcommand*\@@glossarysecstar{}%
    \renewcommand*\@@glossaryseclabel{}%
  \or
    \renewcommand*\@@glossarysecstar{}%
    \renewcommand*\@@glossaryseclabel{\label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar{*}%
    \renewcommand*\@@glossaryseclabel{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{\glsautoprefix\@glo@type}}%
  \fi
}

```

The `nogroupskip` key determines whether or not there should be a vertical gap between glossary groups.

```

\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}

```

The `nopostdot` key has the same effect as the package option of the same name.

```

\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}

```

`\GlsEntryCounterLabelPrefix` Make it easier to redefine the label prefix.

```

\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}

```

The conditionals have been moved inside the appropriate commands to make it easier for the user to redefine them in the preamble and selectively switch the counter display on and off. Previously the helper commands were redefined by the `entrycounter` option, which would counteract any earlier customisation.

The `entrycounter` key is the same as the package option but localised to the current glossary.

```

\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}

```

The `subentrycounter` key is the same as the package option but localised to the current glossary. Note that this doesn't affect the parent/child counter

attributes, which occurs if `subentrycounter` and `entrycounter` package options are set to true.

```
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glssubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
```

The `nonumberlist` key determines if this glossary should have a number list.

```
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
```

The `sort` key sets the glossary sort handler (`\printnoidxglossary` only).

```
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
```

`\@glo@no@assign@sortkey` Issue error if used with `\printglossary`

```
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary \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}%
}
```

`\@glsnonextpages` Suppresses the next number list only. Global assignments required as it may not occur in the same level of grouping as the next numberlist. (For example, if `\glsnonextpages` is place in the entry’s description and 3 column tabular style glossary is used.) `\org@glossaryentrynumbers` needs to be set at the start of each glossary, in the event that `\glossaryentrynumber` is redefined.

```
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}
```

`\@glsnextpages` Activate the next number list only. Global assignments required as it may not occur in the same level of grouping as the next numberlist. (For example, if `\glsnextpages` is place in the entry’s description and 3 column tabular style glossary is used.) `\org@glossaryentrynumbers` needs to be set at the start of each glossary, in the event that `\glossaryentrynumber` is redefined.

```

\newcommand*{\@glsnextpages}{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}

```

`\glsresetentrylist` Resets `\glossaryentrynumbers`

```

\newcommand*{\glsresetentrylist}{%
\global\let\glossaryentrynumbers\org@glossaryentrynumbers}

```

`\glsnonextpages` Outside of `\printglossary` this does nothing.

```

\newcommand*{\glsnonextpages}{}

```

`\glsnextpages` Outside of `\printglossary` this does nothing.

```

\newcommand*{\glsnextpages}{}

```

Process `entrycounter` and then `subentrycounter` options (this ensures the sub-counter can pick up the main counter as the master if required):

```

\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter

```

`\glsresetsubentrycounter` Resets the `glossarysubentry` counter.

```

\newcommand*{\glsresetsubentrycounter}{%
\ifglssubentrycounter
\setcounter{glossarysubentry}{0}%
\fi
}

```

`\glsresetentrycounter` Resets the `glossaryentry` counter.

```

\newcommand*{\glsresetentrycounter}{%
\ifglentrycounter
\setcounter{glossaryentry}{0}%
\fi
}

```

`\glsstepentry` Advance the `glossaryentry` counter if in use. The argument is the label associated with the entry.

```

\newcommand*{\glsstepentry}[1]{%
\ifglentrycounter
\refstepcounter{glossaryentry}%
\label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\fi
}

```

`\glsstepsubentry` Advance the `glossarysubentry` counter if in use. The argument is the label associated with the subentry.

```

\newcommand*{\glsstepsubentry}[1]{%
\ifglssubentrycounter

```

```

        \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
        \refstepcounter{glossarysubentry}%
        \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
    \fi
}

```

`\glsrefentry` Reference the entry or sub-entry counter if in use, otherwise just do `\gls`.

```

\newcommand*\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}

```

`\glssentrycounterlabel` Defines how to display the glossaryentry counter.

```

\newcommand*\glssentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}

```

`\glsssubentrycounterlabel` Defines how to display the glossarysubentry counter.

```

\newcommand*\glsssubentrycounterlabel}{%
  \ifglsssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

`\glsentryitem` Step and display glossaryentry counter, if appropriate.

```

\newcommand*\glsentryitem}[1]{%
  \ifglentrycounter
    \glsstepentry{#1}\glssentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}

```

`\glsssubentryitem` Step and display glossarysubentry counter, if appropriate.

```

\newcommand*\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glsstepsubentry{#1}\glsssubentrycounterlabel
  \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{\glslinkprefix#1}{#2}}
```

As from version 3.08, glossary information is now written to the external files using `\glossentry` and `\subglossentry` instead of `\glossaryentryfield` and `\glossarysubentryfield`. The default definition provides backward compatibility for glossary styles that use the old forms.

`\compatibleglossentry`

```
\compatibleglossentry{<label>}{<page-list>}
```

Removed in v4.50.

`\glossentry`

```
\glossentry{<label>}{<page-list>}
```

```
\newcommand{\glossentry}[2]{\glossentryname{#1} #2}
```

`\glossentryname`

```
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
```

```

        \letcs{\glo@name}{glo@glstetoklabel{#1}@name}%
        \expandafter\glsnamefont\expandafter{\glo@name}%
    }%
}

\Glossentryname
\newcommand*{\Glossentryname}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glsnamefont{\Glsentryname{#1}}%
    }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}

\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}%

```



```

    {%
      \glsuppercase{\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`

```
\compatiblesubglossentry{<level>}{<label>}{<page-list>}
```

Removed in v4.50

`\setglossentrycompatibility`

`\glossaryentryfield`

```
\glossaryentryfield{<label>}{<name>}{<description>}  
{<symbol>}{<page-list>}
```

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

redefined 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 `glsymbols` produces `\glsymbolsgroupname`, the group labelled `glsnumbers` produces `\glsnumbersgroupname` and all the other groups simply produce their label. As mentioned above, the group labels are: `glsymbols`, `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|glsymbols)\Z } { #1 }
{ \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} } }
}
```

`\@gls@noidx@getgrouptitle` Version for the no-indexing app option:

```
\newcommand*\@gls@noidx@getgrouptitle}[2]{%
  \ifcsvoid { l__glossaries_noidx_#1_grouptitle_tl }
  {
    \DTLifint{#1}%
```

```

    {\edef#2{\char#1\relax}}%
    {%
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
    }%
  }
  {
    \tl_set_eq:Nc #2
      { l__glossaries_noidx_#1_grouptitle_tl }
  }
}
\ExplSyntaxOff

```

`\glsgetgrouplabel{<title>}`

This command does the reverse to the previous command. The argument is the group title, and it produces the group label. Note that if you redefine `\glsgetgrouptitle`, you will also need to redefine `\glsgetgrouplabel`.

`\glsgetgrouplabel` This only works for the symbol and number groups, so is best avoided and not documented. However, it has been converted to use L^AT_EX3 commands.

```

\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff

```

The command `\setentrycounter` sets the entry's associated counter (required by `\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{\glsentrycounter\@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{\glsnumber{#1}}}

\hypersf
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}

\hypertt
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}

\hyperbf
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}

\hypermd
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}

\hyperit
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}

\hypersl
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}

\hyperup
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
```


`\hypersc`

```
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
```

`\hyperemph`

```
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
```

2.17 Acronyms

`\oldacronym`

```
\oldacronym[label]{abbrv}{long}{key-val list}
```

This emulates the way the old `glossary` package defined acronyms. It is equivalent to `\newacronym[key-val list]{label}{abbrv}{long}` and it additionally defines the command `\label` which is equivalent to `\gls{label}` (thus `\label` must only contain alphabetical characters). If `\label` is omitted, `\abbrv` is used. This only emulates the syntax of the old `glossary` package. The way the acronyms appear in the list of acronyms is determined by the definition of `\newacronym` and the `glossary` style.

Note that `\label` can't have an optional argument if the `xspace` package is loaded. If `xspace` hasn't been loaded then you can do `\label[insert]` but you can't do `\label[key-val list]`. For example if you define the acronym `svm`, then you can do `\svm['s]` but you can't do `\svm[format=textbf]`. If the `xspace` package is loaded, `\svm['s]` will appear as `svm ['s]` which is unlikely to be the desired result. In this case, you will need to use `\gls` explicitly, e.g. `\gls{svm}['s]`. Note that it is up to the user to load `xspace` if desired.

```
\newcommand\oldacronym[4][\gls@label]{%
  \def\gls@label#2{%
    \newacronym[#4]{#1}{#2}{#3}%
    \ifcsundef{xspace}%
      {%
        \expandafter\edef\csname#1\endcsname{%
          \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
        }%
      }%
    }%
  }%
  \expandafter\edef\csname#1\endcsname{%
    \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
    \noexpand\gls{#1}\noexpand\xspace}%
  }%
}%
}
```

```
\newacronym[key-val list]{label}{abbrv}{long}
```

This is a quick way of defining acronyms, using `\newglossaryentry` with the appropriate values. It sets the glossary type to `\acronymtype` which will be `acronym` if the package option `acronym` has been used, otherwise it will be

the default glossary. Since `\newacronym` merely calls `\newglossaryentry`, the acronym is treated like any other glossary entry.

If you prefer a different format, you can redefine `\newacronym` as required. The optional argument can be used to override any of the settings.

This is just a stub. It's redefined by `\SetGenericNewAcronym` and by `glossaries-extra`. It used to also be redefined by the obsolete acronym styles that have now been removed. If it is used then it means that the acronym style hasn't been initialised.

```
\newacronym
\newcommand{\newacronym}[4] []{%
  \gls@newacronym{#1}{#2}{#3}{#4}%
}
```

```
\gls@newacronym
\newcommand{\gls@newacronym}[4]{%
Setting the acronym style will redefine \newacronym because it calls \SetGenericNewAcronym
but if something weird has happened to alter this, redefine \gls@newacronym
now to prevent an infinite loop.
```

```
  \renewcommand{\gls@newacronym}[4]{%
    \PackageError{glossaries}{incompatible acronym style:
      rollback may be required}%
    {something has gone wrong with \string\newacronym. Try
      \string\usepackage{glossaries}[=v4.49]}%
  }%
  \setacronymstyle{long-short}%
  \newacronym[#1]{#2}{#3}{#4}%
}
```

Set up some convenient short cuts. These need to be changed if `\newacronym` is changed (or if the description key is changed).

```
\acrpluralsuffix Plural suffix used by \newacronym. This just defaults to \glspluralsuffix
but is changed to include \textup if the smallcaps option is used, so that the
suffix doesn't appear in small caps as it doesn't look right. For example, ABCS
looks as though the "s" is part of the acronym, but ABCs looks as though the "s"
is a plural suffix. Since the entire text abcs is set in \textsc, \textup is need
to cancel it out.
```

```
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
```

If `garamondx` has been loaded, need to use `\textulc` instead of `\textup`.

```
\glstextup
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
```

The following are defined for compatibility with version 2.07 and earlier.

```
\glsshortkey
\newcommand*{\glsshortkey}{short}
```

```

\glsshortpluralkey
    \newcommand*\glsshortpluralkey}{shortplural}

\glslongkey
    \newcommand*\glslongkey}{long}

\glslongpluralkey
    \newcommand*\glslongpluralkey}{longplural}

```

Using `\acrlinkfullformat` and `\acrfullformat` is now deprecated. However, these commands are still used in the initial definition of `\acrfull` and `\glsentryfull`. The acronym style now redefines `\glsentryfull`, and `\acrfull` now uses `\acrfullfmt`, which is redefined by the acronym style. Similar for the case and plural variants. This means that `\acrlinkfullformat` and `\acrfullformat` should now longer be used with v4.50, which will ensure that `\setacronymstyle` is used on the first instance of `\newacronym` if a style hasn't been set.

Outlier cases may occur where someone has defined an entry explicitly with `\newglossaryentry` and set the long and short keys and is using `\glsentryfull` or `\acrfull` without a style set. So, for now, the deprecated `\acrlinkfullformat` and `\acrfullformat` are retained even though the remaining obsolete acronym code has been removed. However, they may be removed in a future version.

```

\acrlinkfullformat Format for full links like \acrfull. Syntax: \acrlinkfullformat{<long cs>}
                  {<short cs>}{<options>}{<label>}{<insert>} Deprecated.
                  \newcommand{\acrlinkfullformat}[5]{%
                    \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
                  }

\acrfullformat Default full form is <long> (<short>). Deprecated.
               \newcommand{\acrfullformat}[2]{#1\glsspace(#2)}

\glsspace Robust space to ensure it's written to the .glsdefs file.
           \newrobustcmd{\glsspace}{\space}

```

The following `\acrfullfmt` set of commands are used by the corresponding `\acrfull` command, and are redefined by the acronym style to match the style's full form.

```

\acrfullfmt No case change full format. This default will be overridden by the acronym
            style, so this definition shouldn't actually be used any more, but it still needs
            to be defined so the it doesn't cause a problem when acronym styles redefine
            it. This is initialised to use the deprecated \acrlinkfullformat but will be
            redefined by the acronym style when it's set.

            \newcommand*\acrfullfmt}[3]{%
              \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
            }

```

```

\Acrfullfmt Sentence case full format.
  \newcommand*\Acrfullfmt}[3]{%
    \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
  }

\ACRfullfmt All upper case full format.
  \newcommand*\ACRfullfmt}[3]{%
    \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
  }

\acrfullplfmt No case change plural full format.
  \newcommand*\acrfullplfmt}[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
  }

\Acrfullplfmt Sentence case plural full format.
  \newcommand*\Acrfullplfmt}[3]{%
    \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
  }

\ACRfullplfmt All upper case plural full format.
  \newcommand*\ACRfullplfmt}[3]{%
    \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
  }

\acrfull Full form of the acronym.
  \newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}

  \newcommand*\ns@acrfull[2][ ]{%
    \new@ifnextchar[{\@acrfull{#1}{#2}}%
      {\@acrfull{#1}{#2}[ ]}%
  }

\@acrfull Low-level macro:
  \def\@acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
    \acrfullfmt{#1}{#2}{#3}%
  }

  Default format for full acronym

\Acrfull
  \newrobustcmd*\Acrfull{\@gls@hyp@opt\ns@Acrfull}
Add sentence case mapping.
  \glsmfuaddmap{\acrfull}{\Acrfull}

```

```

\newcommand*\ns@Acrfull[2] []{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@Acrfull#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```

\Acrfullfmt{#1}{#2}{#3}%
}

```

`\ACRfull`

```
\newrobustcmd*\@ACRfull{\@gls@hyp@opt\ns@ACRfull}
```

```

\newcommand*\ns@ACRfull[2] []{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@ACRfull#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```

\ACRfullfmt{#1}{#2}{#3}%
}

```

Plural:

`\acrfullpl`

```
\newrobustcmd*\@acrfullpl{\@gls@hyp@opt\ns@acrfullpl}
```

```

\newcommand*\ns@acrfullpl[2] []{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```

\acrfullplfmt{#1}{#2}{#3}%
}

```

`\Acrfullpl`

```
\newrobustcmd*\@Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
```

```

\newcommand*\ns@Acrfullpl[2] []{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@Acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\Acrfullplfmt{#1}{#2}{#3}%  
}
```

`\ACRfullpl`

```
\newrobustcmd*\@ACRfullpl{\@gls@hyp@opt\ns@ACRfullpl}
```

```
\newcommand*\ns@ACRfullpl[2][]{%  
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%  
    {\@ACRfullpl{#1}{#2}[]}%  
}
```

Low-level macro:

```
\def\@ACRfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\ACRfullplfmt{#1}{#2}{#3}%  
}
```

2.18 Predefined acronym styles

`\acronymfont` Used by the acronym styles to apply a font change to the short form.

```
\newcommand{\acronymfont}[1]{#1}
```

`\firstacronymfont` Used by the acronym styles to apply a font change to the short form on first use.

```
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
```

`\acrnameformat` The styles that allow an additional description use `\acrnameformat{<short>}{<long>}` to determine what information is displayed in the name.

```
\newcommand*\acrnameformat[2]{\acronymfont{#1}}
```

Define some tokens used by `\newacronym`:

`\glskeylisttok`

```
\newtoks\glskeylisttok
```

`\glslabeltok`

```
\newtoks\glslabeltok
```

`\glsshorttok`

```
\newtoks\glsshorttok
```

`\glslongtok`

```
\newtoks\glslongtok
```

`\newacronymhook` Provide a hook for `\newacronym`:

```
\newcommand*\newacronymhook{}
```

`\SetGenericNewAcronym` New improved version of setting the acronym style.

```
\newcommand*\SetGenericNewAcronym{%
```

Change the behaviour of `\Glsentryname` to workaround expansion issues that cause a problem for `\makefirstuc`

```
\let\@Gls@entryname\@Gls@acrentryname
```

Change the way acronyms are defined:

```
\renewcommand{\newacronym}[4][]{%
\ifdefempty{\@glsacronymlists}%
{%
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
}%
}%
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\protected@edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\expandonce{\acronymentry{##2}}},%
sort={\acronymstort{\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}{%
    \glsupercase{\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{<style name>}{<entry format
definition>}{<display definitions>}

```

`\newacronymstyle`

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}{\glsacrfmt}{\glsacrfmt}%
}%
%
```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \glsentrylong{##1}##2\space
  (\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \Glsentrylong{##1}##2\space
  (\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{##1}##2\space
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\space
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

long-sp-short Similar to the previous style but allows the space between the long and short form to be customized.

```

\newacronymstyle{long-sp-short}%
{%

```

Check for long form in case this is a mixed glossary.

```

  \ifglsashaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
}

```

```

\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}

```

`\glsacspace` Space between long and short form for the above style. This uses a non-breakable space if the short form is less than 3em, otherwise it uses a regular space. It's unlikely that the short form will contain any instances of `\gls` so this just uses `\settowidth` instead of `\glsmeasurewidth`.

```

\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}

```

`short-long` $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```

\newacronymstyle{short-long}%
{%

```

Check for long form in case this is a mixed glossary.

```

  \ifglsashaslong{\glslabel}{\glsacgenacfmt}{\glsacgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%

\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}

```

`long-sc-short` $\langle long \rangle$ ($\textsc{\langle short \rangle}$) acronym style.

```

\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}

```

long-sm-short $\langle long \rangle$ ($\text{\textsmaller}\langle short \rangle$) acronym style.

```

\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

sc-short-long $\langle short \rangle$ ($\text{\textsc}\langle long \rangle$) acronym style.

```

\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}

```

sm-short-long $\langle short \rangle$ ($\text{\textsmaller}\langle long \rangle$) acronym style.

```

\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

long-short-desc $\langle long \rangle$ ($\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%

```

```

    {%
      \GlsUseAcrStyleDefs{long-short}%
      \renewcommand*\GenericAcronymFields{}%
      \renewcommand*\acronymsort}[2]{##2}%
      \renewcommand*\acronymentry}[1]{%
        \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})%
      }
}

long-sp-short-desc <long> ({<short>}) acronym style that has an accompanying description (which
the user needs to supply). The space between the long and short form is given
by \glsacspace.
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \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` $\langle short \rangle$ ($\{\langle long \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

`sc-short-long-desc` $\langle long \rangle$ ($\backslash\text{textsc}\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

`sm-short-long-desc` $\langle long \rangle$ ($\backslash\text{textsmaller}\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

`dua` $\langle long \rangle$ 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:
    \glentrylongpl{\glslabel}\glsinsert
    }%
    {%
Plural form, make first letter upper case:
    \Glentrylongpl{\glslabel}\glsinsert
    }%
    {%
Plural form, all caps:
    \glssupercase
    {\glentrylongpl{\glslabel}\glsinsert}%
    }%
    }%
    {%
Singular form
    \glscapscase
    {%
Singular form, don't adjust case:
    \glentrylong{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, make first letter upper case:
    \Glentrylong{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, all caps:
    \glssupercase
    {\glentrylong{\glslabel}\glsinsert}%
    }%
    }%
    }%
    }%
Not an acronym:
    \glsgenentryfmt
    }%
    }%
    {\glscustomtext\glsinsert}%

```

```

}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}}%

\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshorttpl{##2}})}%

\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshorttpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshorttpl{##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{\glsentryshorttpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshorttpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%

```



```

\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%

\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%

```

footnote *<short>*\footnote{*<long>*} acronym style.

```

\newacronymstyle{footnote}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

```

Need to ensure hyperlinks are switched off on first use:

```

\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%

```

Don't use footnotes for \acrfull:

```

\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
}

```

```

\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsupercase{\acronymfont{\glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}}%

```

Similarly for `\glsentryfull` etc:

```

\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}

```

`footnote-sc` `\textsc{<short>}\footnote{<long>}` acronym style.

```

\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%

```

`footnote-sm` `\textsmaller{<short>}\footnote{<long>}` acronym style.

```

\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%

```

`footnote-desc` `<short>\footnote{<long>}` acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{footnote-desc}%
{%

```

```

    \GlsUseAcrEntryDispStyle{footnote}%
  }%
  {%
    \GlsUseAcrStyleDefs{footnote}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
    \renewcommand*\acronymentry}[1]{%
      \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

```
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
```

`\SetDefaultAcronymDisplayStyle` Sets the default acronym display style for given glossary. Used by `glossaries-extra` when restoring `\newacronym`.

```
\newcommand*\SetDefaultAcronymDisplayStyle[1]{%
  \defglsentryfmt[#1]{\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.

`\noexpand@assign@default@plurals` `\@@gls@noexpand@assign@default@plurals` was used in `\DefaultNewAcronymDef`. Added in v4.47 to fix bug #162 (<https://www.dickimaw-books.com/bugtracker.php?key=162>). Removed in v4.50 along with the deprecated styles.

`\@@gls@@default@acr@plural`

`\gls@@default@acr@firstplural`

`\gls@@default@acr@descplural`

`\SetDefaultAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\acrfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*\acrfootnote[3]{\acrlinkfootnote{#1}{#2}{#3}}
```

<code>\acrlinkfootnote</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*{\acrlinkfootnote}[3]{% \footnote{\glslink[#1]{#2}{#3}}% }</pre>
<code>\acrnoflinkfootnote</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*{\acrnoflinkfootnote}[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 <code>text</code> 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>ShortDescriptionDUAAcronymStyle</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 <code>description</code> is used, but not <code>dua</code> or <code>footnote</code> . Store long form in first key and short form in <code>text</code> and <code>symbol</code> 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.

FootnoteAcronymDisplayStyle Sets the acronym display style for given glossary with footnote setting (but not description or dua). Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\FootnoteNewAcronymDef Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\SetFootnoteAcronymStyle If footnote package option is specified, set the first use to append the long form (stored in `description`) as a footnote. Use the `description` key to store the long form. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\glsdoparenifnotempty Do a space followed by the argument if the argument doesn't expand to empty or `\relax`. If argument isn't empty (or `\relax`), apply the macro to it given in the second argument.

```

\newrobustcmd*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}

```

SetSmallAcronymDisplayStyle Sets the acronym display style for given glossary where neither footnote nor description is required, but smallcaps or smaller specified. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\SmallNewAcronymDef Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\SetSmallAcronymStyle Neither footnote nor description required, but smallcaps or smaller specified. Use the `symbol` key to store the short form and `first` to store the long form. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\SetDUADisplayStyle Sets the acronym display style for given glossary with dua setting. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\DUANewAcronymDef Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

\SetDUASyle Always expand acronyms. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`@gls@deprecated@acr@warning`

```
\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
```

`\SetAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

Allow user to define their own custom acronyms. (For compatibility with versions before v3.0, the short form is stored in the `user1` key, the plural short form is stored in the `user2` key, the long form is stored in the `user3` key and the plural long form is stored in the `user4` key.) Defaults to displaying only the acronym with the long form as the description.

`\SetCustomDisplayStyle` Sets the acronym display style. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\CustomAcronymFields` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\CustomNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetCustomStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

2.19 Predefined Glossary Styles

The glossaries bundle comes with some predefined glossary styles. These need to be loaded now for the `style` option to use them.

First, the glossary hyper-navigation commands need to be loaded.

```
\RequirePackage{glossary-hypernav}
```

The styles that use list-like environments. These are not loaded if the `nolist` option is used:

```
\@gls@loadlist
```

The styles that use the `longtable` environment. These are not loaded if the `nolong` package option is used.

```
\@gls@loadlong
```

The styles that use the `supertabular` environment. These are not loaded if the `nosuper` package option is used or if the `supertabular` package isn't installed.

```
\@gls@loadsuper
```

The tree-like styles. These are not loaded if the `notree` package option is used.

```
\@gls@loadtree
```


The default glossary style is set according to the `style` package option, but can be overridden by `\glossarystyle`. The required style must be defined at this point.

```
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glossary@default@style}
\fi
```

2.20 Debugging Commands

`\showgloparent`

```
\showgloparent{<label>}
```

```
\newcommand*\showgloparent[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@parent\endcsname
}
```

`\showglolevel`

```
\showglolevel{<label>}
```

```
\newcommand*\showglolevel[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@level\endcsname
}
```

`\showglotext`

```
\showglotext{<label>}
```

```
\newcommand*\showglotext[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
}
```

`\showgloplural`

```
\showgloplural{<label>}
```

```
\newcommand*\showgloplural[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
}
```

`\showglofirst`

```
\showglofirst{<label>}
```

```
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo\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}@ftype\endcsname
}
```

\showglocounter

```
\showglocounter{\label}
```

```
\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@counter\endcsname
}
```

\showglouseri

```
\showglouseri{\label}
```

```
\newcommand*\showglouseri}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@useri\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@glstdetoklabel{#1}@useriii\endcsname
}

\showglouseriv \showglouseriv{<label>}

\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriv\endcsname
}

\showglouserv \showglouserv{<label>}

\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userv\endcsname
}

\showglouservi \showglouservi{<label>}

\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@uservi\endcsname
}

\showgloname \showgloname{<label>}

\newcommand*{\showgloname}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@name\endcsname
}

\showglodesc \showglodesc{<label>}

\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname
}

```

`\showglodescplural`

```
\showglodescplural{<label>}
```

```
\newcommand*{\showglodescplural}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname  
}
```

`\showglosort`

```
\showglosort{<label>}
```

NB this may not be the same as the sortvalue internal field.

```
\newcommand*{\showglosort}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname  
}
```

`\showglosymbol`

```
\showglosymbol{<label>}
```

```
\newcommand*{\showglosymbol}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname  
}
```

`\showglosymbolplural`

```
\showglosymbolplural{<label>}
```

```
\newcommand*{\showglosymbolplural}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname  
}
```

`\showgloshort`

```
\showgloshort{<label>}
```

```
\newcommand*{\showgloshort}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname  
}
```

`\showglolong`

```
\showglolong{<label>}
```

```
\newcommand*{\showglolong}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@long\endcsname  
}
```

`\showgloindex`

```
\showgloindex{<label>}
```

```
\newcommand*{\showgloindex}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname  
}
```

`\showgloflag`

```
\showgloflag{<label>}
```

```
\newcommand*{\showgloflag}[1]{%  
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname  
}
```

`\showgloloclist`

```
\showgloloclist{<label>}
```

```
\newcommand*{\showgloloclist}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname  
}
```

`\showglofield`

```
\showglofield{<label>}{<field>}
```

```
\newcommand*{\showglofield}[2]{%  
  \csshow{glo@\glsdetoklabel{#1}@#2}%  
}
```

`\showacronymlists`

```
\showacronymlists
```

Show list of glossaries that have been flagged as a list of acronyms.

```
\newcommand*{\showacronymlists}{%  
  \show\@glsacronymlists  
}
```

`\showglossaries`

```
\showglossaries
```

Show list of defined glossaries.

```
\newcommand*{\showglossaries}{%  
  \show\@glo@types  
}
```

`\showglossaryin`

```
\showglossaryin{<glossary-label>}
```

Show the ‘in’ extension for the given glossary.

```
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1in\endcsname
}
```

`\showglossaryout`

```
\showglossaryout{<glossary-label>}
```

Show the ‘out’ extension for the given glossary.

```
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1out\endcsname
}
```

`\showglossarytitle`

```
\showglossarytitle{<glossary-label>}
```

Show the title for the given glossary.

```
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1title\endcsname
}
```

`\showglossarycounter`

```
\showglossarycounter{<glossary-label>}
```

Show the counter for the given glossary.

```
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1counter\endcsname
}
```

`\showglossaryentries`

```
\showglossaryentries{<glossary-label>}
```

Show the list of entry labels for the given glossary.

```
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}
```

2.21 Compatibility with version 2.07 and below

The backward-compatibility options have been removed in version 4.50. Use the rollback feature instead if backward-compatibility is required. Bear in mind that v2.07 was released 2010-07-10 and v3.07 was released 2013-07-05. Any old documents from that long ago will probably have problems with recent kernels.

In order to fix some bugs in v3.0, it was necessary to change the way information is written to the `glo` file, which also meant a change in the format of the Xindy style file. The compatibility option is meant for documents that use a customised Xindy style file with `\noist`. With the compatibility option, hopefully `xindy` will still be able to process the old document, but the bugs will remain. The issues in versions 2.07 and below:

- With `xindy`, the counter used by the entry was hard-coded into the Xindy style file. This meant that you couldn't use the `counter` to swap counters.
- With both `xindy` and `makeindex`, if used with `hyperref` and `\theH{counter}` was different to `\thecounter`, the link in the location number would be undefined.

3 Glossary Styles

3.1 Glossary hyper-navigation definitions (`glossary-hypernav.sty`)

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-hypernav-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-hypernav-2021-11-01.sty}
\DeclareRelease{v4.52}{2022-11-03}{glossary-hypernav-2022-11-03.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Package Definition:

```
\ProvidesPackage{glossary-hypernav}[2025/04/14 v4.6 (NLCT)]
```

The commands defined in this package are provided to help navigate around the groups within a glossary (see [subsection 2.16.](#)) `\printglossary` (and `\printglossaries`) set `\@glo@type` to the label of the current glossary. This is used to create a unique `hypertarget` in the event of multiple glossaries.

```
\glsnavhyperlink[<type>]{<label>}{<text>}
```

This command makes `<text>` a hyperlink to the glossary group whose label is given by `<label>` for the glossary given by `<type>`.

`\glsnavhyperlink` The optional argument is the glossary type and can be used to make the group label unique in case the same group occurs in multiple glossaries. The optional argument was `\@glo@type` but is now `\currentglossary` because that's less likely to change within the glossary. It's unlikely that the optional argument will be used to set it to anything other than `\currentglossary` but it makes it possible to hyperlink to the group from outside of the glossary.

```
\newcommand*\glsnavhyperlink[3][\currentglossary]{%
\protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
\@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
```

`\glsnavhyperlinkname` Expands to the hypertarget name. The first argument is the glossary type. The second argument is the group label.

```
\newcommand*{\glsnavhyperlinkname}[2]{glsn:#1@#2}
```

Version 4.53: reimplemented list of hypertarget groups as a sequence. Previously, the targets were stored as comma-separated lists in `\@gls@hypergrouplist@<glossary>` (where `<glossary>` is the glossary label). They are now stored in a sequence `\g_glossaries_<glossary>_hypergroups_seq`.

```
\ExplSyntaxOn
```

```
\glossaries_add_hypergroup:nn{<glossary>}{<target>}
```

Add a target to the sequence unless it's already contained in it:

```
\cs_new:Npn \glossaries_add_hypergroup:nn #1 #2
{
  \seq_if_exist:cTF
  { g_glossaries_ #1 _hypergroups_seq }
  {
    \seq_if_in:cnF
    { g_glossaries_ #1 _hypergroups_seq } { #2 }
    {
      \seq_gput_right:cn
      { g_glossaries_ #1 _hypergroups_seq }
      { #2 }
    }
  }
}
\cs_generate_variant:Nn \glossaries_add_hypergroup:nn { nx }
```

Check if the given target has been added to the sequence to avoid duplicates.

```
\cs_new:Npn \glossaries_if_hypergroup:nnTF #1 #2 #3 #4
{
  \seq_if_exist:cTF
  { g_glossaries_ #1 _hypergroups_seq }
  {
    \seq_if_in:cnTF
    { g_glossaries_ #1 _hypergroups_seq } { #2 }
    { #3 } { #4 }
  }
  { #4 }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnTF { nxTF }
```


Only true part:

```
\cs_new:Npn \glossaries_if_hypergroup:nnT #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { #3 } { }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnT { nxT }
```

Only false part:

```
\cs_new:Npn \glossaries_if_hypergroup:nnF #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { } { #3 }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnF { nxF }
```

```
\glsnavhypertarget[type]{label}{text}
```

`\glsnavhypertarget`

This command makes *text* a hypertarget for the glossary group whose label is given by *label* in the glossary given by *type*. If *type* is omitted, `\@gls@type` is used which is set by `\printglossary` to the current glossary label. The optional argument complicates things for `glossaries-extra`.

```
\newcommand*\glsnavhypertarget{\protect\@gls@navhypertarget}
```

`\@gls@navhypertarget`

```
\newcommand*\@gls@navhypertarget[3][\currentglossary]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
```

`\glsnavhypergroupdotarget` Provided to make it easier for `glossaries-extra` to adjust the target.

```
\newcommand{\glsnavhypergroupdotarget}[3]{
  \@gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
```

The actual code is now in an internal command that doesn't have an optional argument, which makes it easier to save and restore the original behaviour.

`\@glsnavhypertarget`

```
\newcommand*\@glsnavhypertarget[3]{%
```

Add this group to the aux file for re-run check.

```
\protected@write\@auxout{\string\@gls@hypergroup{#1}{#2}}%
```

Add the target.

```
\glsnavhypergroupdotarget { #1 } { #2 } { #3 }
```

Check list of known groups to determine if a re-run is required.

```
\glossaries_if_hypergroup:nxF { #1 } { #2 }
{
```

This group was not included in the list, so issue a warning.

```

\GlossariesWarningNoLine{Navigation ~ panel ~
  for ~ glossary ~ type ~ '#1'^^Jmissing ~ group ~ '#2'}%
\gdef\gls@hypergrouprerun{%
  \GlossariesWarningNoLine{Navigation ~ panel ~
    has ~ changed. ~ Rerun ~ LaTeX}}%
}
}

```

`\gls@hypergrouprerun` Give a warning at the end if re-run required

```

\let\gls@hypergrouprerun\relax
\AtEndDocument{\gls@hypergrouprerun}

```

`\@gls@hypergroup` This adds to (or creates) the command `\@gls@hypergrouplist@<glossary type>` which lists all groups for a given glossary, so that the navigation bar only contains those groups that are present. However it requires at least 2 runs to ensure the information is up-to-date.

```

\newcommand*{\@gls@hypergroup}[2]{
  \glossaries_add_hypergroup:nx { #1 } { #2 }
}

```

The `\glsnavigation` command displays a simple glossary group navigation. The symbol and number elements are defined separately, so that they can be suppressed if need be. (In earlier versions this command will produce a link to all 28 groups, but some groups may not be defined if there are groups that do not contain any terms, in which case you will get an undefined hyperlink warning. Version 1.14 changed this to only use labels for groups that are present.) Now for the whole navigation bit:

`\glsnavigation`

```

\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \seq_map_inline:cn { g_glossaries_ \currentglossary _hypergroups_seq }
  {
    \@gls@between
    \glsnavigationitem { ##1 }
    \let\@gls@between\glshypernavsep
  }%
}

```

`\glsnavigationitem`

```

\newcommand{\glsnavigationitem}[1]{%
  \@gls@getgrouptitle { #1 } { \@gls@grptitle }
  \exp_args:Nno \glsnavhyperlink { #1 } { \@gls@grptitle }
}

```

`\glshypernavsep` Separator for the hyper navigation bar.

```

\newcommand*{\glshypernavsep}{\space\textbar\space}

```

The `\glssymbolnav` produces a simple navigation set of links for just the symbol and number groups. This used to be used at the start of `\glsnavigation`. This command is no longer needed.

```
\glssymbolnav
\newcommand*\glssymbolnav{%
  \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}
\ExplSyntaxOff
```

3.2 In-line Style (`glossary-inline.sty`)

This defines an in-line style where the entries are comma-separated with just the name and description displayed.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-inline-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-inline-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Declare package:

```
\ProvidesPackage{glossary-inline}[2025/04/14 v4.6 (NLCT)]
```

`inline` Define the inline style.

```
\newglossarystyle{inline}{%
```

Start of glossary sets up first empty separator between entries. (This is then changed by `\glossentry`)

```
\renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{}%
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{}%
  }%
  {\glspostinline}%
```

No header:

```
\renewcommand*\glossaryheader{}%
```

No group headings (if heading is required, add `\glsinlinedopostchild` to start definition in case heading follows a child entry):

```
\renewcommand*\glsgroupheading}[1]{}%
```

Just display separator followed by name and description:

```
\renewcommand{\glossentry}[2]{}%
  \glsinlinedopostchild
```

```

\gls@inlinesep
\glsentryitem{##1}%
\glsinlinenameformat{##1}{%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
{%
  \ifglshasdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\glsinlineifhaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%

```

Sub-entries display description:

```

\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glssubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%

```

Nothing special between groups:

```

\renewcommand*{\glsgroupskip}{}%
}

```

`\glsinlineifhaschildren`

```

\newcommand{\glsinlineifhaschildren}[3]{%
  \ifglshaschildren{#1}{#2}{#3}%
}

```

`\glsinlinedopostchild`

```

\newcommand*\glsinlinedopostchild{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}

```

`\glsinlineseparator` Separator to use between entries.

```
\newcommand*\glsinlineseparator}{;\space}
```

`\glsinlinesubseparator` Separator to use between sub-entries.

```
\newcommand*\glsinlinesubseparator}{,\space}
```

`\glsinlineparentchildseparator` Separator to use between parent and children.

```
\newcommand*\glsinlineparentchildseparator}{:\space}
```

`\glsinlinepostchild` Hook to use between child and next entry

```
\newcommand*\glsinlinepostchild}{}
```

`\glspostinline` Terminator for inline glossary.

```
\newcommand*\glspostinline}{\glspostdescription\space}
```

`\glsinlinenameformat` Formats the name of the entry (first argument label, second argument name):

```
\newcommand*\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
```

`\glsinlinedescformat` Formats the entry's description, symbol and location list:

```
\newcommand*\glsinlinedescformat}[3]{\space#1}
```

`\glsinlineemptydescformat` Formats the entry's symbol and location list when the description is empty:

```
\newcommand*\glsinlineemptydescformat}[2]{}
```

`\glsinlinesubnameformat` Formats the name of the subentry (first argument label, second argument name):

```
\newcommand*\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
```

`\glsinlinesubdescformat` Formats the subentry's description, symbol and location list:

```
\newcommand*\glsinlinesubdescformat}[3]{#1}
```

3.3 List Style (`glossary-list.sty`)

The `glossary-list` style file defines glossary styles that use the `description` environment. Note that since the entry name is placed in the optional argument to the `\item` command, it will appear in a bold font by default.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-list-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-list-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}

```

Declare package:

```
\ProvidesPackage{glossary-list}[2025/04/14 v4.6 (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
\glsentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
```

altlistgroup The altlistgroup glossary style is like the altlist style, but the glossary groups have headings.

```
\newglossarystyle{altlistgroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Each group has a heading:

```
\renewcommand*{\glsgroupheading}[1]{%
\item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
```

altlisthypergroup The altlisthypergroup glossary style is like the altlistgroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{altlisthypergroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%
\glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a hypertext:

```
\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[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
```

3.4 Glossary Styles using `longtable` (`glossary-long.sty`)

The glossary styles defined in the `glossary-long` package used the `longtable` environment in the glossary.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-long-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-long-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2025/04/14 v4.6 (NLCT)]
```

Requires the longtable package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. (There's a chance that the user may specify `nolong` and then load `glossary-long` later, in which case `\glsdescwidth` may have already been defined by `glossary-super`. The same goes for `\glspagelistwidth`.)

```
\@ifundefined{glsdescwidth}{%  
  \newlength{glsdescwidth}  
  \setlength{glsdescwidth}{0.6\hsize}  
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column.

```
\@ifundefined{glspagelistwidth}{%  
  \newlength{glspagelistwidth}  
  \setlength{glspagelistwidth}{0.1\hsize}  
}{}
```

`long` The long glossary style command which uses the `longtable` environment:

```
\newglossarystyle{long}{%
```

Use `longtable` with two columns:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}[lp{glsdescwidth}]}%  
  {\end{longtable}}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%  
  \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline  
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip}{}%
\else
  \renewcommand*\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

`longborder` The `longborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longborder}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`longheader` The `longheader` style is like the `long` style but with a header:

```
\newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Set the table's header:

```
\renewcommand*\glossaryheader}{%
  \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
```

`longheaderborder` The `longheaderborder` style is like the `long` style but with a header and border:

```
\newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
\setglossarystyle{longborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*\glossaryheader}{%
  \hline\bfseries \entryname & \bfseries
  \descriptionname\tabularnewline\hline
  \endhead
  \hline\endfoot}%
}
```

`long3col` The `long3col` style is like `long` but with 3 columns

```
\newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[lp{\glsdescwidth}p{\glspagelistwidth}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \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
}
```

`long3colborder` The `long3colborder` style is like the `long3col` style but with a border:

```
\newglossarystyle{long3colborder}{}%
```

Base it on the `glostylelong3col` style:

```
\setglossarystyle{long3col}{}%
```

Use a `longtable` with 3 columns with vertical lines around them:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`long3colheader` The `long3colheader` style is like `long3col` but with a header row:

```
\newglossarystyle{long3colheader}{%
Base it on the glostylelong3col style:
\setglossarystyle{long3col}%
Set the table's header:
\renewcommand*{\glossaryheader}{%
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long3colheaderborder` The `long3colheaderborder` style is like the above but with a border

```
\newglossarystyle{long3colheaderborder}{%
Base it on the glostylelong3colborder style:
\setglossarystyle{long3colborder}%
Set the table's header and add horizontal line at table's foot:
\renewcommand*{\glossaryheader}{%
\hline
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}
```

`long4col` The `long4col` style has four columns where the third column contains the value of the associated symbol key.

```
\newglossarystyle{long4col}{%
Use a longtable with 4 columns:
\renewenvironment{theglossary}%
{\begin{longtable}{llll}}%
{\end{longtable}}%
No table header:
\renewcommand*{\glossaryheader}{}%
No group headings:
\renewcommand*{\glsgroupheading}[1]{}%
Main (level 0) entries on a single row (name in first column, description in
second column, symbol in third column, page list in last column):
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} &
##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & & \tabularnewline}%
\fi
}
```

`long4colheader` The `long4colheader` style is like `long4col` but with a header row.

```
\newglossarystyle{long4colheader}{%
Base it on the glostylelong4col style:
  \setglossarystyle{long4col}%
Table has a header:
  \renewcommand*\glossaryheader{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long4colborder` The `long4colborder` style is like `long4col` but with a border.

```
\newglossarystyle{long4colborder}{%
Base it on the glostylelong4col style:
  \setglossarystyle{long4col}%
Use a longtable with 4 columns surrounded by vertical lines:
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
Add horizontal lines to the head and foot of the table:
  \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
}
```

`long4colheaderborder` The `long4colheaderborder` style is like the above but with a border.

```
\newglossarystyle{long4colheaderborder}{%
Base it on the glostylelong4col style:
  \setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns surrounded by vertical lines:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{|l|l|l|l|}}%  
  {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*{\glossaryheader}{%  
  \hline\bfseries\entryname&\bfseries\descriptionname&  
  \bfseries \symbolname&  
  \bfseries\pagelistname\tabularnewline\hline\endhead  
  \hline\endfoot}%  
}
```

`altlong4col` The `altlong4col` style is like the `long4col` style but can have multiline descriptions and page lists.

```
\newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```
\newglossarystyle{altlong4colheader}{%
```

Base it on the `glostylelong4colheader` style:

```
\setglossarystyle{long4colheader}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```
\newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
\setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%  
  {\end{longtable}}%  
}
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `glostylolong4colheaderborder` style:

```
\setglossarystyle{long4colheaderborder}{%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}{%
  {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\glspagerlistwidth}|}%
  {\end{longtable}}}%
}
```

3.5 Glossary Styles using `longtable` and `booktabs` (`glossary-longbooktabs.sty`)

The styles here are based on David Carlisle's patch at <http://tex.stackexchange.com/a/56890>

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-longbooktabs-2020-03-19.sty}
```

```
\DeclareRelease{v4.49}{2021-11-01}{glossary-longbooktabs-2021-11-01.sty}
```

```
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2025/04/14 v4.6 (NLCT)]
```

Requires `booktabs` package:

```
\RequirePackage{booktabs}
```

and the base packages for long styles:

```
\RequirePackage{glossary-long}
```

```
\RequirePackage{glossary-longragged}
```

(`longtable` and `array` loaded by those packages).

`long-booktabs` The `long-booktabs` style is similar to the `longheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

As with the `longheader` style, use the `long` style as a base.

```
\setglossarystyle{long}{%
```


Add a header with rules.

```
\renewcommand*\glossaryheader}{%
  \toprule \bfseries \entryname & \bfseries
  \descriptionname\tabularnewline\midrule\endhead
  \bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip}{}%
\else
  \renewcommand*\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`long3col-booktabs` The `long3col-booktabs` style is similar to the `long3colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long3col` style as a base.

```
\setglossarystyle{long3col}%
```

Add a header with rules.

```
\renewcommand*\glossaryheader}{%
  \toprule \bfseries \entryname &
  \bfseries \descriptionname &
  \bfseries \pagelistname
  \tabularnewline\midrule\endhead
  \bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip}{}%
\else
  \renewcommand*\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`long4col-booktabs` The `long4col-booktabs` style is similar to the `long4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long4col` style as a base.

```
\setglossarystyle{long4col}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
  \toprule \bfseries \entryname &
  \bfseries \descriptionname &
  \bfseries \symbolname &
  \bfseries \pagelistname
  \tabularnewline\midrule\endhead
  \bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`altlong4col-booktabs` The `altlong4col-booktabs` style is similar to the `altlong4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{altlong4col-booktabs}{%
```

The patch `\glspatchLToutput` is already applied in `long4col-booktabs` and so doesn't need to be here.

```
\glspatchLToutput
```

Use the `long4col-booktabs` style as a base.

```
\setglossarystyle{long4col-booktabs}%
```

Change the column specifications:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}%
  {\end{longtable}}%
}
```

Ragged styles.

`longragged-booktabs` The `longragged-booktabs` style is similar to the `longragged` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long-booktabs style as a base.

```
\setglossarystyle{long-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}}%  
  {\end{longtable}}%  
}
```

longragged3col-booktabs The longragged3col-booktabs style is similar to the longragged3col style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{longragged3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long3col-booktabs style as a base.

```
\setglossarystyle{long3col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}%  
    >{\raggedright}p{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

altlongragged4col-booktabs The altlongragged4col-booktabs style is similar to the altlongragged4col style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{altlongragged4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the altlong4col-booktabs style as a base.

```
\setglossarystyle{altlong4col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}1%  
    >{\raggedright}p{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

```

\glsLTpenaltycheck
  \newcommand*{\glsLTpenaltycheck}{%
    \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
  }

```

```

\glspenaltygroupskip
  \newcommand{\glspenaltygroupskip}{%
    \noalign{\penalty-50\vskip\normalbaselineskip}}

```

```

\glsrestoreLToutput Provide a way of restoring \LT@output for the user.
  \let\@gls@org@LT@output\LT@output
  \newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}

```

This is David's patch, but I've replaced the hard-coded values with `\glsLTpenaltycheck` to make it easier to adjust.

```

\glspatchLToutput
  \newcommand*{\glspatchLToutput}{%
    \renewcommand*{\LT@output}{%
      \ifnum\outputpenalty <-\@Mi
        \ifnum\outputpenalty > -\LT@end@pen
          \LT@err{floats and marginpars not allowed in a longtable}\@ehc
        \else
          \setbox\z@\vbox{\unvbox\@cclv}%
          \ifdim \ht\LT@lastfoot>\ht\LT@foot
            \dimen@\pagegoal
            \advance\dimen@-\ht\LT@lastfoot
            \ifdim\dimen@<\ht\z@
              \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
              \@makecol
              \@outputpage
              \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
            \fi
          \fi
          \global\@colroom\@colht
          \global\vsizel\@colht
          {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
      \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
      \fi
    }%
  }

```

3.6 Glossary Styles using longtable (glossary-longragged.sty)

The glossary styles defined in the glossary-longragged package used the longtable environment in the glossary and use ragged right formatting for the multiline columns.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-longragged-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-longragged-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2025/04/14 v4.6 (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]{%
  \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}%
  \glspostdescription\space ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

`longraggedborder` The `longraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longraggedborder}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|>{\raggedright}p{\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 `glostyplelongraggedborder` style:

```
\setglossarystyle{longraggedborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline\hline
  \endhead
  \hline\endfoot}%
}
```

`longragged3col` The `longragged3col` style is like `longragged` but with 3 columns

```
\newglossarystyle{longragged3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\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]{%
  \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
}

```

`longragged3colborder` The `longragged3colborder` style is like the `longragged3col` style but with a border:

```

\newglossarystyle{longragged3colborder}{%

```

Base it on the `glostylelongragged3col` style:

```

\setglossarystyle{longragged3col}%

```

Use a `longtable` with 3 columns with vertical lines around them:

```

\renewenvironment{theglossary}%
{
\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}}%
{\end{longtable}}%

```

Place horizontal lines at the head and foot of the table:

```

\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}

```

`longragged3colheader` The `longragged3colheader` style is like `longragged3col` but with a header row:

```

\newglossarystyle{longragged3colheader}{%

```

Base it on the `glostylelongragged3col` style:

```

\setglossarystyle{longragged3col}%

```

Set the table's header:

```

\renewcommand*{\glossaryheader}{%
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\endhead}%
}

```

`longragged3colheaderborder` The `longragged3colheaderborder` style is like the above but with a border

```

\newglossarystyle{longragged3colheaderborder}{%

```

Base it on the `glostylelongragged3colborder` style:

```

\setglossarystyle{longragged3colborder}%

```

Set the table's header and add horizontal line at table's foot:

```

\renewcommand*{\glossaryheader}{%
\hline
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}

```

`altlongragged4col` The `altlongragged4col` style is like the `altlong4col` style defined in the `glossary-long` package, except that ragged right formatting is used for the description and page list columns.

```

\newglossarystyle{altlongragged4col}{%

```


Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[1]>{\raggedright}p{\glsdescwidth}l%
   >{\raggedright}p{\glspagelistwidth}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a single row (name in first column, description in second column, symbol in third column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & \glossentrysymbol{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`altlongragged4colheader` The `altlongragged4colheader` style is like `altlongragged4col` but with a header row.

```
\newglossarystyle{altlongragged4colheader}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[1]>{\raggedright}p{\glsdescwidth}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{v4.6}{2025-04-14}
```

Declare package:

```
\ProvidesPackage{glossary-mcols}[2025/04/14 v4.6 (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}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%

      \let\item\glstreeitem}%
    {\end{multicols}}%
}
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
```

`mcoltree` Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsncols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
}
```

`mcoltreegroup` Like the `mcoltree` style but the glossary groups have headings.

```
\newglossarystyle{mcoltreegroup}{%
```

Base it on the `glostylemcoltree` style:

```
\setglossarystyle{mcoltree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
\renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt{\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{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
  \indexspace}%
}
```

`mcoltreesspannav` Similar to the `mcoltreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcoltreesspannav}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\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}%
}
```

`mcoltreename` Multi-column index style. Same as the `treename`, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltreename}{%
  \setglossarystyle{treename}%
  \renewenvironment{theglossary}%
  {%
```

```

        \begin{multicols}{\glsmcols}
        \setlength{\parindent}{0pt}%
        \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
}

```

`mcoltreenamegroup` Like the `mcoltreename` style but the glossary groups have headings.

```

    \newglossarystyle{mcoltreenamegroup}{%
Base it on the glostylemcoltreename style:
    \setglossarystyle{mcoltreename}%
Give each group a heading:
    \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}

```

`mcoltreenamehypergroup` The `mcoltreenamehypergroup` style is like the `mcoltreenamegroup` style, but has a set of links to the groups at the start of the glossary.

```

    \newglossarystyle{mcoltreenamehypergroup}{%
Base it on the glostylemcoltreename style:
    \setglossarystyle{mcoltreename}%
Put navigation links to the groups at the start of the theglossary environment:
    \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Each group has a heading (in bold with a target) followed by a vertical gap):
    \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}

```

`mcoltreenamepannav` Similar to the `mcoltreenamehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

    \newglossarystyle{mcoltreenamepannav}{%
    \setglossarystyle{treename}%
    \renewenvironment{theglossary}%
    {%
        \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
        \setlength{\parindent}{0pt}%
        \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
}

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```

    \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent

```



```

\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

`mcolalttree` Multi-column index style. Same as the `alttree`, but puts the glossary in multiple columns.

```

\newglossarystyle{mcolalttree}{%
\setglossarystyle{alttree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
}

```

`mcolalttreegroup` Like the `mcolalttree` style but the glossary groups have headings.

```

\newglossarystyle{mcolalttreegroup}{%
Base it on the glostylemcolalttree style:
\setglossarystyle{mcolalttree}%
Give each group a heading.
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}

```

`mcolalttreehypergroup` The `mcolalttreehypergroup` style is like the `mcolalttreegroup` style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcolalttreehypergroup}{%
Base it on the glostylemcolalttree style:
\setglossarystyle{mcolalttree}%
Put the navigation links in the header
\renewcommand*\glossaryheader{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Put a hypertarget at the start of each group
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%

```

```

    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgsetgrouptitle{##1}}}\par
    \indexspace}%
}

```

`mcolalttreespannav` Similar to the `mcolalmtreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

\newglossarystyle{mcolalttreespannav}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}

```

Put a hypertarget at the start of each group

```

\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgsetgrouptitle{##1}}}\par
  \indexspace}%
}

```

3.8 Glossary Styles using supertabular environment (`glossary-super.sty`)

The glossary styles defined in the `glossary-super` package use the `supertabular` environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-super-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-super-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}

```

Declare package:

```
\ProvidesPackage{glossary-super}[2025/04/14 v4.6 (NLCT)]
```

Requires the `supertabular` package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined if `glossary-long` has been loaded.

```

\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
}

```

```

\setlength{\glsdescwidth}{0.6\hsize}
}{}

```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if `glossary-long` has been loaded.

```

\@ifundefined{glspagelistwidth}{%
\newlength\glspagelistwidth
\setlength{\glspagelistwidth}{0.1\hsize}
}{}

```

`super` The super glossary style uses the `supertabular` environment (it uses lengths defined in the `glossary-long` package.)

```

\newglossarystyle{super}{%

```

Put the glossary in a `supertabular` environment with two columns and no head or tail:

```

\renewenvironment{theglossary}%
{\tablehead{}}\tabletail{}%
\begin{supertabular}{lp{\glsdescwidth}}%
{\end{supertabular}}%

```

Do nothing at the start of the table:

```

\renewcommand*{\glossaryheader}{}%

```

No group headings:

```

\renewcommand*{\glsgroupheading}[1]{}%

```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```

\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%

```

Sub entries put in a row (no name, description and page list in second column):

```

\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
##3\tabularnewline
}%

```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}

```

superborder The superborder style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superborder}{%
```

Base it on the glostylesuper style:

```
\setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\hline}\tabletail{\hline}%  
   \begin{supertabular}{|l|p{\glsdescwidth}|}%  
   {\end{supertabular}}%  
}
```

superheader The superheader style is like the super style, but with a header:

```
\newglossarystyle{superheader}{%
```

Base it on the glostylesuper style:

```
\setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and no tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\bfseries \entryname &  
   \bfseries \descriptionname\tabularnewline}%  
   \tabletail{}%  
   \begin{supertabular}{lp{\glsdescwidth}}%  
   {\end{supertabular}}%  
}
```

superheaderborder The superheaderborder style is like the super style but with a header and border:

```
\newglossarystyle{superheaderborder}{%
```

Base it on the glostylesuper style:

```
\setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and horizontal lines above and below the table:

```
\renewenvironment{theglossary}%  
  {\tablehead{\hline\bfseries \entryname &  
   \bfseries \descriptionname\tabularnewline\hline}%  
   \tabletail{\hline}  
   \begin{supertabular}{|l|p{\glsdescwidth}|}%  
   {\end{supertabular}}%  
}
```

super3col The super3col style is like the super style, but with 3 columns:

```
\newglossarystyle{super3col}{%
```

Put the glossary in a supertabular environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}}%
  {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*\glossaryheader{}
```

No group headings:

```
\renewcommand*\glsgruppeheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{& \tabularnewline}%
\fi
}
```

`super3colborder` The `super3colborder` style is like the `super3col` style, but with a border:

```
\newglossarystyle{super3colborder}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
  {\end{supertabular}}}%
}
```

super3colheader The `super3colheader` style is like the `super3col` style but with a header row:

```
\newglossarystyle{super3colheader}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&  
    \bfseries\pagelistname\tabularnewline}\tabletail{}}%  
  \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%  
  {\end{supertabular}}%  
}
```

super3colheaderborder The `super3colheaderborder` style is like the `super3col` style but with a header and border:

```
\newglossarystyle{super3colheaderborder}{%
```

Base it on the `glostylesuper3colborder` style:

```
\setglossarystyle{super3colborder}%
```

Put the glossary in a `supertabular` environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\hline  
    \bfseries\entryname&\bfseries\descriptionname&  
    \bfseries\pagelistname\tabularnewline\hline}%  
  \tabletail{\hline}%  
  \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%  
  {\end{supertabular}}%  
}
```

super4col The `super4col` glossary style has four columns, where the third column contains the value of the corresponding symbol key used when that entry was defined.

```
\newglossarystyle{super4col}{%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{} \tabletail{}}%  
  \begin{supertabular}{l|l|l|l}}{%  
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \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}{1111}}%
  {\end{supertabular}}%
}
```

`super4colborder` The `super4colborder` style is like the `super4col` but with a border.

```
\newglossarystyle{super4colborder}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|1|1|1|1|}%
  {\end{supertabular}}%
}
```

`super4colheaderborder` The `super4colheaderborder` style is like the `super4col` but with a header and border.

```
\newglossarystyle{super4colheaderborder}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\symbolname &
  \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|1|1|1|1|}%
  {\end{supertabular}}%
}
```

`altsuper4col` The `altsuper4col` glossary style is like `super4col` but has provision for multiline descriptions.

```
\newglossarystyle{altsuper4col}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  {\end{supertabular}}%
}
```

`altsuper4colheader` The `altsuper4colheader` style is like the `altsuper4col` but with a header row.

```
\newglossarystyle{altsuper4colheader}{%
```

Base it on the `glostylesuper4colheader` style:

```
\setglossarystyle{super4colheader}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
```



```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\tabletail{}%
        \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
        {\end{supertabular}}%
    }

```

altsuper4colborder The `altsuper4colborder` style is like the `altsuper4col` but with a border.

```
\newglossarystyle{altsuper4colborder}{%
```

Base it on the `glostylesuper4colborder` style:

```
\setglossarystyle{super4colborder}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
    {l|lp{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
    {\end{supertabular}}%
}

```

altsuper4colheaderborder The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```
\newglossarystyle{altsuper4colheaderborder}{%
```

Base it on the `glostylesuper4colheaderborder` style:

```
\setglossarystyle{super4colheaderborder}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
{\tablehead{\hline
    \bfseries\entryname &
    \bfseries\descriptionname &
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
{ l|lp{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
{\end{supertabular}}%
}

```

3.9 Glossary Styles using `supertabular` environment (`glossary-superragged.sty`)

The glossary styles defined in the `glossary-superragged` package use the `supertabular` environment. These styles are like those provided by the `glossary-super` package, except that the multiline columns have ragged right justification.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-superragged-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-superragged-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2025/04/14 v4.6 (NLCT)]
```

Requires the array package:

```
\RequirePackage{array}
```

Requires the supertabular package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`superragged` The superragged glossary style uses the supertabular environment.

```
\newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
```

`superraggedborder` The `superraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superraggedborder}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}%
  {\end{supertabular}}%
}
```

`superraggedheader` The `superraggedheader` style is like the `super` style, but with a header:

```
\newglossarystyle{superraggedheader}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
}
```

`superraggedheaderborder` The `superraggedheaderborder` style is like the `superragged` style but with a header and border:

```
\newglossarystyle{superraggedheaderborder}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and horizontal lines above and below the table:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
  \tabletail{\hline}
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}%
  {\end{supertabular}}%
}
```

`superragged3col` The `superragged3col` style is like the `superragged` style, but with 3 columns:

```
\newglossarystyle{superragged3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

`superragged3colborder` The `superragged3colborder` style is like the `superragged3col` style, but with a border:

```
\newglossarystyle{superragged3colborder}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
  >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}
```

`superragged3colheader` The `superragged3colheader` style is like the `superragged3col` style but with a header row:

```
\newglossarystyle{superragged3colheader}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline}\tabletail{}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
  >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
}
```

`raggedright3colheaderborder` The `superragged3colheaderborder` style is like the `superragged3col` style but with a header and border:

```
\newglossarystyle{superragged3colheaderborder}{%
```

Base it on the `glostylesuperragged3colborder` style:

```
\setglossarystyle{superragged3colborder}%
```

Put the glossary in a supertabular environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
}
```

`altsuperragged4col` The `altsuperragged4col` glossary style is like `altsuper4col` style in the `glossary-super` package but uses ragged right formatting in the description and page list columns.

```
\newglossarystyle{altsuperragged4col}{%
```

Put the glossary in a supertabular environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}

```

`altsuperragged4colheader` The `altsuperragged4colheader` style is like the `altsuperragged4col` style but with a header row.

```
\newglossarystyle{altsuperragged4colheader}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\symbolname &
  \bfseries\pagelistname\tabletail}%
  \begin{supertabular}{1>{\raggedright}p{\glstdescwidth}1%
  >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
}

```

`altsuperragged4colborder` The `altsuperragged4colborder` style is like the `altsuperragged4col` style but with a border.

```
\newglossarystyle{altsuperragged4colborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuper4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}%
  {1|>{\raggedright}p{\glstdescwidth}1|1%
  >{\raggedright}p{\glspagelistwidth}1|}%
  {\end{supertabular}}%
}

```

`superragged4colheaderborder` The `altsuperragged4colheaderborder` style is like the `altsuperragged4col` style but with a header and border.

```
\newglossarystyle{altsuperragged4colheaderborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline
    \bfseries\entryname &
    \bfseries\descriptionname &
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}%
    {|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{v4.6}{2025-04-14}

```

Declare package:

```
\ProvidesPackage{glossary-tree}[2025/04/14 v4.6 (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⟩}`

`\GlsTreeUpdateWidestNameOrSymbol`

```
\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⟩}`

`\GlsTreeUpdateWidestNameAndSymbol`

```
\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 <{level}> <{label}>
<{title}>
```

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 <{parent-label}>
<{group title}>
```

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 \glsupdatewidest` 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_namesonlywidth #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_namesonlywidth #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
{

```

```

    \l_glossaries_tree_show_loc:nnn { #1 } { #2 } { #3 }
  }

```

Separator between name and symbol:

```

\l_new:N \l_glossaries_tree_name_sym_sep_tl
\l_set:Nn \l_glossaries_tree_name_sym_sep_tl { ~ }
\l_new:N \l_glossaries_tree_child_name_sym_sep_tl
\l_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 \l_glossaries_tree_name_sym_item:nnn
{
  \exp_args:NV \l_glossaries_tree_name_sym_item:nnnn
  \l_glossaries_tree_name_sym_sep_tl
  { #1 } { #2 } { #3 }
}
\cs_new:Nn \l_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 \l_glossaries_tree_name_sym_subitem:nnn
{
  \exp_args:NV \l_glossaries_tree_name_sym_subitem:nnnn
  \l_glossaries_tree_child_name_sym_sep_tl
  { #1 } { #2 } { #3 }
}
\cs_new:Nn \l_glossaries_tree_name_sym_subitem:nnnn
{
  \l_glossaries_tree_name_sym_item:nnnn
  { #1 } { #2 } { #3 } { #4 }
}

```

Separators between name/symbol and description:

```

\l_new:N \l_glossaries_tree_post_namesymbol_tl
\l_set:Nn \l_glossaries_tree_post_namesymbol_tl { ~ }
\l_new:N \l_glossaries_tree_post_child_namesymbol_tl
\l_set:Nn \l_glossaries_tree_post_child_namesymbol_tl
{
  \l_glossaries_tree_post_namesymbol_tl
}
\l_new:N \l_glossaries_tree_pre_desc_tl
\l_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
  { l__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
    { l__glossaries_tree_namewidth #1 _dim }
  }
  {
    \dim_compare_p:nNn
    { \dim_use:c { l__glossaries_tree_namewidth #1 _dim } }
    >
    { \c_zero_dim }
  }
}
{
  \tl_if_exist:cTF { l__glossaries_tree_namealign #1 _tl }
  {
    \glossaries_tree_namebox:vvn
    { l__glossaries_tree_namewidth #1 _dim }
    { l__glossaries_tree_namealign #1 _tl }
    { #2 }
  }
  {
    \glossaries_tree_namebox:vVn
    { l__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:vvn
    { l__glossaries_tree_symbolwidth #1 _dim }
    { l__glossaries_tree_symbolalign #1 _tl }
    { #2 }
  }
  {
    \glossaries_tree_symbolbox:vVn
    { 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
      {

```

```

        \l_glossaries_tree_name_font:n
        {
            \exp_args:NV \l_glossaries_tree_name_case:n
            \@glswidestname
        }
    }
}
}
\dim_add:Nn \l_glossaries_tree_namesymbolwidth_dim
{ \l_glossaries_tree_namewidth_padding_dim }
}
\cs_if_eq:NNF \l_glossaries_tree_symbol_encap:n \use_none:n
{
    \settoheight \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 }
    }
    {
        \settoheight \l_glossaries_tmpa_dim
        {
            \l_glossaries_tree_symbol_encap:n
            {
                \l_glossaries_tree_symbol_font:n
                {
                    \exp_args:NV \l_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 “unsorted” sub-groups:

```

sub-header-font .code:n =
{
  \tl_if_eq:nTF { #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
  { \glssuppercase { ##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
{ \glssuppercase { ##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
  { \glssuppercase { ##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
    { \glssupercase { ##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:nmmn
{
  \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:nmmn
  {
    \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 .choices:nn =
{ l , r , c }
{
  \tl_set_eq:NN

```

```

        \l__glossaries_tree_prenamesym_align_tl
        \l_keys_choice_tl
    } ,
sub-item-counter-align .choices:nn =
{ l , r , c }
{
    \tl_set_eq:NN
    \l__glossaries_tree_presubnamesym_align_tl
    \l_keys_choice_tl
} ,
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 .choices:nn =
{ l , r , c }
{
  \tl_set_eq:NN
  \l__glossaries_tree_namealign_tl
  \l_keys_choice_tl
} ,
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 .choices:nn =
{ l , r , c }
{

```

```

\tl_set_eq:NN
  \l__glossaries_tree_symbolalign_tl
  \l_keys_choice_tl
},
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 .choices:nn =
{ l , r , c }
{
  \tl_set_eq:NN
    \l__glossaries_tree_namesymbolalign_tl
    \l_keys_choice_tl
},
}

```

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
```

```

    {
      \glxtrgetgrouptitle { #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 }
  }
  {
    \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 \glxtrgetgrouptitle
    {
      \glxtrgetgrouptitle { #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 }
}
{
  \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_hypernav_bool
{
  \tl_set:Ne \l__glossaries_tree_group_title_tl
  {
    \exp_not:N \glsnavhypertarget { #1 }
    { \exp_not:V \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 \__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_namesymbol: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_namesymbol: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
}

```

`\glstreenamfmt` 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*{\glstreenamfmt}[1]{\textbf{#1}}

```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamfmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamfmt` would've also affected the group headings.

```

\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamfmt{#1}}

```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```

\newcommand*{\glstreenavigationfmt}[1]{\glstreenamfmt{#1}}

```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```

\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}

```

`\glstreesubitem` Level 1 item used in index style.

```

\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}

```

`\glstreesubsubitem` Level 1 item used in index style.

```

\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}

```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `almtree` style).

```
\newcommand{\glstreepredesc}{\space}
```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treenoname` and `almtree` style).

```
\newcommand{\glstreechildpredesc}{\space}
```

`index` The index glossary style is similar in style to the way indices are usually typeset using `\item`, `\subitem` and `\subsubitem`. The entry name is set in bold. If an entry has a symbol, it is placed in brackets after the name. Then the description is displayed, followed by the number list. This style allows up to three levels.

```
\newglossarystyle{index}{%
```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```
\renewenvironment{theglossary}%  
  {\setlength{\parindent}{0pt}%  
   \setlength{\parskip}{0pt plus 0.3pt}%  
   \let\item\glstreeitem  
   \let\subitem\glstreesubitem  
   \let\subsubitem\glstreesubsubitem  
  }%
```

```
{\par}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headers:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entry starts a new item with the name in bold followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand*{\glossentry}[2]{%  
  \item\glstreeitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%  
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%  
  \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%  
}%
```

Sub entries: level 1 entries use `\subitem`, levels greater than 1 use `\subsubitem`. The level (`##1`) shouldn't be 0, as that's catered by `\glossentry`, but for completeness, if the level is 0, `\item` is used. The name is put in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%  
  \ifcase##1\relax  
    % level 0  
    \item  
  \or  
    % level 1
```

```

        \subitem
        \glssubentryitem{##2}%
    \else
        % all other levels
        \subsubitem
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \ifglshassymbol{##2}{\space\glossentrysymbol{##2}}{}%
    \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%

```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}}
```

indexgroup The `indexgroup` style is like the `index` style but has headings.

```
\newglossarystyle{indexgroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Add a heading for each group. This puts the group's title in bold followed by a vertical gap.

```

    \renewcommand*\glsgroupheading}[1]{%
        \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
        \indexspace
    }%
}

```

indexhypergroup The `indexhypergroup` style is like the `indexgroup` style but has hyper navigation.

```
\newglossarystyle{indexhypergroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Put navigation links to the groups at the start of the glossary:

```

\renewcommand*\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%

```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```

\renewcommand*\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
}

```

tree The `tree` glossary style is similar in style to the `index` style, but can have arbitrary levels.

```
\newglossarystyle{tree}{%
```


Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
```

Do nothing at the start of the theglossary environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries: name in bold, followed by symbol in brackets (if it exists), the description and the page list:

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\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{\glsgetgrouptitle{##1}}\par
  \indexspace}%
}
```

`treehypergroup` The `treehypergroup` style is like the `treegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treehypergroup}{%
```

Base it on the `glostyletree` style:

```
\setglossarystyle{tree}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`\glstreeindent` Length governing left indent for each level of the tree style.

```
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
```

`treenoname` The `treenoname` glossary style is like the `tree` style, but doesn't print the name or symbol for sub-levels.

```
\newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}
```

No header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries: the name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name and symbol are omitted. The description followed by the page list are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

```

\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
  \glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
\glossentrydesc{##2}\glspostdescription\space##3\par
}%

```

Vertical gap between groups is the same as that used by indices:

```

\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}

```

treenonamegroup Like the `treenoname` style but the glossary groups have headings.

```

\newglossarystyle{treenonamegroup}{%
Base it on the glostyletreenoname style:
  \setglossarystyle{treenoname}%
Give each group a heading:
  \renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsgetgrouptitle{##1}}\par\indexspace}%
}

```

treenonamehypergroup The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{treenonamehypergroup}{%
Base it on the glostyletreenoname style:
  \setglossarystyle{treenoname}%
Put navigation links to the groups at the start of the theglossary environment:
  \renewcommand*{\glossaryheader}{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Each group has a heading (in bold with a target) followed by a vertical gap):
  \renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}

```

\glsfindwidesttoplevelname Find the widest name over all parentless entries in the given glossary or glossaries.

```

\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%

```

```

\for glossaries [\@gls@type]{\@glo@label}%
{%
  \if glossary has parent {\@glo@label}%
  {}%
  {%
    \gls measure width {\dimen@}%
    {\@gls@tree name format {\@gls@entry name {\@glo@label}}}%
    \if dim \dimen@ > \gls@tmplen
      \gls@tmplen = \dimen@
      \let cs {\@gls@widest name} = \glo@{\@gls@detok label {\@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

```

`\@gls@widestname` Initialise `\@gls@widestname`.

```

\newcommand*{\@gls@widestname}{}

```

`\gls@tree name box` Used by the `alttree` style to create the box for the name and associated information.

```

\newcommand*{\gls@tree name box}[2]{%
  \makebox[#1][l]{#2}%
}

```

`alttree` The `alttree` glossary style is similar in style to the `tree` style, but the indentation is obtained from the width of `\@gls@widestname` which is set using `\glssetwidest`.

```

\newglossarystyle{alttree}{%

```

Redefine the `glossary` environment.

```

\renewenvironment{theglossary}%
{\def\@gls@prevlevel{-1}%
  \mbox{}\par}%
{\par}%

```

Set the header and group headers to nothing.

```

\renewcommand*{\glossaryheader}{}%
\renewcommand*{\gls@group heading}[1]{}%

```

Redefine the way that the level 0 entries are displayed.

```
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
```

Find out how big the indentation should be by measuring the widest entry.

```
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
```

Set the hangindent and paragraph indent.

```
  \hangindent\glstreeindent
  \parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
  \makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}
```

Do the description followed by the description terminator and location list.

```
  \glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
  \def\@gls@prevlevel{0}%
}%
```

Redefine the way sub-entries are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

Increment and display the sub-entry counter if this is a level 1 entry and the sub-entry counter is in use.

```
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
  \ifnum\@gls@prevlevel=##1\relax
  \else
```

Compute the widest entry for this level, or for level 0 if not defined for this level.

Store in `\gls@tmplen`

```
  \@ifundefined{@glswidestname\romannumeral##1}{%
    \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
    \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{
      \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
  \ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
\setlength\glstreeindent\gls@tmplen
\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
```

Depth has decreased, so subtract width of the widest entry from the previous level to `\glstreeindent`. First determine the width of the widest entry for the previous level and store in `\glstreeindent`.

```
\ifundefined{@glswidestname\romannumeral@gls@prevlevel}{%
\glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
\@glswidestname\space}}}{%
\glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral@gls@prevlevel
\endcsname\space}}}%
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
```

Set the hanging indentation.

```
\hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```
\def@gls@prevlevel{##1}%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

`almtreegroup` Like the `almtree` style but the glossary groups have headings.

```
\newglossarystyle{almtreegroup}{%
```

Base it on the `glostylealmtree` style:

```
\setglossarystyle{almtree}%
```

Give each group a heading.

```
\renewcommand{\glsgroupheading}[1]{\par
\def@gls@prevlevel{-1}%
```

```

    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
    \par\indexspace}%
}

```

`almtreehypergroup` The `almtreehypergroup` style is like the `almtreegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{almtreehypergroup}{%
```

Base it on the `glostylealmtree` style:

```
\setglossarystyle{almtree}%
```

Put the navigation links in the header

```

\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Put a hypertarget at the start of each group

```

\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

4 Multi-Lingual Support

4.1 Babel Captions (`glossaries-babel.sty`)

Many thanks to everyone who contributed to the translations both via email and on `comp.text.tex`. Language support has now been split off into independent language modules.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-babel-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-babel-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}

```

Declare package

```
\ProvidesPackage{glossaries-babel}[2025/04/14 v4.6 (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{v4.6}{2025-04-14}
```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2025/04/14 v4.6 (NLCT)]
```

Load tracklang to obtain language settings.

```
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
}%
{\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak
```



```

        Language modules need to be installed separately.\MessageBreak
        Please check on CTAN for a bundle called\MessageBreak
        ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
  }%
}%
{}%

```

5 Prefix Support (glossaries-prefix Code)

This package provides a means of adding prefixes to your glossary entries. For example, you may want to use “a \gls{*label*}” on first use but use “an \gls{*label*}” on subsequent use.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-prefix-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-prefix-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}

```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2025/04/14 v4.6 (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}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglschasprefixfirstplural{#2}%
      {\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
  }%
  \@glspl@{#1}{#2}[#3]%
}%
}
```

Now for the first letter upper case versions:

```
\PglS
\newrobustcmd{\PglS}{\@gls@hyp@opt\PglS}
```

Add sentence case mapping.

```
\glsmfuaddmap{\pgls}{\PglS}
```

`\@PglS` Unstarred version.

```
\newcommand*{\@PglS}[2][{}]{%
  \new@ifnextchar[%
  {\@PglS@{#1}{#2}}%
  {\@PglS@{#1}{#2}[{}]}%
}
```

`\@PglS@` Read in the final optional argument:

```
\def\@PglS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefix{#2}%
      {%
        \Glsentryprefix{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
    {%
      \ifglschasprefixfirst{#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}%
    {%
      \ifgls@hasprefixplural{#2}%
      {%
        \Glsentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
    {%
      \ifgls@hasprefixfirstplural{#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}%
{\glsuppercase{\glsentryprefix{#2}\glsprefixsep}}{}}%
}%
{%
\ifgls hasprefixfirst{#2}%
{\glsuppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{}}%
}%
\@GLS@{#1}{#2}[#3]%
}%
}
```

Plural version:

```
\PGLSp1
\newrobustcmd{\PGLSp1}{\@gls@hyp@opt\PGLSp1}
```

\@PGLSp1 Unstarred version.

```
\newcommand*{\@PGLSp1}[2] [] {%
\new@ifnextchar [%
{\@PGLSp1@{#1}{#2}}%
{\@PGLSp1@{#1}{#2} []}%
}
```

\@PGLSp1@ Read in the final optional argument:

```
\def\@PGLSp1@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
}
```

```

\glsdoifexists{#2}%
{%
  \ifglsused{#2}%
  {%
    \ifglschasprefixplural{#2}%
    {\glsuppercase{\glsentryprefixplural{#2}\glsprefixsep}}{}}%
  }%
  {%
    \ifglschasprefixfirstplural{#2}%
    {\glsuppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}}%
  }%
  \@GLSp1@{#1}{#2}[#3]%
}%
}

```

6 Accessibility Support (glossaries-accsupp Code)

The `glossaries-accsupp` package is experimental. It is intended to provide a means of using the PDF accessibility support in glossary entries. See the `accsupp` documentation for further details about accessibility support.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-accsupp-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-accsupp-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}

```

Package version number now in line with main `glossaries` package number.

```

\ProvidesPackage{glossaries-accsupp}[2025/04/14 v4.6 (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@desclpluralaccess{#1}%  
}
```

shortaccess The replacement text corresponding to the short key:

```
\define@key{glossentry}{shortaccess}{%  
  \def\@glo@shortaccess{#1}%  
}
```

shortpluralaccess The replacement text corresponding to the shortplural key:

```
\define@key{glossentry}{shortpluralaccess}{%  
  \def\@glo@shortpluralaccess{#1}%  
}
```

longaccess The replacement text corresponding to the long key:

```
\define@key{glossentry}{longaccess}{%  
  \def\@glo@longaccess{#1}%  
}
```

longpluralaccess The replacement text corresponding to the longplural key:

```
\define@key{glossentry}{longpluralaccess}{%  
  \def\@glo@longpluralaccess{#1}%  
}
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
\define@key{glossentry}{user1access}{%  
  \def\@glo@useriaccess{#1}%  
}
```

user2access The replacement text corresponding to the user2 key:

```
\define@key{glossentry}{user2access}{%  
  \def\@glo@useriiaccess{#1}%  
}
```

user3access The replacement text corresponding to the user3 key:

```
\define@key{glossentry}{user3access}{%  
  \def\@glo@useriiiaccess{#1}%  
}
```

user4access The replacement text corresponding to the user4 key:

```
\define@key{glossentry}{user4access}{%  
  \def\@glo@userivaccess{#1}%  
}
```

`user5access` The replacement text corresponding to the `user5` key:

```
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
```

`user6access` The replacement text corresponding to the `user6` key:

```
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
```

For any custom keys, the replacement text would have to be explicitly put in the value, e.g., `user1={\glsshortaccsupp{inches}{in}}`.

Append these new keys to `\@gls@keymap`:

```
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%

  {descriptionaccess}{descaccess},%
  {descriptionpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
```

`\@gls@noaccess` Indicates that no replacement text has been provided.

```
\def\@gls@noaccess{\relax}
```

Previously, the access key was initialised to the value of the symbol key at the start for backwards compatibility. This causes a problem for situations where the replacement text is provided for symbol but not for name so this behaviour has been removed.

```
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
}
```

Initialise the other keys:

```
\def@glo@textaccess{\@glo@access}%
\def@glo@firstaccess{\@glo@access}%
\def@glo@pluralaccess{\@glo@textaccess}%
\def@glo@firstpluralaccess{\@glo@pluralaccess}%
\def@glo@symbolaccess{\relax}%
\def@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def@glo@descaccess{\relax}%
\def@glo@descpluralaccess{\@glo@descaccess}%
\def@glo@shortaccess{\relax}%
\def@glo@shortpluralaccess{\@glo@shortaccess}%
\def@glo@longaccess{\relax}%
\def@glo@longpluralaccess{\@glo@longaccess}%
\def@glo@useriaccess{\relax}%
\def@glo@useriiaaccess{\relax}%
\def@glo@useriiaaaccess{\relax}%
\def@glo@useriiaaaaccess{\relax}%
\def@glo@useriivaaccess{\relax}%
\def@glo@userivaaccess{\relax}%
\def@glo@useriiaaaccess{\relax}%
}
```

Add to the end hook:

```
\let@gls@oldnewglossaryentryposthook@newglossaryentryposthook
\renewcommand*{@newglossaryentryposthook}{%
  @gls@oldnewglossaryentryposthook}
```

Store the access information:

```
\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
  \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
  \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
  \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
  \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
  \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
  \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
  \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
```

```

        \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
\@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
\@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
\@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
\@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
\@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
\@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
\@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
\@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
\@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
\@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
\@glo@userviaccess}%
}

```

6.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

\newcommand*\glsentryaccess}[1]{%
\@gls@entry@field{#1}{access}%
}

```

`\glsentrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

\newcommand*\glsentrytextaccess}[1]{%
\@gls@entry@field{#1}{textaccess}%
}

```

`\glsentryfirstaccess` Get the value of the firstaccess key for the entry with the given label:

```

\newcommand*\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}

```

`\glsentrypluralaccess` Get the value of the pluralaccess key for the entry with the given label:

```

\newcommand*\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}

```

`\glsentryfirstpluralaccess` Get the value of the firstpluralaccess key for the entry with the given label:

```

\newcommand*\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}

```

`\glsentrysymbolaccess` Get the value of the symbolaccess key for the entry with the given label:

```

\newcommand*\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}

```

`\glsentrysymbolpluralaccess` Get the value of the symbolpluralaccess key for the entry with the given label:

```

\newcommand*\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}

```

`\glsentrydescaccess` Get the value of the descriptionaccess key for the entry with the given label:

```

\newcommand*\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}

```

`\glsentrydescpluralaccess` Get the value of the descriptionpluralaccess key for the entry with the given label:

```

\newcommand*\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}

```

`\glsentryshortaccess` Get the value of the shortaccess key for the entry with the given label:

```

\newcommand*\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}

```

`\glsentryshortpluralaccess` Get the value of the shortpluralaccess key for the entry with the given label:

```

\newcommand*\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}

```

`\glsentrylongaccess` Get the value of the longaccess key for the entry with the given label:

```

\newcommand*\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

`\glentrylongpluralaccess` Get the value of the `longpluralaccess` key for the entry with the given label:

```
\newcommand*{\glentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
```

`\glentryuseriaccess` Get the value of the `user1access` key for the entry with the given label:

```
\newcommand*{\glentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
```

`\glentryuseriiaccess` Get the value of the `user2access` key for the entry with the given label:

```
\newcommand*{\glentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
```

`\glentryuseriiiaccess` Get the value of the `user3access` key for the entry with the given label:

```
\newcommand*{\glentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
```

`\glentryuserivaccess` Get the value of the `user4access` key for the entry with the given label:

```
\newcommand*{\glentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
```

`\glentryuservaccess` Get the value of the `user5access` key for the entry with the given label:

```
\newcommand*{\glentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
```

`\glentryuserviaccess` Get the value of the `user6access` key for the entry with the given label:

```
\newcommand*{\glentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
```

There are three types of replacement text:

Alt Description of some content that's non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, "St" could be short for "saint" or "street").

Therefore, rather than having one command for all fields, it's better to have a command dependent on the field type. For example, the `short` and `shortpl` keys would require **E**, the `symbol` key would require **ActualText**, and a field that contains an image would require **Alt**.


```
\glsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\glsfieldaccsupp`

Test if there's a command called `\gls{field}accsupp`. If there is then use that otherwise use `\glsaccsupp`. The first argument should be the internal field label (not the key). The final argument is the entry label. If `glossaries-extra` has been loaded, this first checks for `\glsxtr{category}{field}accsupp` and `\glsxtr{category}accsupp`.

```
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
```

```
\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\xglsfieldaccsupp`

As `\glsfieldaccsupp` but fully expand replacement text.

```
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
```

```
\glsshortaccsupp{<replacement text>}{<text>}
```

`\glsshortaccsupp`

```
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

```
\glsshortplaccsupp{<replacement text>}{<text>}
```

`\glsshortplaccsupp`

```
\newcommand*\glsshortplaccsupp{\glsshortaccsupp}
```

```
\glsaccsupp{<replacement text>}{<text>}
```

`\glsaccsupp`

```
\newcommand*\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
```

`\xglsaccsupp` Fully expands replacement text before calling `\glsaccsupp`

```
\newcommand*\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
```

`\@gls@access@display` Deprecated in v4.45. Use `\@gls@fieldaccess@display` instead.

```
\newcommand*\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
```

```
\@gls@fieldaccess@display{<label>}{<field>}{<content>}
{<replacement>}
```

`\@gls@fieldaccess@display`

```
\newcommand*\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \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}}}%
}
```

`\gls textaccessdisplay` As above but for the textaccess replacement text.

```
\newrobustcmd*\gls textaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@textaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
```

`\glspluralaccessdisplay` As above but for the pluralaccess replacement text.

```

\newrobustcmd*{\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@pluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}

```

`\glsfirstaccessdisplay` As above but for the firstaccess replacement text.

```

\newrobustcmd*{\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@firstaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}

```

`\glsfirstpluralaccessdisplay` As above but for the firstpluralaccess replacement text.

```

\newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@firstpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}

```

`\glssymbolaccessdisplay` As above but for the symbolaccess replacement text.

```

\newrobustcmd*{\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@symbolaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}

```

`\glssymbolpluralaccessdisplay` As above but for the symbolpluralaccess replacement text.

```

\newrobustcmd*{\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@symbolpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}

```

`\glsdescriptionaccessdisplay` As above but for the descriptionaccess replacement text.

```

\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@descaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}

```

`\glsdescriptionpluralaccessdisplay` As above but for the descriptionpluralaccess replacement text.

```

\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@descpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}

```

`\glsshortaccessdisplay` As above but for the shortaccess replacement text.

```
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{short}{#1}{\glstryshortaccess{#2}}}%
}
```

`glsshortpluralaccessdisplay` As above but for the shortpluralaccess replacement text.

```
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstryshortpluralaccess{#2}}}%
}
```

`\glslongaccessdisplay` As above but for the longaccess replacement text.

```
\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@longaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{long}{#1}{\glstrylongaccess{#2}}}%
}
```

`\glslongpluralaccessdisplay` As above but for the longpluralaccess replacement text.

```
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@longpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glstrylongpluralaccess{#2}}}%
}
```

`\glsuseriaccessdisplay` As above but for the user1access replacement text.

```
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@useriaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useri}{#1}{\glstryuseriaccess{#2}}}%
}
```

`\glsuseriiaccessdisplay` As above but for the user2access replacement text.

```
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@useriiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userii}{#1}{\glstryuseriiaccess{#2}}}%
}
```

`\glsuseriiiaccessdisplay` As above but for the user3access replacement text.

```
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glstryuseriiiaccess{#2}}}%
}
```

`\glsuserivaccessdisplay` As above but for the user4access replacement text.

```

\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\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
  {%
    \glsifplural
    {%
      Plural form
      \glsapscase
    }%
  }%
}

```

Don't adjust case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glspluralaccessdisplay  
  {\glsentryplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
{%
```

First use

```
\glsfirstpluralaccessdisplay  
  {\glsentryfirstplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
}%  
{%
```

Make first letter upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use.

```
\glspluralaccessdisplay  
  {\Glsentryplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
{%
```

First use

```
\glsfirstpluralaccessdisplay  
  {\Glsentryfirstplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
}%  
{%
```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glspluralaccessdisplay  
  {\glsuppercase{\glsentryplural{\glslabel}}}%  
  {\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
{%
```

First use

```
\glsfirstpluralaccessdisplay  
  {\glsuppercase{\glsentryfirstplural{\glslabel}}}%
```

```

        {\glslabel}%
    \glsuppercase{\glsinsert}%
  }%
}%
}%
{%
```

Singular form

```

\glscapscase
{%
```

Don't adjust case

```

\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  
}%  
}
```

`\glsgenacfmt` Redefine to include accessibility information.

```
\renewcommand*{\glsgenacfmt}{%  
  \ifdefempty\glscustomtext  
  {%  
    \ifglsused\glslabel  
    {%
```

Subsequent use:

```
\glsifplural  
{%
```

Subsequent plural form:

```
\glsapscase  
{%
```

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
{\glstryshortpl{\glslabel}}{\glslabel}}%
\glsinsert}%
}%
}%
{%
```

Subsequent singular form

```
\glscapscase
{%
```

Subsequent singular form, don't adjust case:

```
\acronymfont
{\glsshortaccessdisplay{\glstryshort{\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{\glstryshort{\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:

```
\glsuppercase  
{\genacrfullformat{\glslabel}{\glsinsert}}}%  
}%  
}%  
}%  
{%
```

User supplied text. (The insert should be empty at this point.) The accessibility information, if required, will have to be explicitly included in the custom text.

```
\glscustomtext  
}%  
}
```

`\genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\genacrfullformat}[2]{%  
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space  
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%  
}
```

`\Genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\Genacrfullformat}[2]{%  
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space  
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%  
}
```

```

\genplacrfullformat Redefine to include accessibility information.
    \renewcommand*{\genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
    }

\Genplacrfullformat Redefine to include accessibility information.
    \renewcommand*{\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
    }

\@acrshort
\def\@acrshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\relax

        \let\glsifplural\@secondoftwo
        \let\glscapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
        }%
    }

    Call \@gls@link
        \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
}

\@Acrshort
\def\@Acrshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\relax

        \let\glsifplural\@secondoftwo
        \let\glsapspace\@secondofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
        }%
    }

    Call \@gls@link
        \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
    }%

```

```

    \glspostlinkhook
  }

\@ACRshort
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay
        {\glsuppercase{\glsentryshort{#2}}{#2}}#3%
    }%
  }%

  Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\@acrlong
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
  }%

  Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\@Acrlong
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

```



```

\@glsfirst@
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glsentryfirst{#2}}{#2}#3}%
}

\@Glsfirst@
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}

\@GLSfirst@
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstaccessdisplay{\glsupercase{\glsentryfirst{#2}}}{#2}%
  \glsupercase{#3}}%
}

\@glsplural@
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}

\@Glsplural@
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}

\@GLSplural@
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glspluralaccessdisplay{\glsupercase{\glsentryplural{#2}}}{#2}%
  \glsupercase{#3}}%
}

\@glsfirstplural@
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}

\@Glsfirstplural@
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

\@GLSfirstplural@
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\glsupercase{\glsentryfirstplural{#2}}}{#2}%
  \glsupercase{#3}}%
}

```

```

\@glsname@
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}

\@Glsname@
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}

\@GLSname@
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsnameaccessdisplay{\glsuppercase{\glsentryname{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdesc@
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}

\@Glsdesc@
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}

\@GLSdesc@
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionaccessdisplay{\glsuppercase{\glsentrydesc{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdescplural@
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}

\@Glsdescplural@
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}

\@GLSdescplural@
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionpluralaccessdisplay{\glsuppercase{\glsentrydescplural{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

```

\@glssymbol@
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}

\@Glsymbol@
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}

\@GLSsymbol@
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolaccessdisplay{\glsupercase{\glsentrysymbol{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsymbolplural@
\def\@glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}

\@Glsymbolplural@
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}

\@GLSsymbolplural@
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsymbolpluralaccessdisplay{\glsupercase{\glsentrysymbolplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseri@
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}

\@Glsuseri@
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}

\@GLSuseri@
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\glsupercase{\glsentryuseri{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```



```

\@glsuserii@
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

\@Glsuserii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}

\@GLSuserii@
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriiaccessdisplay{\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{\glsuppercase{\glsentryuserv{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsuservi@
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}

\@Glsuservi@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}

\@GLSuservi@
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserviaccessdisplay{\glsuppercase{\glsentryuservi{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

6.3 Displaying the Glossary

We need to redefine the way the glossary entries are formatted to include the accessibility support. The predefined glossary styles use `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol`, but we need to provide compatibility with earlier versions in case users have defined their own styles using `\accsuppglossaryentryfield` and `\accsuppglossarysubentryfield`.

Now redefine `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol` etc so they use the accessibility stuff.

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}

```

```

\renewcommand*\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

\renewcommand*\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

```

\accsuppglossaryentryfield

```

\newcommand*\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}

```

\accsuppglossarysubentryfield

```

\newcommand*\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}

```

6.4 Acronyms

Redefine acronym styles provided by glossaries:

long-short $\langle long \rangle$ ($\langle short \rangle$) acronym style.

```
\renewacronymstyle{long-short}%  
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%  
}%  
{%  
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%  
\renewcommand*{\genacrfullformat}[2]{%  
\glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space  
\glsshortaccessdisplay  
\protect\firstacronymfont{\glsentryshort{##1}}{##1})%  
}%  
\renewcommand*{\Genacrfullformat}[2]{%  
\glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space  
\glsshortaccessdisplay  
\protect\firstacronymfont{\glsentryshort{##1}}{##1})%  
}%  
\renewcommand*{\genplacrfullformat}[2]{%  
\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space  
\glsshortpluralaccessdisplay  
\protect\firstacronymfont{\glsentryshortpl{##1}}{##1})%  
}%  
\renewcommand*{\Genplacrfullformat}[2]{%  
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space  
\glsshortpluralaccessdisplay  
\protect\firstacronymfont{\glsentryshortpl{##1}}{##1})%  
}%  
\renewcommand*{\acronymentry}[1]{%  
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}  
\renewcommand*{\acronymsort}[2]{##1}%  
\renewcommand*{\acronymfont}[1]{##1}%  
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%  
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%  
}
```

short-long $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```
\renewacronymstyle{short-long}%  
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%  
}%  
{%  
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%  
\renewcommand*{\genacrfullformat}[2]{%
```

```

\glsshortaccessdisplay
  {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\Glentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
  {\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

`long-short-desc` *⟨long⟩* (*{⟨short⟩}*) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})%
}

```

`long-sc-short-desc` *⟨long⟩* (`\textsc{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%

```

```

    {%
      \GlsUseAcrStyleDefs{long-sc-short}%
      \renewcommand*\GenericAcronymFields{}%
      \renewcommand*\acronymsort}[2]{##2}%
      \renewcommand*\acronymentry}[1]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
    }

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (which the user needs to supply).
  \renewacronymstyle{long-sm-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-sm-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sm-short}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
    \renewcommand*\acronymentry}[1]{%
      \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }

short-long-desc <short> ({<long>}) acronym style that has an accompanying description (which
the user needs to supply).
  \renewacronymstyle{short-long-desc}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
    \renewcommand*\acronymentry}[1]{%
      \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }

sc-short-long-desc <long> (\textsc{<short>}) acronym style that has an accompanying description
(which the user needs to supply).
  \renewacronymstyle{sc-short-long-desc}%
  {%
    \GlsUseAcrEntryDispStyle{sc-short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{sc-short-long}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
  }

```

```

\renewcommand*\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))%
}

```

sm-short-long-desc *⟨long⟩* (`\textsmaller{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))%
  }
}

```

dua *⟨long⟩* only acronym style.

```

\renewacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
  \ifglsahaslong{\glslabel}%
  {%
    \glsifplural
    {%

```

Plural form:

```

\gls caps case
{%

```

Plural form, don't adjust case:

```

\gls long plural access display {\glsentrylongpl{\glslabel}}{\glslabel}%
\gls insert
}%
{%

```

Plural form, make first letter upper case:

```

\gls long plural access display {\Glsentrylongpl{\glslabel}}{\glslabel}%
\gls insert
}%
{%

```

Plural form, all caps:

```

\gls long plural access display
{\glsupercase{\glsentrylongpl{\glslabel}}}{\glslabel}%

```

```

        \glsuppercase{\glsinsert}%
    }%
}%
{%-
Singular form
    \glscapscase
    {%-
Singular form, don't adjust case:
    \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%-
Subsequent singular form, make first letter upper case:
    \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%-
Subsequent singular form, all caps:
    \glslongaccessdisplay
    {\glsuppercase
     {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
    \glsuppercase{\glsinsert}%
    }%
}%
{%-
Not an acronym:
    \glsgenentryfmt
    }%
}%
{\glscustomtext\glsinsert}%
}%
{%-
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay
        {\glsuppercase{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%-
    \glslink[##1]{##2}{%-

```



```

\glslongpluralaccessdisplay
  {\glsentrylongpl{##2}}{##2}##3\space
(\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsupercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%

```

}%

footnote *<short>*\footnote{*<long>*} acronym style.

```
\renewacronymstyle{footnote}%
```

{%

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
```

}%

{%

```
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

Need to ensure hyperlinks are switched off on first use:

```
\glshyperfirstfalse
```

```
\renewcommand*{\genacrfullformat}[2]{%
```

```
\glsshortaccessdisplay
```

```
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
```

}%

```
\renewcommand*{\Genacrfullformat}[2]{%
```

```
\glsshortaccessdisplay
```

```
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
```

}%

```
\renewcommand*{\genplacrfullformat}[2]{%
```

```
\glsshortpluralaccessdisplay
```

```
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
```

}%

```
\renewcommand*{\Genplacrfullformat}[2]{%
```

```
\glsshortpluralaccessdisplay
```

```
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
```

}%

```
\renewcommand*{\acronymentry}[1]{%
```

```
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
```

```
\renewcommand*{\acronymsort}[2]{##1}%
```

```
\renewcommand*{\acronymfont}[1]{##1}%
```

```
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
```

Don't use footnotes for \acrfull:

```
\renewcommand*{\acrfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
```

```
(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%
```

```
\renewcommand*{\Acrfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```
\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
```

```
(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%
```

```
\renewcommand*{\ACRfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```

\glsshortaccessdisplay
  {\glsupercase
   {\acronymfont{\glsentryshort{##2}}{##2}##3\space
   (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
   \glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
   \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##2}}{##2}##3\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
   \glsshortpluralaccessdisplay
    {\glsupercase
     {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
     (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}}}%

```

Similarly for `\glsentryfull` etc:

```

\renewcommand*\glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##1}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
  {\acronymfont{\Glsentryshortpl{##1}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
}

```

`footnote-sc \textsc{<short>}\footnote{<long>}` acronym style.

```

\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glsstextup{\glspluralsuffix}}%
}%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
```

```
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
```

```
\glsdefaultshortaccess{\langle long\rangle}{\langle short\rangle}
```

`\glsdefaultshortaccess`

Default shortaccess value.

```
\newcommand*\glsdefaultshortaccess}[2]{#1}
```

Use `\newacronymhook` to modify the key list to set the access text to the long version by default.

```
\renewcommand*\newacronymhook{%
\protected@edef\gls@keylist{%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
\the\glskeylisttok}%
\expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
```

`\DefaultNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionFootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\SmallNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

The following are kept for compatibility with versions before 3.0:

`\glsshortaccesskey`

```
\newcommand*\glsshortaccesskey{\glsshortkey access}%
```

`\glsshortpluralaccesskey`

```
\newcommand*\glsshortpluralaccesskey{\glsshortpluralkey access}%
```

`\glslongaccesskey`

```
\newcommand*\glslongaccesskey{\glslongkey access}%
```

```
\glslongpluralaccesskey
    \newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

6.5 Debugging Commands

```
\showglongnameaccess
    \newcommand*{\showglongnameaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
    }
```

```
\showglongtextaccess
    \newcommand*{\showglongtextaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
    }
```

```
\showglongpluralaccess
    \newcommand*{\showglongpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
    }
```

```
\showglongfirstaccess
    \newcommand*{\showglongfirstaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
    }
```

```
\showglongfirstpluralaccess
    \newcommand*{\showglongfirstpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
    }
```

```
\showglongsymbolaccess
    \newcommand*{\showglongsymbolaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
    }
```

```
\showglongsymbolpluralaccess
    \newcommand*{\showglongsymbolpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
    }
```

```
\showglongdescaccess
    \newcommand*{\showglongdescaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
    }
```

```
\showglongdescpluralaccess
    \newcommand*{\showglongdescpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
    }
```

```
\showgloshortaccess
```

```
\newcommand*{\showgloshortaccess}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@shortaccess\endcsname  
}
```

```
\showgloshortpluralaccess
```

```
\newcommand*{\showgloshortpluralaccess}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@shortpluralaccess\endcsname  
}
```

```
\showglolongaccess
```

```
\newcommand*{\showglolongaccess}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@longaccess\endcsname  
}
```

```
\showglolongpluralaccess
```

```
\newcommand*{\showglolongpluralaccess}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@longpluralaccess\endcsname  
}
```

7 Backwards Compatibility

7.1 glossaries-compatible-207

Provides compatibility with version 2.07 and below. This uses original glossaries xindy and makeindex formatting, so can be used with old documents that had customized style files, but hyperlinks may not work properly.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-207-2020-03-19.sty}  
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-207-2021-11-01.sty}  
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2025/04/14 v4.6 (NLCT)]
```

Trigger error and that's it.

```
\PackageError{glossaries}  
{glossaries-compatible-207.sty no longer available. Use rollback if you need it}%  
{Try \string\usepackage{glossaries}[=v4.46]}
```

7.2 glossaries-compatible-307

Provides compatibility with version 3.07 and below.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-307-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-307-2021-11-01.sty}
\DeclareCurrentRelease{v4.6}{2025-04-14}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2025/04/14 v4.6 (NLCT)]
```

Trigger error and that's it.

```
\PackageError{glossaries}
{glossaries-compatible-307.sty no longer available. Use rollback if you need it}%
{Try \string\usepackage{glossaries}[=v4.46]}
```

8 Rollback

8.1 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}%
{
  \@gls@docloadedtrue
}%
{
  \@gls@docloadedfalse
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
```



```

\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\gls@debug@nr\relax
% debug=true
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\@glsshowtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
\nfss@text{\glsshowtargetfont [#1]}%
\else
\ifinner
{\glsshowtargetfont [#1]}%

```

```

\else
  \glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetouter[1]{%
  \glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbolsymbol{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget[1]{%
\newrobustcmd*\@@glsshowtarget[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp[3]{%
\newrobustcmd*\glsshowaccsupp[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error, warn, ignore}{%
  \ifcase\gls@seenoinde@nr
  \renewcommand*\@gls@see@noindex{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key}%
  }%
}

```

```

\or
  \renewcommand*\@gls@see@noindex}{%
    \GlossariesWarning{'\gls@xr@key' key ignored}%
  }%
\or
  \renewcommand*\@gls@see@noindex}{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec}{section}}%
  {\newcommand*\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec{#1}}
\newcommand*\@@glossarysecstar}{*}%
\newcommand*\@@glossaryseclabel}{%
\newcommand*\glsautoprefix{}%
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
    \label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}

```

```

\@gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\glo@seeautonumberlist}{%
    \def\glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationstrue
\newcommand*\gls@loadlong{\RequirePackage{glossary-long}[=v4.46]}
\@gls@declareoption{nolong}{\renewcommand*\gls@loadlong{}}
\IfFileExists{supertabular.sty}{%
  \newcommand*\gls@loadsuper{\RequirePackage{glossary-super}[=v4.46]}%
  \newcommand*\gls@loadsuper{}}
\@gls@declareoption{nosuper}{\renewcommand*\gls@loadsuper{}}
\newcommand*\gls@loadlist{\RequirePackage{glossary-list}[=v4.46]}
\@gls@declareoption{nolist}{%
  \renewcommand*\gls@loadlist}{%
    \ifdefstring{\glossary@default@style}{list}%
    {\let\glossary@default@style\relax}%
  }%
}
\newcommand*\gls@loadtree{\RequirePackage{glossary-tree}[=v4.46]}
\@gls@declareoption{notree}{\renewcommand*\gls@loadtree{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*\gls@loadlong}{%
  \renewcommand*\gls@loadsuper}{%
  \renewcommand*\gls@loadlist}{%
  \renewcommand*\gls@loadtree}{%
  \let\glossary@default@style\relax
}
\newcommand*\gls@postdescription}{%
  \ifgls@postdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
\gls@nopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{%
\gls@nogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{%
\@ifclassloaded{memoir}
{%
  \gls@ucmarktrue
}%
}
}
\gls@ucmarkfalse

```

```

}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
    \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  }%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
    \newcounter{glossarysubentry}[glossaryentry]%
    \else
    \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%

```

```

\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
  \ifx\@glo@sort\@glsdefaultsort
    \let\@glo@sort\@glo@name
  \fi
  \let\glsdosanitizesort\@gls@sanitizesort
  \glsprestandardsort{\@glo@sort}{##1}{##2}%
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}
\newcommand*\@gls@setupsort@def{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \ifcsundef{glossary@##1@sortcount}%
  {\@gls@defsortcount{##1}}%
  {}%
  \expandafter\global\expandafter
  \advance\csname glossary@##1@sortcount\endcsname by 1\relax
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
    \expandafter\glsortnumberfmt
    {\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%

```

```

\edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\edef\@glo@type{\csname glo@##1@type\endcsname}%
\edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected\edef\csname glo@##1@sort\endcsname{%
\expandafter\gls@sortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}{gls}{acronym}[true]{%
\ifglsacronym
\renewcommand{\@gls@do@acronymsdef}{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
}

```

```

        \renewcommand*\acronymtype}{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
\else
    \let\gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
    \ifglsacronym
        \ifbool{glscompatible-3.07}%
            {}%
            {%
                \providecommand*\printacronyms}[1] []{%
                    \printglossary[type=\acronymtype,#1]}%
            }%
        \fi
    }
\newcommand*\gls@do@acronymsdef{}
\gls@declareoption{acronyms}{%
    \glsacronymtrue
    \def\gls@do@acronymsdef{%
        \DeclareAcronymList{acronym}%
        \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
        \renewcommand*\acronymtype}{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
}
\newcommand*\glsacronymlists{}
\newcommand*\@addtoacronymlists}[1]{%
    \ifx@glsacronymlists\@empty
        \protected@xdef\glsacronymlists{#1}%
    \else
        \protected@xdef\glsacronymlists{\glsacronymlists,#1}%
    \fi
}
\newcommand*\DeclareAcronymList}[1]{%
    \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
    \edef\do@gls@islistofacronyms{%
        \noexpand@gls@islistofacronyms{#1}{\glsacronymlists}}%
    \do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
    \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
        \def\@gls@before{##1}\def\@gls@after{##2}}%
}

```



```

\gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
\ifx\@gls@after\@nnil
#4%
\else
#3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
\glsIfListOfAcronyms{#1}%
{\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists}[1]{%
\renewcommand*\@gls@acronymlists{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
\DeclareAcronymList{#1}%
}
\newcommand*\@gls@counter{page}
\define@key{glossaries.sty}{counter}{%
\renewcommand*\@gls@counter{#1}%
}
\newcommand*\@gls@nohyperlist{}
\newcommand*\@GlsDeclareNoHyperList}[1]{%
\ifdefempty\@gls@nohyperlist
{%
\renewcommand*\@gls@nohyperlist{#1}%
}%
{%
\appto\@gls@nohyperlist{,#1}%
}%
}
\define@key{glossaries.sty}{nohypertypes}{%
\GlsDeclareNoHyperList{#1}%
}
\newcommand*\@GlossariesWarning}[1]{%
\PackageWarning{glossaries}{#1}%
}
\newcommand*\@GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{#1}%
}
\newcommand*\@glosortentrieswarning{%
\typeout{Using TeX to sort glossary entries---this may
take a while}%
}
\@gls@declareoption{nowarn}{%
\if@gls@debug
\GlossariesWarning{Warnings can't be suppressed in debug mode}%
\else
\renewcommand*\@GlossariesWarning}[1]{%

```

```

        \renewcommand*\GlossariesWarningNoLine}[1]{}%
        \renewcommand*\glosortentrieswarning}{}%
        \renewcommand*\@gls@missinglang@warn}[2]{}%
    \fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
    \PackageWarningNoLine{glossaries}%
    {No language module detected for ‘#1’.\MessageBreak
    Language modules need to be installed separately.\MessageBreak
    Please check on CTAN for a bundle called\MessageBreak
    ‘glossaries-#2’ or similar}%
}
\@gls@declareoption{no-langwarn}{%
    \renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined}{%
    \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheglossdefined}{%
    \GlossariesWarning{Overriding ‘theglossary’ environment}%
}
\@gls@declareoption{no-undefwarn}{%
    \renewcommand*\@gls@warnonglossdefined}{}%
    \renewcommand*\@gls@warnontheglossdefined}{}%
}
\newcommand*\@gls@sanitizedesc}{%
}
\newcommand*\glssetexpandfield}[1]{%
    \csdef{gls@assign@#1@field}##1##2{%
        \@gls@expand@field{##1}{#1}{##2}%
    }%
}
\newcommand*\glssetnoexpandfield}[1]{%
    \csdef{gls@assign@#1@field}##1##2{%
        \@gls@noexpand@field{##1}{#1}{##2}%
    }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{desclplural}
\newcommand*\@gls@sanitizename}{%
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol}{%
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort}{%
    \ifgls@sanitizesort
        \@gls@sanitizesort
    \else
        \@gls@nosanitizesort
    \fi
}

```

```

\fi
}
\newcommand*{\@@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@nosanitizesort}{%}
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*{\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x@\@secondoftwo
  \let\@tabacckludge@\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let^\@firstofone
  \let"@@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let.\@firstofone

```

```

\let~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifglS@sanitize@symbol
\glssetnoexpandfield{symbol}%

```

```

        \glsssetnoexpandfield{symbolplural}%
    \else
        \glsssetexpandfield{symbol}%
        \glsssetexpandfield{symbolplural}%
    \fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesorttrue
            \glsssetnoexpandfield{sortvalue}%
        }%
    \else
        \glsssetexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesortfalse
            \glsssetexpandfield{sortvalue}%
        }%
    \fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
}
\GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%

```

```

}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@gls@usetranslator{%
      \@ifpackageloaded{polyglossia}%
      {%
        \let\glsifusetranslator\@secondoftwo
      }%
    }%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
}

```

```

        }%
        {}%
    }%
}
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\@endfortrue
}%
{}%
}
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{%
\ifbool{glsacrdescription}%
{}%
{%
\renewcommand*{\@gls@sanitizedesc}{}}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%

```

```

\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{}
\glsxindy@glsnumberstrue
\def\@xdy@main@language{\languagename}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{}
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\glsxindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect

```



```

        (\string\makeglossaries\space has already been used)}%
    }%
    {%
    \renewcommand{\@domakeglossaries}[1]{%
        \PackageInfo{glossaries}{\string\makeglossaries\space and
        \string\makenoidxglossaries\space have been disabled}%
    }%
    }%
    }%
}
\@gls@declareoption{restoremakegloss}{%
    \ifdefequal\makeglossaries\@no@makeglossaries
    {%
        \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makenoidxglossaries\space has already been used)}%
    }%
    {%
        \ifdefequal\@makeglossary\@gobble
        {%
            \GlossariesWarning{Option ‘restoremakegloss’ has no effect
            (\string\makeglossaries\space has already been used)}%
        }%
        {%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been restored}%
            \let\@domakeglossaries\@firstofone
        }%
    }%
}
}
\newcommand*\@do@write@glslabels{%
    \AtEndDocument{\@do@write@glslabels}%
    \let\@do@write@glslabels\relax
}
\newcommand*\@do@write@glslabels@file{%
    \newwrite\@gls@labelsfile
    \immediate\openout\@gls@labelsfile=\jobname.glslabels
    \forallglsentries[\@glo@types,\@ignored@glossaries]{\@glsentry}%
    {\ifdefempty{\@glsentry}{\immediate\write\@gls@labelsfile{\@glsentry}}}%
    \immediate\closeout\@gls@labelsfile
}
\@gls@declareoption{writeglslabels}{\@do@write@glslabels}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
    [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
    \ifnum\gls@automake@nr=1\relax
        \glsautomakefalse
    \else
        \glsautomaketrue
    \fi
}

```

```

\ifglsautomake
  \renewcommand*{\@gls@doautomake}{%
    \PackageError{glossaries}{You must use
    \string\makeglossaries\space with automake=true}
    {%
      Either remove the automake=true setting or
      add \string\makeglossaries\space to your document preamble.%
    }%
  }%
\else
  \renewcommand*{\@gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\gls savewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}{%
    {%
      \booltrue{glscompatible-3.07}%
    }%
  }%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
instead. If you \MessageBreak need the original definition of
\string\makeglossary\space use \MessageBreak the package options
kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict)}}%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed

```

```

with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%
\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]}%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef

```

```

}
\newcommand*{\@gls@do@numbersdef}{}
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{}
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
  \newcommand*{\newterm}[2][%
    \newglossaryentry{##2}{%
      {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \@expandtwoargs
      \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
    \ifin@
      \@use@ption
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.46]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts

```

```

        \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
\ifglsentrycounter
    \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}{\glsupertextup{\glsacrpluralsuffix}}
\providecommand*{\seename}{see}
\providecommand*{\andname}{\&}
\newcommand*{\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}}%
}
\newcommand*{\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.1df}%
}
\newcommand*{\addglossarytocaptions}[1]{%
\ifglstranslate
  \RequirePackage{tracklang}
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*{\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{%
          {%
            \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
            \expandafter\toks@\expandafter{\@gls@tmp

```

```

        \renewcommand*{\glossaryname}{\translate{Glossary}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
} %
} %
{} %
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
} %
{} %
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
} %
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc{}

```

```

\let\org@glspostdescription\glspostdescription
\def\glspostdescription{%
  \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{}\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*\glstfilebase{#1}%
  \ifglxindy
    \def\istfilename{\glstfilebase.xdy}
  \else
    \def\istfilename{\glstfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\glstfilebase.xdy}
\else
  \def\istfilename{\glstfilebase.ist}
\fi
\newcommand*\glstfilebase{\jobname}
\newcommand*\@istfilename}[1]{
\newcommand*\glscpositor}{.}
\newcommand*\glstSetCompositor}[1]{%
  \renewcommand*\glscpositor{#1}}
\@onlypremakeg\glstSetCompositor
\newcommand*\@glstAlphacompositor{\glscpositor}
\ifglxindy
  \newcommand*\glstSetAlphaCompositor[1]{%
    \renewcommand*\@glstAlphacompositor{#1}}
\else
  \newcommand*\glstSetAlphaCompositor[1]{%
    \glstnoxywarning\glstSetAlphaCompositor}
\fi
\@onlypremakeg\glstSetAlphaCompositor
\newcommand*\glstsuffixF{}
\newcommand*\glstSetSuffixF}[1]{%
  \renewcommand*\glstsuffixF{#1}}
\@onlypremakeg\glstSetSuffixF
\newcommand*\glstsuffixFF{}
\newcommand*\glstSetSuffixFF}[1]{%
  \renewcommand*\glstsuffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glstnumberformat}[1]{#1}%
}%
{%
  \newcommand*\glstnumberformat}[1]{\glsthypernumber{#1}}%
}

```



```

\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%

```

```

\else
  \@mkboth{#1}{#1}%
\fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*\@glossarysection[2]{%
\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname[#1]{#2}%
}%
{%
  \csname\@glossarysec\endcsname*{#2}%
  \gls@toc{#1}{\@glossarysec}%
}%
\@glossaryseclabel
}
\newcommand*\@pglossarysection[2]{%
\gls@clearpage
\phantomsection
\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname{#2}%
}%
{%
  \gls@toc{#1}{\@glossarysec}%
  \csname\@glossarysec\endcsname*{#2}%
}%
\@glossaryseclabel
}
\newcommand*\gls@docclearpage{%
\ifthenelse{\equal{\@glossarysec}{chapter}}{%
{%
  \ifcsundef{cleardoublepage}%
  {%
    \clearpage
  }%
  {%
    \ifcsdef{if@openright}%
    {%
      \if@openright
        \cleardoublepage
      \else
        \clearpage
      \fi
    }%
  }%
  \cleardoublepage
}%
}

```

```

}%
{}%
}
\newcommand*\glsclearpage{\gls@doclearpage}
\newcommand*\@gls@toc}[2]{%
  \ifglstoc
    \ifglsnumberline
      \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
    \else
      \addcontentsline{toc}{#2}{#1}%
    \fi
  \fi
}
\newcommand*\glsnoxindywarning}[1]{%
  \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
  \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
  \edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
  \edef\xdyattributelist{}%
\fi
\ifglsxindy
  \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
  \def\@do@ifinlist##1,#1,##2\end@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
  \newcommand*\@xdycounters{\glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \@for\@gls@ctr:=#1\do{%
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
    }%
  }%

```

```

        \do@addcounter
    }
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
\glsnoxywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glssaddxdycounters{%
\PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglxsindy
\newcommand*\@glssaddxyattribute[2]{%
\edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{##2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
\ifx\@xdyattributelist\@empty
\edef\@xdyattributelist{#1}%
\else
\edef\@xdyattributelist{\@xdyattributelist,#1}%
\fi
\@for\@this@counter:=\@xdycounters\do{%
\protected@edef\gls@do@addxyattribute{%
\noexpand\@glssaddxyattribute{#1}{\@this@counter}%
}
\gls@do@addxyattribute
}%
\let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxywarning\GlsAddXdyAttribute}
\fi

```

```

\ifglxindy
\newcommand*{\@gls@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{glsnumber}
  \GlsAddXdyAttribute{hyperm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{glsignore}
}
\else
  \let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^^J
      (define-alphabet "#1" (#2))}
  \else
    \newcommand*{\GlsAddXdyAlphabet}[2]{%
      \glsnoxywarning\GlsAddXdyAlphabet}
  \fi
\ifglxindy
  \newcommand*{\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"

```

```

        \string"roman-numbers-lowercase\string" :sep \string}}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string}}}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
    \string"roman-numbers-lowercase\string"%
  }%
\else
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
    :sep \string"\@gls@roman\string"%
  }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
:sep \string"\@gls@Alphacompositor\string"
\string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][[]]{%
  \def\@gls@tmp{#1}%
  \ifx\@gls@tmp\@empty
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%

```

```

        (define-location-class \string"#2\string"^^J\space\space
        \space(:sep \string"{}\glsoopenbrace\string" #3
        :sep \string"\glsclosebrace\string"))
    }%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsoopenbrace"
#1
:sep "\glsclosebrace\glsoopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation[2]{%
\glsnoinxindwarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnoinxindwarning\GlsSetXdyLocationClassOrder}
\fi
\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\@xdy@language#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][{}]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage\@empty
      \ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
    \fi
  }
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}

```



```

\fi
\ifglxsindy
  \ifglxs@xindy@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
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glslacronymlists\do{\ifx#1\empty\else#2\fi}%
}
\newcommand*\forallglsentries[3][\@glsldefaulttype]{%
  \edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglsentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forallglsentries[\@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel[1]{#1}
\newcommand{\ifglsentryexists}[3]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \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}%
    \edef@gl@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@gl@entries\expandafter
      [\csname glo@\@gl@thislabel @type\endcsname]
      {\glo@label}%
    {%
      \letcs\glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gl@thislabel\glo@parent
        {%

```

```

        \def\do@glshaschildren{#2}%
        \@endfortrue
    }%
    {}%
    }%
    \do@glshaschildren
}
}
\newcommand{\ifglshasparent}[3]{%
    \glsdoifexists{#1}%
    {%
        \ifcsempy{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
    }%
}
\newcommand*{\ifglshasdesc}[3]{%
    \ifcsempy{glo@\glsdetoklabel{#1}@desc}%
    {#3}%
    {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
}
}
\newrobustcmd*{\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
        \ifdefequal\@glo@symbol\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
        \ifdefequal\@glo@long\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
        \ifdefequal\@glo@short\@gls@default@value

```

```

    {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
    {%
      \ifdefempty\@glo@thisvalue
      {%
        #4%
      }%
    }%
    {%
      \ifdefequal\@glo@thisvalue\@gls@default@value
      {%
        #4%
      }%
    }%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \@gls@fetchfield{\@gls@thisfield}{#1}%
  \ifdef\@gls@thisfield
  {%
    \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  {%
    \ifdefequal\@glo@thisvalue\@gls@default@value
    {%
      #4%
    }%
  }%
  \let\glscurrentfieldvalue\@glo@thisvalue
  #3%
}%
}%
}%
\GlossariesWarning{Unknown entry field '#1'}%
#4%
}%
}%

```

```

}%
}
\newcommand*\glscurrentfieldvalue}{
\newcommand*\@glo@types}{,}
\newcommand*\@gls@provide@newglossary{%
  \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
  \let\@gls@provide@newglossary\relax
}
\newcommand*\defglsentryfmt}[2][\glsdefaulttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@forbidtextext}[1]{%
  \ifboolexpr{test \ifdefstring{#1}{tex}}
    or test \ifdefstring{#1}{TEX}}
  {%
    \def#1{nottex}%
    \PackageError{glossaries}%
      {Forbidden '.tex' extension replaced with '.nottex'}%
      {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
        Don't use '.tex' as an extension for a temporary file.}%
  }%
  {%
  }%
}
\newcommand*\@gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}]}
\def\@gls@gobbleopt[#1]{
\newcommand*\@newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaulttype
    {%
      \gdef\glsdefaulttype{#2}%
    }-}%
  \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
  \expandafter\gdef\csname glolist@#2\endcsname{,}%
  \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
  \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
  \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
  \expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
  \expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
  \expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
  \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
  \@gls@provide@newglossary
  \protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}%
  \ifcsundef{gls@#2@entryfmt}%

```

```

{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}{\glscounter}}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \edef\@ignored@glossaries{#1}%
  }%
  {%
    \eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist}{#1}%
  }%

```

```

    {%
      \eappto\@gls@nohyperlist{,#1}%
    }%
  }
  \newcommand*\@ignored@glossaries{}
  \newcommand*\ifignoredglossary}[3]{%
    \edef\@gls@igtype{#1}%
    \expandafter\DTLifinlist\expandafter
      {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
  }
  \define@key{glossentry}{name}{%
    \def\@glo@name{#1}%
  }
  \define@key{glossentry}{description}{%
    \def\@glo@desc{#1}%
  }
  \define@key{glossentry}{descriptionplural}{%
    \def\@glo@descplural{#1}%
  }
  \define@key{glossentry}{sort}{%
    \def\@glo@sort{#1}}
  \define@key{glossentry}{text}{%
    \def\@glo@text{#1}%
  }
  \define@key{glossentry}{plural}{%
    \def\@glo@plural{#1}%
  }
  \define@key{glossentry}{first}{%
    \def\@glo@first{#1}%
  }
  \define@key{glossentry}{firstplural}{%
    \def\@glo@firstplural{#1}%
  }
  \newcommand*\@gls@default@value{\relax}
  \define@key{glossentry}{symbol}{%
    \def\@glo@symbol{#1}%
  }
  \define@key{glossentry}{symbolplural}{%
    \def\@glo@symbolplural{#1}%
  }
  \define@key{glossentry}{type}{%
    \def\@glo@type{#1}}
  \define@key{glossentry}{counter}{%
    \ifcsundef{c@#1}%
      {%
        \PackageError{glossaries}%
          {There is no counter called ‘#1’}%
          {%
            The counter key should have the name of a valid counter
            as its value%
          }
      }
    }

```

```

    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key[3]{%
  \renewcommand*\@gls@xr@key{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*\@gls@xr@key{see}
\newcommand*\@gls@checkseeallowed{%
  \@gls@see@noindex
}
\newcommand*\@gls@checkseeallowed@preambleonly{%
  \GlossariesWarning{glossaries}%
  {'\@gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
  [\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]}%
{%
  \ifcase\@gls@nonumberlist@nr\relax
    \def\@glo@prefix{\@gls@nonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@gls@nextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*\@gls@savenonumberlist[1]{}
\newcommand*\@gls@initnonumberlist{}%
\newcommand*\@gls@storenonumberlist[1]{}
\newcommand*\@gls@enablesavenonumberlist{%
  \renewcommand*\@gls@initnonumberlist{}%
  \undef\@glo@nonumberlist
}%
\renewcommand*\@gls@savenonumberlist[1]{%
  \def\@glo@nonumberlist{##1}%
}%
\renewcommand*\@gls@storenonumberlist[1]{%

```



```

\ifdef\@glo@nonumberlist
{%
  \cslet{glo\@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
}%
{}%
}%
\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}

```

```

}%
}%
\newcommand*{\@glsdefaultplural}{ }
\newcommand*{\@gls@missingnumberlist}[1]{%
??%
\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}
  {%
    \ifdequal{#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}%
      }%
    }%
  }%
}
\ifdequal{#4}{\@gls@default@value}%
{%
  \@gls@expand@field{#2}{#3}{#1}%
}%
{%
  \@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\gls@expandfields{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*\gls@noexpandfields{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*\gls@defdocnewglossaryentry{%
  \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
  \let\newglossaryentry\new@glossaryentry
}

```

```

\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
  \ifglentryexists{#1}{}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
  \@gls@writedef{#1}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\@gls@begindocdefs{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{%}{%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\ifglentryexists{#1}{}\glspercentchar^^J%
    \expandafter\@gobble\string{\glspercentchar^^J%
      \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
      \expandafter\@gobble\string{\glspercentchar%
    }%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%
      }%
    }%
  }%
}

```

```

        \glspercentchar
    }%
    }%
    {}%
}%
\gls.writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar%
}%
}
\newcommand*\@gls@keymap{%
    {name}{name},%
    {sort}{sortvalue},% unescaped sort value
    {type}{type},%
    {first}{first},%
    {firstplural}{firstpl},%
    {text}{text},%
    {plural}{plural},%
    {description}{desc},%
    {descriptionplural}{descplural},%
    {symbol}{symbol},%
    {symbolplural}{symbolplural},%
    {user1}{useri},%
    {user2}{userii},%
    {user3}{useriii},%
    {user4}{useriv},%
    {user5}{userv},%
    {user6}{uservi},%
    {long}{long},%
    {longplural}{longpl},%
    {short}{short},%
    {shortplural}{shortpl},%
    {counter}{counter},%
    {parent}{parent}%
}
\newcommand*\@gls@fetchfield}[2]{%
    \edef\@gls@thisval{#2}%
    \for\@gls@map:=\@gls@keymap\do{%
        \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
        \ifdefequal{\@this@key}{\@gls@thisval}%
        {%
            \edef#1{\expandafter\@secondoftwo\@gls@map}%
            \@endfortrue
        }%
    }%
}%
}

```

```

\newcommand*\glsaddstoragekey}{\ifstar\sglsaddstoragekey\glsaddstoragekey}
\newcommand*\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddstoragekey{#1}%
}
\newcommand*\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  {}%
  \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newcommand*\glsaddkey}{\ifstar\sglsaddkey\glsaddkey}
\newcommand*\@sglsaddkey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddkey{#1}%
}
\newcommand*\@glsaddkey}[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
  }%
}

```

```

\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
   '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1@\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[[]]}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1@\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
   '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2}[[]]}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2][]{%

```

```

        \new@ifnextchar [%
            {\csuse{@GLS@user@#1@}{##1}{##2}}%
            {\csuse{@GLS@user@#1@}{##1}{##2}[]}%
        \csdef{@GLS@user@#1@}##1##2[##3]{%
            \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
        }%
        \newrobustcmd*{#7}{%
            \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
        }%
    }%
}
{
\PackageError{glossaries}{Key ‘#1’ already exists}{%
}
}
}
\newcommand{\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*{\gls.writedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{}{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
      \let\@org@newglossaryentryprehook\@newglossaryentryprehook
      \long\def\@newglossaryentryprehook{%
        \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
        \@org@newglossaryentryprehook
      }%
      \renewcommand*{\gls@assign@desc}[1]{%
        \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
        \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
      }
      \gls@defglossaryentry{#1}{#2}%
    \egroup
  }
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{}%
  {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \edef\glo@label{\glsdetoklabel{#1}}%

```

```

\let\glslabel\@glo@label
\let\@glo@name\@gls@name
\let\@glo@desc\@gls@desc
\let\@glo@descplural\@gls@descplural
\let\@glo@type\@gls@type
\let\@glo@symbol\@gls@symbol
\let\@glo@symbolplural\@gls@symbolplural
\let\@glo@text\@gls@text
\let\@glo@plural\@gls@plural
\let\@glo@first\@gls@first
\let\@glo@firstplural\@gls@firstplural
\let\@glo@sort\@gls@sort
\let\@glo@counter\@gls@counter
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonnumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
  a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
  in it}%
}%
{}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@gls@desc

```

```

        \let\@glo@desc\@empty
    \fi
}%
{
}%
\protected@edef\@glo@list@\csname glo@list@\@glo@type\endcsname}%
\expandafter\xdef\csname glo@list@\@glo@type\endcsname{%
    \@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
    \ifdefequal\@glo@label\@glo@parent%
        {
            \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
            \def\@glo@parent{}%
            \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
        }%
        {
            \ifglsentryexists{\@glo@parent}%
            {
                \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
                    \@glo@parent}%
                \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
                \advance\gls@level by 1\relax
                \ifx\@glo@name\@glsnoname
                    \expandafter\let\expandafter\@glo@name
                    \csname glo@\@glo@parent @name\endcsname
                \ifx\@glo@plural\@gls@default@value
                    \expandafter\let\expandafter\@glo@plural
                    \csname glo@\@glo@parent @plural\endcsname
                \fi
            }%
        }%
    }%
    \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}{%
\ifdefvoid\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%

```

```

\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\@glo@list{,%}
  \for@gl@entries[\@glo@type]{\@glo@label}%
  {%
    \if@defequal\@glo@thislabel\@glo@label
      {}{\@eappto\@glo@list{\@glo@label,}}%
    }%
  \cslet\@glo@list[\@glo@type]{\@glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\if@gl@xindy
  \newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\@glossentry}
\fi
\if@gl@xindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\@subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string\@subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
  \edef\@glo@esclabel{#1}%
  \@gls@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gls@checkmkidxchars\@glo@sort
  \@gls@checkmkidxchars\@glo@prefix
  \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
  \if@gl@xindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string\@glo@sort\string" %
          \string\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
        }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string\@glo@sort\string" %
          \string\@glo@prefix\@glossarysubentryfield
            {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
        }%
    \fi
  \else
    \ifx\@glo@parent\@empty

```

```

        \onelevel@sanitize\@glo@prefix
        \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
            \@glo@sort\@gls@actualchar\@glo@prefix
            \@glossaryentryfield{\@glo@esclabel}%
        }%
    \else
        \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
            \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
            \@glo@sort\@gls@actualchar\@glo@prefix
            \@glossarysubentryfield
                {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
            }%
        \fi
    \fi
}
\AtBeginDocument{%
    \ifpackageloaded{amsmath}%
    {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
    {}%
}
\newcommand*\@gls@ifnotmeasuring[1]{%
    \ifmeasuring@
    \else
    #1%
    \fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
    \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
    \ifdef\TX@trial
    {%
        \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
        \let\glspatchtabularx\relax
    }%
    {}%
}
\newcommand*\glsreset[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glsreset{#1}%
        }%
    }%
}
\newcommand*\glslocalreset[1]{%
    \gls@ifnotmeasuring
    {%

```



```

\glsdoifexists{#1}%
{%
  \@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%
\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glsunset{#1}%
  }%
}%
}
\newcommand*\glslocalunset}[1]{%
\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glslocalunset{#1}%
  }%
}%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}

```

```

}%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*\@glslocalreset}[1]{%

```

```

    \@glslocalreset{##1}%
    \csdef{glo@glsdetoklabel{##1}@currcount}{0}%
  }%
\def\@cgl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cgl@spl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cgl@splformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@spl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGls@splformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
}

```

```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*\string\@gls@entry@count}[2]{}}%
\forallglsentries{\@glsentry}{%
  \ifglsused{\@glsentry}%
  {\immediate\write\@auxout
    {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
  }%
}%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cglS}{\@gls@hyp@opt\@cglS}
\newcommand*{\@cglS}[2][{}]{%
  \new@ifnextchar[{\@cglS@{#1}{#2}}{\@cglS@{#1}{#2}}[{}]}%
}
\def\@cglS@#1#2[#3]{%
  \GlossariesWarning{\string\cglS\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@cglS@{#1}{#2}[#3]}%
}
\newcommand*{\cglSformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2][{}]{%
  \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}}[{}]}%
}
\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@Gls@{#1}{#2}[#3]}%
}
\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglSpl}{\@gls@hyp@opt\@cglSpl}
\newcommand*{\@cglSpl}[2][{}]{%
  \new@ifnextchar[{\@cglSpl@{#1}{#2}}{\@cglSpl@{#1}{#2}}[{}]}%
}
\def\@cglSpl@#1#2[#3]{%
  \GlossariesWarning{\string\cglSpl\space is defaulting to
    \string\glSpl\space since you haven't enabled entry counting}%
  \@glSpl@{#1}{#2}[#3]}%
}
\newcommand*{\cglSplformat}[2]{%
  \ifglshaslong{#1}{\glSentrylongpl{#1}}{\glSentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}}[{}]}%
}

```

```

}
\def\cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glsformat}[1]{#1}
\newcommand*\glsentryfmt{%
  \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*\@@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
  }%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
  }%
}

```

```

#2{\Glsentryplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}%
}%
}%
{%
\ifbool{glscompatible-3.07}%
{%
\protected@edef\@glo@etext{%
#1{\Glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
\xmakefirstuc\@glo@etext
}%
{%
#1{\Glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}%
}%
}%
}%
}%
\ifglsused\glslabel
{%
\mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
{%
\mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
}%
}%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glsentrytext{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
{%
#1{\glsentryfirst{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
}%
}

```

```

{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #1{\Glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
}
{%
  \ifglsused{\glslabel}%

```

```

    {%
      #2{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glsentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glsentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
  }%
}

```



```

{%
  \ifglused\glslabel
  {%
    \glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\newcommand*{\glsnacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```



```

\newcommand*\genacrfullformat}[2]{%
  \glsentrylong{#1}#2\space
  (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\glsdisplayfirst}[4]{#1#4}
\newcommand*\glsdisplay}[4]{#1#4}
\newcommand*\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```

```

    }%
    {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
    }%
    }%
    }%
    \@gls@doentrydef
}
\define@key{glslink}{counter}{%
    \ifcsundef{c@#1}%
    {%
        \PackageError{glossaries}%
        {There is no counter called ‘#1’}%
        {%
            The counter key should have the name of a valid counter
            as its value%
        }%
    }%
    }%
    {%
        \def\@gls@counter{#1}%
    }%
}
\define@key{glslink}{format}{%
    \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
    \glslinkvar{#1}{#2}{#1}%
    \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
        you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
    \let\glslinkvar\@firstofthree
    \let\@gls@hyp@opt@cs#1\relax
    \@ifstar{\s@gls@hyp@opt}%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@secondofthree
    \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@thirdofthree
    \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
    \@gls@hyp@opt\@gls@@link
}

```

```

\newcommand*{\@gls@link}[3][\%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\@glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
}

```

```

\glslinkpostsetkeys
\@gls@saveentrycounter
\@gls@setsort{\glslabel}%
\@do@wrglossary{#2}%
\ifKV@glslink@hyper
  \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\else
  \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}%
  {
    \ifcsundef{xatlevel@}%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
    {%
      \ifx\xatlevel@\@empty
        \edef\theglsentrycounter{\expandafter\noexpand
          \csname the\@gls@counter\endcsname}%
      \else
        \savecounters@
        \advance\c@equation by 1\relax
        \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
        \ifcsundef{theH\@gls@counter}%
        {%
          \def\@gls@Hcounter{\theglsentrycounter}%
        }%
        {%
          \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
        }%
        \protected@edef\theHglsentrycounter{\@gls@Hcounter}%
        \restorecounters@
      \fi
    }%
  }%
  {%
    \edef\theglsentrycounter{\expandafter\noexpand
      \csname the\@gls@counter\endcsname}%
  }%
  \ifx\@gls@Hcounter\@empty
    \ifcsundef{theH\@gls@counter}%
    {%
      \def\theHglsentrycounter{\theglsentrycounter}%
    }%
  }%

```

```

    }%
    {%
        \protected@edef\theHglentrycounter{\expandafter\noexpand
            \csname theH\@gls@counter\endcsname}%
    }%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
    \expandafter\@glo@check@mkidrangechar#3\@nil
    \protected@edef#1{%
        \@glo@prefix setentrycounter[#4]{#2}%
        \expandafter\string\csname\@glo@suffix\endcsname
    }%
    \@gls@checkmkidchars#1%
}
\def\@glo@check@mkidrangechar#1#2\@nil{%
\if#1\relax
    \def\@glo@prefix{()}%
    \if\relax#2\relax
        \def\@glo@suffix{glsnumberformat}%
    \else
        \def\@glo@suffix{#2}%
    \fi
\else
    \if#1\relax
        \def\@glo@prefix{}%
        \if\relax#2\relax
            \def\@glo@suffix{glsnumberformat}%
        \else
            \def\@glo@suffix{#2}%
        \fi
    \else
        \def\@glo@prefix{\def\@glo@suffix{#1#2}}%
    \fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
    \def\@gls@checkedmkidx{}%
    \let\gls@xdystring=#1\relax
    \@onelevel@sanitize\gls@xdystring
    \edef\do@gls@xdycheckbackslash{%
        \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
        \@backslashchar\@backslashchar\noexpand\null}%
    \do@gls@xdycheckbackslash
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
    \@for\@gls@tmp:=\gls@protected@pagefmts\do
    {%
        \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
    }
}

```

```

    \@onelevel@sanitize\@gls@sanitized@tmp
    \edef\gls@dostsubst{%
      \noexpand\DTLsubstituteall\noexpand\gls@xdystring
      {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
    }%
    \gls@dostsubst
  }%
  \let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil\\"\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil\?\?\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  \fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\@nil#2\@nil
    \ifx\@nil#3\@nil
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkquote{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
      \def\@gls@checkquote{\@gls@checkquote#3\null}%
    \fi
  \fi
}

```



```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

\@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}{%
\newcommand*{\gls@tmpb}{\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\ifx\noexpand\null##2\noexpand\null
\ifx\noexpand\null##3\noexpand\null
\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\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}%
\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]{%
\gls@showtarget{#1}%
\settoheight\gls@tmplen{#2}%
\raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*{\glsdohyperlink}[2]{%
\gls@showtarget{#1}%
\hyperlink{#1}{#2}%
}
\newcommand*{\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
\let\glslink\glsdonohyperlink

```

```

}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][{}]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]]}%
}
\def\@Gls@#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@secondofthree
  \let\glscustomtext\@empty
  \def\glsinsert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \ifKV@gl@link@local
    \glslocalunset{#2}%
  \else
    \glsunset{#2}%
  \fi
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gl@hyp@opt\@GLS}
\newcommand*{\@GLS}[2][{}]{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]]}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gl@hyp@opt\@glspl}
\newcommand*{\@glspl}[2][{}]{%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2}[]]}%
}
\def\@glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
  }%
}

```

```

\def\glsinsert{#3}%
\def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2][{}]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else

```



```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][{}]{%
    \glsdoifexists{#2}{%
        \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
        \let\glsifplural\@secondoftwo
        \let\glsescapscase\@firstofthree
        \def\glscustomtext{#3}%
        \def\glsinsert{}%
        \def\glo@text{\csname gls@glstype @entryfmt\endcsname}%
        \@gls@link{#1}{#2}{\@glo@text}%
        \ifKV@glslink@local
            \glslocalunset{#2}%
        \else
            \glsunset{#2}%
        \fi
    }%
    \glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}{%
        {%
            \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
            \@gls@link{#1}{#2}{#3}%
        }%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2][{}]{%
    \new@ifnextchar[{\@glsstext@{#1}{#2}}{\@glsstext@{#1}{#2}[]}}
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][{}]{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\newcommand*{\@Glstext}[2][{}]{%
    \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
\def\@Glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
}

```

```

\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []]}
\def\@glsfirst@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []]}
\def\@Glsfirst@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []]}
\def\@GLSfirst@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []]}
\def\@glsplural@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []]}
\def\@Glsplural@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []]}
\def\@GLSplural@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []]}
\def\@glsfirstplural@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []]}
\def\@Glsfirstplural@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] [] {%

```

```

\new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2] []{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2] []{%
  \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\newcommand*{\@Glsdescplural}[2] []{%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
    \new@ifnextchar [ {\@GLSdescplural@{#1}{#2}} {\@GLSdescplural@{#1}{#2} [] }
}
\def \@GLSdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
    \new@ifnextchar [ {\@glssymbol@{#1}{#2}} {\@glssymbol@{#1}{#2} [] }
}
\def \@glssymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] [] {%
    \new@ifnextchar [ {\@Glsymbol@{#1}{#2}} {\@Glsymbol@{#1}{#2} [] }
}
\def \@Glsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
    \new@ifnextchar [ {\@GLSsymbol@{#1}{#2}} {\@GLSsymbol@{#1}{#2} [] }
}
\def \@GLSsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
    \new@ifnextchar [ {\@glssymbolplural@{#1}{#2}} {\@glssymbolplural@{#1}{#2} [] }
}
\def \@glssymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
    \new@ifnextchar [ {\@Glsymbolplural@{#1}{#2}} {\@Glsymbolplural@{#1}{#2} [] }
}
\def \@Glsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
    \new@ifnextchar [ {\@glsuseri@{#1}{#2}} {\@glsuseri@{#1}{#2} [] }
}
\def \@glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
}

```

```

\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][{}]{%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]]}
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][{}]{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]]}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][{}]{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]]}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2][{}]{%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]]}
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\newcommand*{\@GLSuserii}[2][{}]{%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]]}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2][{}]{%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]]}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2][{}]{%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]]}
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\newcommand*{\@GLSuseriii}[2][{}]{%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]]}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2][{}]{%

```

```

    \new@ifnextchar [ {\@glsuseriv@{#1}{#2}} {\@glsuseriv@{#1}{#2} [] }
\def \@glsuseriv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuseriv{#2}#3}%
}
\newrobustcmd* {\Glsuseriv} {\@gls@hyp@opt \@Glsuseriv}
\newcommand* {\@Glsuseriv}[2] [] {%
    \new@ifnextchar [ {\@Glsuseriv@{#1}{#2}} {\@Glsuseriv@{#1}{#2} [] }
\def \@Glsuseriv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\Glsentryuseriv{#2}#3}%
}
\newrobustcmd* {\GLSuseriv} {\@gls@hyp@opt \@GLSuseriv}
\newcommand* {\@GLSuseriv}[2] [] {%
    \new@ifnextchar [ {\@GLSuseriv@{#1}{#2}} {\@GLSuseriv@{#1}{#2} [] }
\def \@GLSuseriv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd* {\glsuserv} {\@gls@hyp@opt \@glsuserv}
\newcommand* {\@glsuserv}[2] [] {%
    \new@ifnextchar [ {\@glsuserv@{#1}{#2}} {\@glsuserv@{#1}{#2} [] }
\def \@glsuserv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuserv{#2}#3}%
}
\newrobustcmd* {\GLSuserv} {\@gls@hyp@opt \@GLSuserv}
\newcommand* {\@GLSuserv}[2] [] {%
    \new@ifnextchar [ {\@GLSuserv@{#1}{#2}} {\@GLSuserv@{#1}{#2} [] }
\def \@GLSuserv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd* {\glsuservi} {\@gls@hyp@opt \@glsuservi}
\newcommand* {\@glsuservi}[2] [] {%
    \new@ifnextchar [ {\@glsuservi@{#1}{#2}} {\@glsuservi@{#1}{#2} [] }
\def \@glsuservi@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuservi{#2}#3}%
}
\newrobustcmd* {\Glsuservi} {\@gls@hyp@opt \@Glsuservi}
\newcommand* {\@Glsuservi}[2] [] {%
    \new@ifnextchar [ {\@Glsuservi@{#1}{#2}} {\@Glsuservi@{#1}{#2} [] }
\def \@Glsuservi@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\Glsentryuservi{#2}#3}%
}
\newrobustcmd* {\GLSuservi} {\@gls@hyp@opt \@GLSuservi}
\newcommand* {\@GLSuservi}[2] [] {%
    \new@ifnextchar [ {\@GLSuservi@{#1}{#2}} {\@GLSuservi@{#1}{#2} [] }
\def \@GLSuservi@#1#2[#3] {%

```

```

\@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]%
}
\def\@acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapsaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
\def\@ACRshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapsaps\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] []{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}

```



```

\def\@CRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\acrlong{\@gl@hyp@opt\@ns@acrlong}
\newcommand*\ns@acrlong[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\Acrlong{\@gl@hyp@opt\@ns@Acrlong}
\newcommand*\ns@Acrlong[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\newcommand*{\ns@ACRlong}[2] [] {%
  \new@ifnextchar [{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2} []}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2] [] {%
  \new@ifnextchar [{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2} []}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@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\gls caps case\@secondofthree
\let\glsinsert\@empty
\def\gls custom text{%
  \Glsentrylongpl{#2}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\gls post link hook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
  }%
  ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
  entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
  label and the field name}%
}%
}%

```

```

    ???
  }%
}
\newcommand*\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  \else
    \expandafter\ifx\@gls@body\glsentryshort\relax
    \expandafter\Glsentryshort\@gls@rest
  \else
    \expandafter\ifx\@gls@body\acronymfont\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \xmakefirstuc{\@glo@text}%
  \fi
  \fi
  \fi
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newrobustcmd*\Glsentrydesc}[1]{%
  \@Gls@entry@field{#1}{desc}%
}
\newcommand*\glsentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
}

```

```

\newrobustcmd*{\Glsentrydescplural}[1]{%
  \@Gls@entry@field{#1}{descplural}%
}
\newcommand*{\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newrobustcmd*{\Glsentrytext}[1]{%
  \@Gls@entry@field{#1}{text}%
}
\newcommand*{\glsentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
\newrobustcmd*{\Glsentryplural}[1]{%
  \@Gls@entry@field{#1}{plural}%
}
\newcommand*{\glsentrysymbol}[1]{%
  \@gls@entry@field{#1}{symbol}%
}
\newrobustcmd*{\Glsentrysymbol}[1]{%
  \@Gls@entry@field{#1}{symbol}%
}
\newcommand*{\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*{\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*{\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*{\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \xcapitalisewords{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsentrytitlecase}[2]{%
    \texorpdfstring
      {\@glsentrytitlecase{#1}{#2}}%
      {\@gls@entry@field{#1}{#2}}%
}

```

```

}
}
{
  \newcommand*\glentrytitlecase}[2]{\@glentrytitlecase{#1}{#2}}
}
\newcommand*\glentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newrobustcmd*\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*\glentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*\glentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*\glentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*\glentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*\glentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
}
\newcommand*\glentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*\Glsentryshort}[1]{%

```

```

    \@Gls@entry@field{#1}{short}%
  }
  \newcommand*\glentryshortpl}[1]{\@Gls@entry@field{#1}{shortpl}}
  \newrobustcmd*\Glsentryshortpl}[1]{%
    \@Gls@entry@field{#1}{shortpl}%
  }
  \newcommand*\glentrylong}[1]{\@Gls@entry@field{#1}{long}}
  \newrobustcmd*\Glsentrylong}[1]{%
    \@Gls@entry@field{#1}{long}%
  }
  \newcommand*\glentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}
  \newrobustcmd*\Glsentrylongpl}[1]{%
    \@Gls@entry@field{#1}{longpl}%
  }
  \newcommand*\glentryfull}[1]{%
    \acrfullformat{\glentrylong{#1}}{\acronymfont{\glentryshort{#1}}}%
  }
  \newrobustcmd*\Glsentryfull}[1]{%
    \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glentryshort{#1}}}%
  }
  \newcommand*\glentryfullpl}[1]{%
    \acrfullformat{\glentrylongpl{#1}}{\acronymfont{\glentryshortpl{#1}}}%
  }
  \newrobustcmd*\Glsentryfullpl}[1]{%
    \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glentryshortpl{#1}}}%
  }
  \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
          \edef\@glo@label{\glsdetoklabel{#1}}%
          \let\@org@glsnumberformat\glsnumberformat

```

```

\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
  \csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*\glsnumlistsep{, }
\newcommand*\glsnumlistlastsep{ \& }
\newcommand*\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd}[2][{}]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{\glsnumberformat}%
    \edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}

```



```

\newrobustcmd*\glsaddall}[1] [] {%
  \edef\glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{ }
\edef\glsclosebrace{\expandafter\@gobble\string\} }
\edef\glsbackslash{\expandafter\@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter\@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter}{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsxindy
  \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsxindy
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
  }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

        \renewcommand*{\@xdynumbergrouporder}{#1}%
        \@onelevel@sanitize\@xdynumbergrouporder
    }
\else
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
        \glsnoxywarning\GlsSetXdyNumberGroupOrder}
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
        \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsxindy
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \renewcommand*{\@glsminrange}{#1}}
\else
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
    \def\writeist{%
        \ifundef{glswrite}{\newwrite\glswrite}{}%
        \@gls@addpredefinedattributes
        \openout\glswrite=\istfilename
        \write\glswrite{;; xindy style file created by the glossaries
            package}%
        \write\glswrite{;; for document '\jobname' on
            \the\year-\the\month-\the\day}%
        \write\glswrite{^^J; required styles^^J}
        \@for\@xdystyle:=\@xdyrequiredstyles\do{%
            \ifx\@xdystyle\@empty
            \else
                \protected@write\glswrite}{(require
                    \string"\@xdystyle.xdy\string")}%
            \fi
        }%
        \write\glswrite{^^J%
            ; list of allowed attributes (number formats)^^J}%
        \write\glswrite{(define-attributes ((\@xdyattributes)))}%
        \write\glswrite{^^J; user defined alphabets^^J}%
        \write\glswrite{\@xdyuseralphabets}%
        \write\glswrite{^^J; location class definitions^^J}%
        \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
            \protected@write\glswrite}{(define-location-class
                \string"\@gls@classI\string"^^J\space\space\space
                (
                    :sep "{}{"
                    \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
                    :sep "}")
                )
            ^^J\space\space\space

```

```

:min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}\{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}\{"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{}\{\string\providecommand*%
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
        {%
          \string\setentrycounter
            [\expandafter\@gobble\string\#1]{\@this@ctr}%
          \expandafter\string

```

```

\csname\@this@attr\endcsname
  {\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
  \string\begin
    {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
    \space\space:close \string"\glpercentchar\glstildechar n\string
    \end{theglossary}\string\glossarypostamble
    \glstildechar n\string" ^^J\space\space\space
    :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string\relax \string"}
\glossaryentrynumbers\string\{\string\relax \string"}

```

```

\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string"\string}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\@ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%

```

```

\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesobar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkesquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkesquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkesquote{%
\noexpand\@gls@checkesquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkesquote{%
\noexpand\@gls@checkesquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%

```



```

        \noexpand\else
        \noexpand\def\noexpand\@gls@checkescquote{%
            \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
            ####3\noexpand\null}%
        \noexpand\fi
        \noexpand\fi
        \noexpand\@gls@checkescquote
    }%
}
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
    {\string\GlsSetQuote\space not permitted here}%
    {Move \string\GlsSetQuote\space earlier in the preamble, as
    soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand*\@noist}{%
    \@gls@addpredefinedattributes
    \let\writeist\relax
}
\newcommand*\@makeglossary}[1]{%
    \ifglossaryexists{#1}%
    {%
        \ifglssavewrites
            \expandafter\newtoks\csname glo@#1@filetok\endcsname
        \else
            \expandafter\newwrite\csname glo@#1@file\endcsname
            \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
        \fi
        \@gls@renewglossary
        \writeist
    }%
    {%
        \PackageError{glossaries}%
        {Glossary type ‘#1’ not defined}%
        {New glossaries must be defined before using \string\makeglossaries}%
    }%
}
\newcommand*\@glsopenfile}[2]{%
    \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
    \PackageInfo{glossaries}{Writing glossary file
        \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*\@closegls}[1]{%
    \closeout\csname glo@#1@file\endcsname
}
\ifglsexindy
\newcommand*\@gls@automake}[1]{%
    \ifglossaryexists{#1}

```

```

{%
  \@closegls{#1}%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-M ord/letorder }}%
  {\let\@gls@order\@empty}%
  \ifcsundef{@xdy@#1@language}%
  {\let\@gls@langmod\@xdy@main@language}%
  {\letcs\@gls@langmod{@xdy@#1@language}}%
  \edef\@gls@dothiswrite{\noexpand\write18{xindy
    -I xindy
    \@gls@order
    -L \@gls@langmod\space
    -M \@gls@istfilebase\space
    -C \@gls@codepage\space
    -t \jobname.\csuse{@glotype@#1@log}
    -o \jobname.\csuse{@glotype@#1@in}
    \jobname.\csuse{@glotype@#1@out}}}%
  }%
  \@gls@dothiswrite
}%
{
  \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-l }}%
    {\let\@gls@order\@empty}%
    \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
      -s \istfilename\space
      -t \jobname.\csuse{@glotype@#1@log}
      -o \jobname.\csuse{@glotype@#1@in}
      \jobname.\csuse{@glotype@#1@out}}}%
    }%
    \@gls@dothiswrite
  }%
  \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%

```

```

{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilename\space
-C \gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
}

```

```

\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
\ifundef\@@gls@extramakeindexopts
{}%
{%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{} }
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{} }{}{}%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4] []{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\def\warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
}%

```

```

\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
\@domakeglossaries
{%
\renewcommand{\@gls@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*\@gls@writefiles{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{%
    }%
    }%
    {\def\gls@tmp{\expandafter\the
      \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    }%
  }%
}

```



```

        \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
            \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*\@gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand*\@gls@renewglossary{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*\@gls@wrglossary}[2]{%
    \ifgls@savewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp~J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
            }%
            {%
                \ifignoredglossary{#1}{}%
                    {%
                        \GlossariesWarning{No file defined for glossary ‘#1’}%
                    }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*\@gls@writeentry}[2]{%
    \ifgls@indexonlyfirst
        \ifglsused{#1}{#2}%
    \else
        #2%
    \fi
}

```

```

\fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
\@for\@gls@this:=\gls@protected@pagefmts\do
{%
\expandafter\let\@gls@this\relax
}%
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\glsaddprotectedpagefmt}[1]{%
\ea\pt\o\gls@protected@pagefmts{\,\expandonce{\csname gls#1page\endcsname}}%
\csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
\ea\pt\o\@wrglossarynumberhook{%
\noexpand\let\expandonce{\csname org@gls#1\endcsname}%
\expandonce{\csname#1\endcsname}%
\noexpand\def\expandonce{\csname#1\endcsname}{%
\noexpand\@wrglossary@pageformat
\expandonce{\csname gls#1page\endcsname}%
\expandonce{\csname org@gls#1\endcsname}%
}%
}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
\ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@do@wrglossary}[1]{%
\ifglseclocations
\@do@esc@wrglossary{#1}%
\else
\@do@noesc@wrglossary{#1}%
\fi
}
\newcommand*{\@do@noesc@wrglossary}[1]{%
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
\ifx\@glsHlocref\@glslocref
\def\@glo@counterprefix{}%
\else
\protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix

```

```

\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\thegl@entrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHgl@entrycounter\thegl@entrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHgl@entrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary

```

```

}
\newcommand*{\@do@wrglossary}{%
  \ifglxindy
    \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
    (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
      :locref \string{\@glo@counterprefix}{\@glslocref}\string" %
      :attr \string"\@gls@counter\@glo@suffix\string"
      \@glo@range
    )
  }%
\else
  \set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
  {\@glo@counterprefix}%
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
    \string@glossaryentry{\csname glo@\@gls@label @index\endcsname
      \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
  \fi
}
\newcommand*\@gls@getcounterprefix[2]{%
  \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
  \ifx\@glo@counterprefix\@empty
    \GlossariesWarning{Hyper target '#2' can't be formed by
      prefixing location '#1'. You need to modify the
      definition of \string\theH\@gls@counter otherwise you
      will get the warning: "'name{\@gls@counter.#1}' has been
      referenced but does not exist"}%
  \fi
\fi
}

```

```

\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :key (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
    \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{#{#2#3}}%
  \fi
}
\def\@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*{\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
}
\newcommand*{\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*{\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseeitem{#2}}
\newrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }

```

```

\newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*{\gls@save@numberlist}[1]{%
  \ifgls@save@numberlist
    \toks@{#1}%
    \edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@\gls@current@entry@label @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}{\@do@writeaux@info}%
  \fi
}
\newcommand*{\warn@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{
  \@gls@warn@gloss@defined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forall@glossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forall@glossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
}
\newcommand*{\@gls@preglossaryhook}{%
}
\newcommand{\@printglossary}[2]{%

```

```

\def@glo@type{glsdefaulttype}%
\def@glossarytitle{\csname @glo@type @title\endcsname}%
\def@glossarytoctitle{glossarytitle}%
\let\org@glossarytitle@glossarytitle
\def@glossarystyle{%
  \ifx@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def@gls@dotoc@title{glssettoc@title{glo@type}}%
\let@org@glossaryentrynumbers@glossaryentrynumbers
\begin{group}
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{glo@type}%
  {%
    \ifx@glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary@glo@type
    \let\org@glossaryentrynumbers@glossaryentrynumbers
    \let\glsnonextpages@glsnonextpages
    \let\glsnextpages@glsnextpages
    \let\nopostdesc@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield@glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
  }%
\end{group}
\global\let@glossaryentrynumbers@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}

```

```

\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglxindy
    \ifcsundef{@xdy@\glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
            \noexpand\immediate\noexpand\write\@auxout{%
              \string\@xdylanguage{\glo@type}{\@xdy@main@language}%
            }%
          }%
        }%
      }%
    }%
    {\edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\glo@type}{\csname @xdy@\glo@type
              @language\endcsname}%
          }%
        }%
      }%
    }%
  \do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@gls@codepage{\glo@type}{\gls@codepage}%
        }%
      }%
    }%
  \do@auxoutstuff
}
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
  hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \edef\@glo@type{#2}%

```



```

\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%
  }%
  }%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    }%
  }%
  \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    }%
  \expandafter\@glo@sortedinsert
  \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
  \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
  {%
    }%
  }%
  \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
}

```

```

    }%
  }%
  {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
  }%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@word}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=word]}{}}%
}%
}
\newcommand*\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=letter]}{}}%
  }%
}
\newcommand*\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@\glsorder}%
    {%
      \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=standard]}{}}%
  }%
}
\newcommand*\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=case]}{}}%
  }%
}
\newcommand*\@glo@sortmacro@nocase}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{%
  \forlssentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listeadadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
  }%
  \glossarysection[\@glossarytoctitle]{\@glossarytitle}%
  \glossarypreamble
  \def\@gls@currentlettergroup{}%
  \begin{theglossary}%

```

```

\glossaryheader
\glsresetentrylist
\forlistcsloop{\@gls@noidx@do}{@glsref@{@glo@type}}%
\end{theglossary}%
\glossarypostamble
}%
{%
\@gls@noref@warn{@glo@type}}%
}%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\ucode‘#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@=‘#1\relax
}%
{%
}%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{glo@glstdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{glo@glstdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
}
}

```

```

    {%
      \subglossentry{\gls@level}{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
  {%
    \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
    \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
    {%
      \ifdefempty{\@gls@currentlettergroup}{}%
      {%
        \global\let\@glo@thislettergrp\@glo@thislettergrp
        \glsgroupskip
      }%
      \glsgroupheading{\@glo@thislettergrp}%
    }%
    \global\let\@gls@currentlettergroup\@glo@thislettergrp
    \ifdefvoid{\@gls@loclist}
    {%
      \glossentry{#1}{}%
    }%
    {%
      \glossentry{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
}
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%

```

```

\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
  \@gls@noidxloclist@sep
  \@gls@noidxloclist@prev
  \def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
    \ifinlistcs{#2}{\@glsref@#1}%
    {}%
    {\listcsgadd{\@glsref@#1}{#2}}%
    \ifcsundef{glo@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@dotocitle\relax
}
\define@key{printgloss}{style}{%
  \ifcsundef{\@glsstyle@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \def\@glossarystyle{\setglossentrycompatibility
      \csname @glsstyle@#1\endcsname}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{}%

```

```

\ifcase\gls@numberedsection@nr\relax
  \renewcommand*\@@glossarysecstar}{*}%
  \renewcommand*\@@glossaryseclabel}{}%
\or
  \renewcommand*\@@glossarysecstar}{}%
  \renewcommand*\@@glossaryseclabel}{}%
\or
  \renewcommand*\@@glossarysecstar}{}%
  \renewcommand*\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
  \renewcommand*\@@glossarysecstar}{*}%
  \renewcommand*\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
\newcommand*\GlsEntryCounterLabelPrefix{\glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glsesubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%

```



```

}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%

```

```

\ifglssubentrycounter
  \theglossarysubentry)\space
\fi
}
\newcommand*\glssentryitem}[1]{%
  \ifglssentrycounter
    \glssstepentry{#1}\glssentrycounterlabel
  \else
    \glssresetsubentrycounter
  \fi
}
\newcommand*\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glssstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
  {%
    \newenvironment{theglossary}{}{}%
  }%
  {%
    \@gls@warnontheglossdefined
    \renewenvironment{theglossary}{}{}%
  }
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\providecommand*\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@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}{glsymbols}}
 or test{\ifstrequal{#1}{glsnumbers}}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
 {%
 \def#2{#1}%
 }%
 }%
 }%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
 \DTLifint{#1}%
 {\edef#2{\char#1\relax}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
}
\newcommand*{\glsgetgrouplabel}[1]{%
 \ifthenelse{\equal{#1}{\glsymbolsgroupname}}{\glsymbols}{%
 \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
}
\newcommand*{\setentrycounter}[2][1]{%
 \def\@glo@counterprefix{#1}%
 \ifx\@glo@counterprefix\@empty
 \def\@glo@counterprefix{.}%
 \else
 \def\@glo@counterprefix{.#1}%
 \fi
 \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
 \ifcsundef{@glsstyle@#1}%
 {%
 \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
}

```

```

}%
{%
  \csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
     I recommend you switch to \string\setglossarystyle\space unless
     you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%
}

```

```

}%
{%
  \def\glsnumber#1{\@glsnumber#1\nohyperpage{}}\@nil}
}
\def\@glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glsnumber#3\@nil
  \fi
}
\def\@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
  \@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\x{\the\toks@}%
  \ifx\x\empty
  \else
    \hyperlink{\glsentrycounter\@gls@counterprefix\the\toks@}%
      {\the\toks@}%
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
}

```

```

\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
  }%
}
\newcommand{\newacronym}[4][{}]{
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}
\newrobustcmd*\glsstextup[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey{short}
\newcommand*\glsshortpluralkey{shortplural}
\newcommand*\glslongkey{long}
\newcommand*\glslongpluralkey{longplural}
\newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
  {\@acrfull{#1}{#2} []}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4} []}%
}

```

```

\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
\newrobustcmd{\glsspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
    {\@Acrfull{#1}{#2} []}%
  }
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
    {\@ACRfull{#1}{#2} []}%
  }
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
    {\@acrfullpl{#1}{#2} []}%
  }
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
    {\@Acrfullpl{#1}{#2} []}%
  }
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
    {\@ACRfullpl{#1}{#2} []}%
  }
}

```



```

}
\def\ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand*\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\ACRlongpl}{\ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acrenryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \gslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink{##1}{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullfmt}[3]{%

```

```

\glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}%
\renewcommand*\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}%
\renewcommand*\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields>{description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*\acronymsort}[2]{#1}
\newcommand*\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{%
}%
}%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
}%
}%
}
\newcommand*\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{%
}%
}%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%

```

```

\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}

```

```

}%
\renewcommand*\genplacrfullformat}[2]{%
  \glentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glentryshort{##1}}##2\space
    (\glentrylong{##1})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glentryshort{##1}}##2\space
    (\glentrylong{##1})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glentryshortpl{##1}}##2\space
    (\glentrylongpl{##1})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glentryshortpl{##1}}##2\space
    (\glentrylongpl{##1})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%

```

```

{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%

```

```

{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylong{##2}##3\space}
  (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space}
  (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylong{##2}##3\space}
    (\acronymfont{\glsentryshort{##2}})}%
  }%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space}
  (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space}
  (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space}
    (\acronymfont{\glsentryshortpl{##2}})}%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%

```



```

{%
\ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
\acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
\acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%

```

```

\renewcommand*\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*\DefineAcronymSynonyms{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acp\Acrshortpl
  \let\acl\aclong
  \let\Acl\Aclong
  \let\aclp\aclongpl
  \let\Aclp\Aclongpl
  \let\acf\acrfull
  \let\Acf\Acrfull
  \let\acfp\acrfullpl
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\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{\@gls@acrpluralsuffix}%
}
\newcommand*\@acrfootnote}[3]{\@acrlinkfootnote{#1}{#2}{#3}}
\newcommand*\@acrlinkfootnote}[3]{%
    \footnote{\@glslink[#1]{#2}{#3}}%
}
\newcommand*\@acrnohreffootnote}[3]{%
    \footnote{#3}%
}
\newcommand*\@SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
    \def\glsentryfmt[#1]{%
        \ifdefempty\gls@customtext

```

```

{%
  \ifglused{\glslabel}%
  {%
    \acronymfont{\glsgenentryfmt}%
  }%
  {%
    \firstacronymfont{\glsgenentryfmt}%
    \ifglshassymbol{\glslabel}%
    {%
      \expandafter\protect\expandafter\acrfootnote\expandafter
      {\@gls@link@opts}{\@gls@link@label}%
      {%
        \glsifplural
          {\glsentrysymbolplural{\glslabel}}%
          {\glsentrysymbol{\glslabel}}%
        }%
      }%
    }%
  }%
  }%
  {\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    symbol={\the\glslongtok},%
    symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}

```

```

}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\SetDescriptionFootnoteAcronymStyle{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*\acronymfont[1]{\textsc{##1}}%
    \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsacrdua
    \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
      can’t both be set}{}%
  \fi
}%
\newcommand*\SetDescriptionDUAAcronymDisplayStyle[1]{%
  \defglsentryfmt[##1]{\glsgenentryfmt}%
}
\newcommand*\DescriptionDUANewAcronymDef{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},

```

```

text={\the\glslongtok},%
first={\the\glslongtok},%
plural={\noexpand\expandonce\noexpand\@glo@longpl},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \ifglsacrsmaller
      \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
        can’t both be set}{}%
    \fi
  \fi
\renewcommand{\newacronym}[4][[]]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%

```

```

\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*\SetDescriptionAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsentryfmt}\gls@org@insert
}%
{%
\glsentryfmt
\ifgls hassymbol{\glslabel}%
{%
\glsifplural
{%
\def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
}%
{%
\def\@glo@symbol{\glsentrysymbol{\glslabel}}%
}%
\space(\protect\firstacronymfont
{\gls caps case
{\@glo@symbol}
{\@glo@symbol}
{\mfirstucMakeUppercase{\@glo@symbol}}})%
}%
{}}%
}%
{\gls custom text\glsinsert}%
}%
}
\newcommand*\DescriptionNewAcronymDef){%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronym type,%
name={\noexpand
\acrnameformat{\the\gls short tok}{\the\gls long tok}},%
sort={\the\gls short tok},%
first={\the\gls long tok},%

```



```

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

```

```

\ifglsacrsmaller
  \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert@empty
\ifglsused{\glslabel}%
{%
\acronymfont{\glsentryfmt}\gls@org@insert
}%
{%
\firstacronymfont{\glsentryfmt}\gls@org@insert
\ifglsashaslong{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
}%
\glsifplural
{\glsentrylongpl{\glslabel}}%
{\glsentrylong{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*\@SetFootnoteAcronymStyle{%
  \renewcommand*\@newacronym}[4][ ]{%
    \ifx\@gls@acronymlists\@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:=\@gls@acronymlists\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\gls@insert
      \let\gls@insert\@empty
      \ifglsused{\gls@label}%
      {%
        \acronymfont{\gls@genentryfmt}\gls@org@insert
      }%
      {%
        \gls@genentryfmt
        \ifgls@shassymbol{\gls@label}%
        {%
          \gls@ifplural
          {%
            \def\@glo@symbol{\gls@entrysymbolplural{\gls@label}}%
          }%
          {%
            \def\@glo@symbol{\gls@entrysymbol{\gls@label}}%
          }%
          \space
          (\gls@scapscase
           {\firstacronymfont{\@glo@symbol}}%
           {\firstacronymfont{\@glo@symbol}}%
           {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
      }%
    }%
    {\gls@customtext\gls@insert}%
  }%
}
\newcommand*\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{the\gls@labeltok}%
    {%
      type=\acronymtype,%
    }
  }

```

```

name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
}

```

```

\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand*\acronymfont[1]{\textsc{##1}}
\renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
\renewcommand*\acronymfont[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
\ifglsacrsmallcaps
\PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
\PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
}%
\newcommand*\SetDUADisplayStyle[1]{%
\defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\the\glsshorttok},%
text={\the\glslongtok},%
first={\the\glslongtok},%
plural={\noexpand\expandonce\noexpand\@glo@longpl},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural

```

```

\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\let\gl@assign@descplural\org@gl@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@sacronymlists\@empty
      \def\@gl@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@gl@type}%
      \SetDUADisplayStyle{\@gl@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@sacronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle
  \ifgl@acrdescription
    \ifgl@sacrfootnote
      \SetDescriptionFootnoteAcronymStyle
    \else
      \ifgl@sacrdua
        \SetDescriptionDUAAcronymStyle
      \else
        \SetDescriptionAcronymStyle
      \fi
    \fi
  \fi
}

```

```

\else
  \ifglacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glacrsmallcaps}\OR
      \boolean{glacrsmaller}}{%
      {%
        \SetSmallAcronymStyle
      }%
    }%
    \ifglacrdua
      \SetDUASStyle
    \fi
  }%
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      user1={\the\glsshorttok},%
      user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
      user3={\the\glslongtok},%
      user4={\the\glslongtok\noexpand\acrpluralsuffix},%
      \CustomAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}

```



```

\newcommand*\SetCustomStyle{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetCustomDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \CustomNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetCustomDisplayStyle{\@gls@type}%
  }%
}
\RequirePackage{glossary-hypernav}[=v4.46]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*\showglocounter}[1]{%

```

```

    \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
  }
  \newcommand*{\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@glstdetoklabel{#1}@loclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \csshow{glo@glstdetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show@glssacronymlists
}
\newcommand*\showglossaries}{%
  \show@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glo@list@#1\endcsname
}
\csname ifglsc-compatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.46]
\fi

```

8.2 Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2020/03/19 v4.46 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}
{%
  \ifx@glsxtr@doaccsupp\empty
  \GlossariesWarning{The ‘glossaries-accsupp’
    package has been loaded\MessageBreak
    after the ‘glossaries-extra’ package. This\MessageBreak
    can cause a failure to integrate both packages. \MessageBreak
    Either use the ‘accsupp’ option when you load\MessageBreak
    ‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
    before loading ‘glossaries-extra’}%
}

```

```

\fi
}
{}
\def\compatibleglossentry#1#2{%
\ toks@{#2}%
\protected@edef\do@glossentry{%
\ noexpand\accsuppglossaryentryfield{#1}%
\ noexpand\glsnamefont
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
\ the\toks@}%
}%
\do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\ toks@{#3}%
\protected@edef\do@subglossentry{%
\ noexpand\accsuppglossarysubentryfield{\number#1}%
\ #2}%
\ noexpand\glsnamefont
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
\ the\toks@}%
}%
\do@subglossentry
}
\RequirePackage{glossaries}[v4.46]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}
}
}
{}
\newcommand{\glsaccessibility}[4][[]]{%
\@glsshowaccsupp{#1}{#2}{#3}%
\gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
\def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
\def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%

```

```

\def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
\def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
\def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
\def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
\def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
\def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
\def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
\def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
\def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
\def\@glo@useriiiaccess{#1}%
}
\define@key{glossentry}{user4access}{%
\def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
\def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
\def\@glo@userviaccess{#1}%
}
}

```

```

\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
  \def\@glo@userivaccess{\relax}%
  \def\@glo@uservaccess{\relax}%
  \def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%

```

```

        \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
    \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
    \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
    \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter

```

```

\protected@xdef\csname glo@\glo@label @servaccess\endcsname{%
  \@glo@servaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @serviaccess\endcsname{%
  \@glo@serviaccess}%
}
\newcommand*\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
\newcommand*\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}

```



```

}
\newcommand*{\glentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
  #1%
  \else

```

```

        \xglsaccsupp{\@glo@access}{#1}%
    \fi
}
\newcommand*\@gls@fieldaccess@display[4]{%
    \protected@edef\@glo@access{#4}%
    \ifdefequal\@glo@access\@gls@noaccess
        {#3}%
        {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
    }
\newrobustcmd*\@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}{\glstentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glstentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glstentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glstentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glstentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glstentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriii}{#1}{\glstentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@userivaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriv}{#1}{\glstentryuserivaccess{#2}}}%
}
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@uservaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userv}{#1}{\glstentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
\ifcsundef{gls#1accessdisplay}%
{%
\PackageError{glossaries-accsupp}{No accessibility support
for key ‘#1’}{%
}%
}%
\csname gls#1accessdisplay\endcsname{#2}{#3}%
}%
}
\renewcommand*{\@gls@default@entryfmt}[2]{%
\ifdefempty\glscustomtext
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}
{\glsinsert}}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\glsentryfirstplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
}%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\Glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
}
}
}

```

```

        {\glsinsert}%
    }%
    {%
        #1{\glsfirstpluralaccessdisplay
            {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
                {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\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}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
#2{\glstextaccessdisplay
  {\Glsentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\Glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glstextaccessdisplay
  {\glsentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
\MakeUppercase{%
#1{\glsfirstaccessdisplay
  {\glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%

```

```

        {\glsinsert}}%
    }%
} %
} %
} %
{ %
\ifglsused{\glslabel}%
{ %
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
} %
{ %
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
} %
} %
}
\renewcommand*{\glsgenentryfmt}{%
\ifdefempty\glscustomtext
{ %
    \glsifplural
    { %
        \glscapscase
        { %
            \ifglsused\glslabel
            { %
                \glspluralaccessdisplay
                {\glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
            } %
            { %
                \glsfirstpluralaccessdisplay
                {\glsentryfirstplural{\glslabel}}{\glslabel}%
                \glsinsert
            } %
        } %
    } %
    { %
        \ifglsused\glslabel
        { %
            \glspluralaccessdisplay
            {\Glsentryplural{\glslabel}}{\glslabel}%
            \glsinsert
        } %
    } %
}

```

```

}%
{%
  \glsfirstpluralaccessdisplay
    {\Glsentryfirstplural{\glslabel}}{\glslabel}%
  \glsinsert
}%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%

```



```

        \glstextaccessdisplay
          {\mfirstucMakeUppercase{\glentrytext{\glslabel}}}{\glslabel}%
        \mfirstucMakeUppercase{\glsinsert}%
      }%
    {%
      \glsfirstaccessdisplay
        {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}
}
}
{%
  \glscustomtext\glsinsert
}
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\glentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
      }%
      \acronymfont
      {\glsshortpluralaccessdisplay
        {\Glsentryshortpl{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
    {%
      \mfirstucMakeUppercase
      {\acronymfont
        {\glsshortpluralaccessdisplay
          {\glentryshortpl{\glslabel}}{\glslabel}}%
        \glsinsert}%
    }%
  }%
}
}
}
{%
  \glscapscase
  {%
    \acronymfont
    {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
  \glsinsert
}

```



```

(\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{#1}}}{#1})%
}
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glentryshort{#2}}{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glsshortaccessdisplay
    {\MakeUppercase{\glsentryshort{#2}}}{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@AcrLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \MakeUppercase{\glsentrylong{#2}}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

\glspostlinkhook
}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glstextentrytext{#2}}{#2}#3}%
}
\def\@GLstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\GLstextentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glstextaccessdisplay{\mfirstucMakeUppercase{\glstextentrytext{#2}}}{#2}%
  \mfirstucMakeUppercase{#3}}%
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glstextentryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\GLstextentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glstextentryfirst{#2}}}{#2}%
  \mfirstucMakeUppercase{#3}}%
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glstextentryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\GLstextentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glstextentryplural{#2}}}{#2}%
  \mfirstucMakeUppercase{#3}}%
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glstextentryfirstplural{#2}}{#2}#3}%
}
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\GLstextentryfirstplural{#2}}{#2}#3}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glstextentryfirstplural{#2}}}{#2}%
  \mfirstucMakeUppercase{#3}}%
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glstextentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glssymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glssymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}

```

```

}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

```

```

}
\def\@GLSuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseraccessdisplay{\mfirstucMakeUppercase{\glsentryuser{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
}
\renewcommand*\@glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*\@glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*\@glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*\@Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*\@glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*\@Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%

```



```

        \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
    }%
}
\newcommand*\accsuppglossaryentryfield}[5]{%
    \glossaryentryfield{#1}%
    {\glsnameaccessdisplay{#2}{#1}}%
    {\glsdescriptionaccessdisplay{#3}{#1}}%
    {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*\accsuppglossarysubentryfield}[6]{%
    \glossarysubentryfield{#1}{#2}%
    {\glsnameaccessdisplay{#3}{#2}}%
    {\glsdescriptionaccessdisplay{#4}{#2}}%
    {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
    \renewcommand*\genacrfullformat}[2]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
    }%
    \renewcommand*\Genacrfullformat}[2]{%
        \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
    }%
    \renewcommand*\genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*\acronymentry}[1]{%
        \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}
    }
    \renewcommand*\acronymsort}[2]{##1}%
    \renewcommand*\acronymfont}[1]{##1}%
    \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
    \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%

```

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            \mfirstucMakeUppercase{\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
          \mfirstucMakeUppercase{\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
}

```

```

\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
}

```

```

(\glslongaccessdisplay{\glsentrylong{##2}{##2}})}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
    (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
        (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glsentrylongpl{##2}{##2}})}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glsentrylongpl{##2}})}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}{##2}})}%
\renewcommand*\glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}{##1}})}%
\renewcommand*\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}{##1}})}%
\renewcommand*\glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}{##1}})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}{##1}})}%
}
\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}

```

```

\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
}
}

```



```

\newcommand*\glsdefaultshortaccess}[2]{#1}
\renewcommand*\newacronymhook{%
  \edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*\DefaultNewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      description={\the\glslongtok},%
      descriptionaccess=\relax,
      text={\the\glsshorttok},%
      access={\noexpand\@glo@textaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      first={\noexpand\glslongaccessdisplay
        {\the\glslongtok}{\the\glslabeltok}\space
        (\noexpand\glsshortaccessdisplay
          {\the\glsshorttok}{\the\glslabeltok})},%
      plural={\the\glsshorttok\acrpluralsuffix},%
      firstplural={\noexpand\glslongpluralaccessdisplay
        {\noexpand\@glo@longpl}{\the\glslabeltok}\space
        (\noexpand\glsshortpluralaccessdisplay
          {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
      firstaccess=\relax,
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%

```

```

    \@gls@expand@field{##1}{descplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      symbol={\the\glslongtok},%
      symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@symbolplural##1##2{%
    \@gls@expand@field{##1}{symbolplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%

```

```

type=\acronymtype,%
name={\noexpand
  \acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
access={\noexpand\@glo@textaccess},%

```

```

plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\org@gls@assign@plural\gls@assign@plural
\let\org@gls@assign@firstpl\gls@assign@firstpl
\let\org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\org@gls@assign@plural
\let\gls@assign@firstpl\org@gls@assign@firstpl
\let\gls@assign@descplural\org@gls@assign@descplural
}
\renewcommand*\SmallNewAcronymDef{%
\edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    access={\noexpand\@glo@symbolaccess},%
    sort={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    text={\noexpand\@glo@short},%
    textaccess={\noexpand\@glo@shortaccess},%
    plural={\noexpand\@glo@shortpl},%
    first={\the\glslongtok},%
    firstaccess=\relax,
    firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\noexpand\@glo@first},%
    descriptionplural={\noexpand\@glo@firstplural},%
    symbol={\the\glsshorttok},%
    symbolaccess={\the\glslongtok},%
  }
}

```

```

        symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglongnameaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglongtextaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglongpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglongfirstaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglongfirstpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglongsymbolaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglongsymbolpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
}

```

```

}
\newcommand*\showglodescaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*\showglodescpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*\showgloshortaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*\showgloshortpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*\showglolongaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*\showglolongpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}
}

```

8.3 Rollback v4.46 (glossaries-babel-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
}
}
}

```

8.4 Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2020/03/19 v4.46 (NLCT)]
\ifglxindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\xdyattributes{\xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J%
    (markup-locref
    :open \string"\string~\n\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter@gobble\string\{\string" ^^J
    :close \string"\expandafter@gobble\string\}\string" ^^J
    :attr \string"#1\string")}}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxindy
  \renewcommand*\GlsAddXdyLocation[2]{%
    \edef\xdyuserlocationdefs{%
      \xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
  }

```

```

    }%
    \edef\xdyuserlocationnames{%
        \xxyuserlocationnames^^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 :key (\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=\listfilename
    \write\glswrite{;; xindy style file created by the glossaries
        package in compatible-2.07 mode}%
    \write\glswrite{;; for document '\jobname' on
        \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\xdystyle:=\xdyrequiredstyles\do{%
        \if\xdystyle\@empty

```



```

\else
  \protected@write\glswrite{}{(require
    \string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
  ; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes (\@xdyattributes))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-lowercase\string")
    :min-range-length \@glsminrange)}%
\else
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (:sep "\@gls@roman")
    :min-range-length \@glsminrange)}%
\fi
\write\glswrite{(define-location-class
  \string"Roman-page-numbers\string"^^J\space\space\space
  (\string"roman-numbers-uppercase\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-page-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"alpha-page-numbers\string"^^J\space\space\space
  (\string"alpha\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
   :sep \string"\glscompositor\string"
   \string"arabic-numbers\string")
   :min-range-length \@glsminrange)}}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter@gobble
  \string~%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi

```

```

\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string\string\glsgroupskip\string~n\string)}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string\string\glsgroupheading
\glsopenbrace\string^^J\space\space\space
:close-head \string\glsclosebrace\string)}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\gls@actualchar{\string?}
\edef\gls@encapchar{\string|}
\edef\gls@levelchar{\string!}
\edef\gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\gls@actualchar'}
\write\glswrite{encap '\gls@encapchar'}
\write\glswrite{level '\gls@levelchar'}
\write\glswrite{quote '\gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string"}
\write\glswrite{preamble \string\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string\string\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string\string\string\n\string"}
\write\glswrite{item_1 \string\string\string\n\string"}
\write\glswrite{item_2 \string\string\string\n\string"}
\write\glswrite{item_01 \string\string\string\n\string"}
\write\glswrite{item_x1

```

```

\string\string\relax \string\glsresetentrylist\string\n
\string}
\write\glswrite{item_12 \string\string\%\string\n\string}
\write\glswrite{item_x2
\string\string\relax \string\glsresetentrylist\string\n
\string}
\write\glswrite{delim_0 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
\string\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
\string\string\}\string\relax
\string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string"glssymbols\string}
\write\glswrite{numhead_positive \string"glnumbers\string}
\write\glswrite{page_compositor \string"glscompositor\string}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2020/03/19 v4.46 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
\ifcsundef{@glscompstyle@#1}%
{%
\csdef{@glscompstyle@#1}{#2}%
}%
}

```

```

    {%
      \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
    }%
  }
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
        {%
          \glsresetsubentrycounter
          \glsinlineparentchildseparator
          \def\gls@inlinesubsep{}%
          \def\gls@inlinepostchild{\glsinlinepostchild}%
        }%
        {}%
      \def\gls@inlinesep{\glsinlineseparator}%
    }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
      ##3\glspostdescription\space ##5%
  \renewcommand*\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.%
  }
}
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%

```

```

\renewcommand*\glossaryentryfield}[5]{%
  \item[\glentryitem{##1}\glstarget{##1}{##2}]%
  \mbox{}\par\nobreak\@afterheading
  ##3\glspostdescription\space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
  \par
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}\glstarget{##1}{##2}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
  \renewcommand*\glossarysubentryfield}[6]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{##3}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
  }%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscompstyle@listdotted}%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
  }%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*\glossaryentryfield}[5]{%

```

```

\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
        \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%

```



```

\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%

```

```

\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else
\settothewidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
\hangindent\glstreeindent
\parindent\glstreeindent
\fi
\makebox[Opt][r]{\makebox[\glstreeindent][l]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription \space ##5\par
\def\@gls@prevlevel{0}%

```

```

}%
\renewcommand{\glossarysubentryfield}[6]{%
  \ifnum##1=1\relax
    \glosssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \@ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
      \settowidth{\gls@tmplen}{\textbf{%
        \csname @glswidestname\romannumeral##1\endcsname\space}}}%
    \ifnum\@gls@prevlevel<##1\relax
      \setlength\glstreeindent\gls@tmplen
      \addtolength\glstreeindent\parindent
      \parindent\glstreeindent
    \else
      \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
        \settowidth{\glstreeindent}{\textbf{%
          \@glswidestname\space}}{%
          \settowidth{\glstreeindent}{\textbf{%
            \csname @glswidestname\romannumeral\@gls@prevlevel
              \endcsname\space}}}%
        \addtolength\parindent{-\glstreeindent}%
        \setlength\glstreeindent\parindent
      \fi
      \fi
      \hangindent\glstreeindent
      \makebox[0pt][r]{\makebox[\gls@tmplen][l]{%
        \textbf{\glstarget{##2}{##3}}}}%
      \ifx##5\relax\relax
      \else
        (##5)\space
      \fi
      ##4\glspostdescription\space ##6\par
      \def\@gls@prevlevel{##1}%
    }%
  }%
}%
\compatglossarystyle{almtreegroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%

```

```

\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenoname}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
}%
\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%

```

```

\renewcommand*\glossaryentryfield}[5]{%
  \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*\glossarysubentryfield}[6]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscompstyle@super}%
}%

```

```

\compatglossarystyle{super3col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%

```

8.6 Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \PackageWarningNoLine{glossaries}%
      {No language module detected for ‘\this@dialect’.\MessageBreak
      Language modules need to be installed separately.\MessageBreak
      Please check on CTAN for a bundle called\MessageBreak
      ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
  }%
}%
}
```

8.7 Rollback v4.46 (glossaries-prefix-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2020/03/19 v4.46 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[v4.46]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}}%
}
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
```

```

\let\@glo@entryprefixfirstplural\@gls@default@value
}
\appto\@newglossaryentryposthook{%
\gls@assign@field{}\@glo@label}{prefix}\@glo@entryprefix}%
\gls@assign@field{}\@glo@label}{prefixplural}\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefix\endcsname}\@glo@label}{prefixfirst}%
  {\@glo@entryprefixfirst}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefixplural\endcsname}\@glo@label}%
  {prefixfirstplural}\@glo@entryprefixfirstplural}%
}
\newcommand*\@glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
\newcommand*\@glsentryprefixfirstplural}[1]{%
\csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*\@glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
\newcommand*\@glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*\@Glsentryprefixfirst}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixfirstplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefix}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newcommand*\@ifglshasprefix}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixplural}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixplural}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirst}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixfirst}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirstplural}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixfirstplural}%

```



```

    {#3}%
    {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
    {\@glsxtr@record{#1}{#2}{glslink}}%
  }%
}
\newcommand{\glsprefixsep}{-}
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
    {\@pgls@{#1}{#2}}%
    {\@pgls@{#1}{#2}[{}]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{-}%
    }%
    {%
      \ifglschasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{-}%
    }%
    \@gls@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@pglsp1@{#1}{#2}}%
    {\@pglsp1@{#1}{#2}[{}]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{-}%
    }%
    {%
      \ifglschasprefixfirstplural{#2}%
        {\glsentryprefixfirstplural{#2}\glsprefixsep}{-}%
    }%
    \@glspl@{#1}{#2}[#3]%
  }%
}
}

```

```

\newrobustcmd{\Pgl}{\@gls@hyp@opt\Pgl}
\newcommand*{\@Pgl}[2][\@Pgl]{%
  \new@ifnextchar[%
    {\@Pgl@{#1}{#2}}%
    {\@Pgl@{#1}{#2}[]}%
  }
\def\@Pgl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifgls@hasprefix{#2}%
      {%
        \Glsentryprefix{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
    {%
      \ifgls@hasprefixfirst{#2}%
      {%
        \Glsentryprefixfirst{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
  }%
}
\newrobustcmd{\Pglsp}{\@gls@hyp@opt\Pglsp}
\newcommand*{\@Pglsp}[2][\@Pglsp]{%
  \new@ifnextchar[%
    {\@Pglsp@{#1}{#2}}%
    {\@Pglsp@{#1}{#2}[]}%
  }
\def\@Pglsp@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifgls@hasprefixplural{#2}%
      {%
        \Glsentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
  }%
}

```

```

}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glsp1@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@gls@hyp@opt\PGLS}
\newcommand*{\@PGLS}[2][ ]{%
  \new@ifnextchar [%
    {\@PGLS@{#1}{#2}}%
    {\@PGLS@{#1}{#2} [ ]}%
  }
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
}
\newrobustcmd{\PGLSp1}{\@gls@hyp@opt\PGLSp1}
\newcommand*{\@PGLSp1}[2][ ]{%
  \new@ifnextchar [%
    {\@PGLSp1@{#1}{#2}}%
    {\@PGLSp1@{#1}{#2} [ ]}%
  }
}
\def\@PGLSp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {\mfirstucMakeUppercase{\glentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%

```

```

        \ifglshasprefixfirstplural{#2}%
        {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}%
    }%
    \@GLSpl@{#1}{#2}{#3}%
} %
}

```

8.8 Rollback v4.46 (glossary-hypernav-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2020/03/19 v4.46 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
  \edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
  \@glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
  \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@prerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
    \fi
  }
\let\gls@hypergroup@prerun\relax
\AtEndDocument{\gls@hypergroup@prerun}
\newcommand*{\@gls@hypergroup}[2]{%
  \@ifundefined{@gls@hypergroup@list@#1}{%
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
  }{%
    \expandafter\let\expandafter\@gls@tmp
      \csname @gls@hypergroup@list@#1\endcsname
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
      \@gls@tmp,#2}%
  }%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergroup@list@\@glo@type}%

```

```

{%
  \def\@gls@list{%
}%
{%
  \expandafter\let\expandafter\@gls@list
  \csname @gls@hypergroup\list@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
  \@gls@between
  \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
  \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
  \let\@gls@between\glshypernavsep
}%
}
\newcommand*\gls@hypernavsep{\space\textbar\space}
\newcommand*\gls@symbolnav{%
  \glsnavhyperlink{gls@symbols}{\gls@getgrouptitle{gls@symbols}}%
  \gls@hypernavsep
  \glsnavhyperlink{gls@numbers}{\gls@getgrouptitle{gls@numbers}}%
  \gls@hypernavsep
}

```

8.9 Rollback v4.46 (glossary-inline-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2020/03/19 v4.46 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\@gls@inlinesep{}%
    \def\@gls@inlinesubsep{}%
    \def\@gls@inlinepostchild{}%
  }%
  {\gls@postinline}%
\renewcommand*\@glossaryheader{}%
\renewcommand*\@gls@groupheading[1]{}%
\renewcommand*\@glossentry[2]{%
  \gls@inlinedopostchild
  \gls@inlinesep
  \gls@entryitem{##1}%
  \gls@inlinenameformat{##1}%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \gls@inlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
}

```

```

    {%
      ##2%
    }%
  }%
  {%
    \ifglshasdesc{##1}%
    {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
    {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
  }%
  \ifglshaschildren{##1}%
  {%
    \glsresetsubentrycounter
    \glsinlineparentchildseparator
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{\glsinlinepostchild}%
  }%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glssubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{}
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{}
\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.10 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%

```

```

\par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*\glslistgroupheaderfmt}[1]{#1}
\newcommand*\glslistnavigationitem}[1]{\item[#1]}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\begin{description}}{\end{description}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]%
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[]\makebox[\glslistdottedwidth][1]{%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut\glossentrydesc{##1}}%
  \renewcommand*{\subglossentry}[3]{%
    \item[]\makebox[\glslistdottedwidth][1]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut\glossentrydesc{##2}}%
  }
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
}

```

8.11 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[lp{\glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%}
  \renewcommand*{\glsgroupheading}[1]{%}
  \renewcommand*{\glossentry}[2]{%
    \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*\glsgroupheading[1]{ }%
            \renewcommand{\glossentry}[2]{%
                \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
                \glossentrydesc{##1} & ##2\tabularnewline
            }%
            \renewcommand{\subglossentry}[3]{%
                &
                \glssubentryitem{##2}%
                \glstarget{##2}{\strut}\glossentrydesc{##2} &
                ##3\tabularnewline
            }%
            \ifglsgroupskip
                \renewcommand*\glsgroupskip{ }%
            \else
                \renewcommand*\glsgroupskip{ & & \tabularnewline}%
            \fi

```

```

}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l|l|l|l}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&

```

```

        \bfseries \symbolname&
        \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
}

```

8.12 Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput

```

```

\setglossarystyle{long4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{altlong4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
            \@outputpage
            \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
          \fi
        \fi
      \fi
    }%
}

```

```

        \fi
        \fi
        \global\@colroom\@colht
        \global\vsizel\@colht
        {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
    \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
    \fi
}
}
}

```

8.13 Rollback v4.46 (glossary-longragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
    \newlength\glsdescwidth
    \setlength\glsdescwidth{0.6\hsize}
}{%
\@ifundefined{glspagelistwidth}{%
    \newlength\glspagelistwidth
    \setlength\glspagelistwidth{0.1\hsize}
}{%
\newglossarystyle{longragged}{%
    \renewenvironment{theglossary}{%
        {\begin{longtable}{1>\raggedright}p{glsdescwidth}}}%
        {\end{longtable}}}%
    \renewcommand*\glossaryheader{}%
    \renewcommand*\glsgroupheading}[1]{}%
    \renewcommand\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription\space ##2%
        \tabularnewline
    }%
    \renewcommand\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}%
        \glspostdescription\space ##3%
        \tabularnewline
    }%
}
}
}

```

```

}%
\ifglsgroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & \tabularnewline}%
\fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}||}%
    {\end{longtable}}%
    \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*\glossaryheader{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
  }
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*\glossaryheader{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
  }
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
\ifglsgroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & & \tabularnewline}%
\fi

```

```

}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```



```

        {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
          >{\raggedright}p{\glspagelistwidth}}}%
      {\end{longtable}}%
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

8.14 Rollback v4.46 (glossary-mcols-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.46]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\gls mcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\gls mcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }
  }

```

```

    }%
    {\end{multicols}}%
}
\newglossarystyle{mcolindexgroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
\setglossarystyle{index}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glscols}{\noindent\glstreenavigationfmt{\glsnavigation}}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
}
\newglossarystyle{mcoltree}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glscols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
\setglossarystyle{mcoltree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreesspannav}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenoname}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand*{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%

```

```

}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}

```

8.15 Rollback v4.46 (glossary-super-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength\glsdescwidth{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength\glspagelistwidth{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
  \begin{supertabular}{lp{\glsdescwidth}}%
  {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%

```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
\newglossarystyle{superheader}{%
    \setglossarystyle{super}%
\renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
        \bfseries\descriptionname\tabularnewline}%
        \tabletail{}}%
        \begin{supertabular}{|l|p{\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}{|l|p{\glsdescwidth}p{\glspagelistwidth}}%
        {\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
}%
\ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
\else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{super3colborder}{%
\setglossarystyle{super3col}%
\renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
        \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }

```

```

}
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
    {\begin{supertabular}{llll}}{%
    \end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    \tabletail{}}%

```

```

        \begin{supertabular}{|l|l|l|}%
        \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{\glsdescwidth}lp{\glspagelistwidth}}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colheader}{%
\setglossarystyle{super4colheader}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}}%
\begin{supertabular}{|lp{\glsdescwidth}lp{\glspagelistwidth}}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colborder}{%
\setglossarystyle{super4colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
\setglossarystyle{super4colheaderborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline
\bfseries\entryname &
\bfseries\descriptionname &

```



```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
  {1|p{\glsdescwidth}|1|p{\glspagelistwidth}|}%
\end{supertabular}}%
}

```

8.16 Rollback v4.46 (glossary-superragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{\tabletail}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
}

```

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
   \tabularnewline}%
   \tabletail{}}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
   \bfseries \descriptionname\tabularnewline\hline}%
   \tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}%
   \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
  \setglossarystyle{superragged3col}%
}

```

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superragged3colheader}{%
  \setglossarystyle{superragged3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
     \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
     >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{supertabular}}%
}
\newglossarystyle{superragged3colheaderborder}{%
  \setglossarystyle{superragged3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
     \bfseries\entryname&\bfseries\descriptionname&
     \bfseries\pagelistname\tabularnewline\hline}%
     \tabletail{\hline}%
     \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
     {\end{supertabular}}%
}
\newglossarystyle{altsuperragged4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
     \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}|}}%
     {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}

```

```

}
\newglossarystyle{altsuperragged4colheader}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
  \setglossarystyle{altsuper4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {l|>{\raggedright}p{\glsdescwidth}l|}%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {l|>{\raggedright}p{\glsdescwidth}l|}%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{supertabular}}}%
}

```

8.17 Rollback v4.46 (glossary-tree-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem

```

```

{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
     \let\item\glstreeitem
     \let\subitem\glstreesubitem
     \let\subsubitem\glstreesubsubitem
    }%
    {\par}%
  \renewcommand*\{\glossaryheader}{}%
  \renewcommand*\{\glsgroupheading}[1]{}%
  \renewcommand*\{\glossentry}[2]{%
    \item\glstreeitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space\glossentrysymbol{##1}}{}}%
    \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifcase##1\relax
      % level 0
      \item
    \or
      % level 1
      \subitem
      \glssubentryitem{##2}%
    \else
      % all other levels
      \subsubitem
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \ifglshassymbol{##2}{\space\glossentrysymbol{##2}}{}}%
    \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
  }%
  \renewcommand*\{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
  \setglossarystyle{index}%
  \renewcommand*\{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
    \indexspace
  }%
}
\newglossarystyle{indexhypergroup}{%
  \setglossarystyle{index}%
  \renewcommand*\{\glossaryheader}{%

```

```

        \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
\newglossarystyle{tree}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
    \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
  }%
  \renewcommand*\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%

```

```

\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
  \forglentries[\@gls@type]{\@glo@label}%
  {%
  \ifglshasparent{\@glo@label}%
  {}%
  {%

```



```

\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\@glswidestname\space}}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def\@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.18 Rollback v4.49 (glossaries-2021-11-01.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareCurrentRelease{v4.49}{2021-11-01}
\ProvidesPackage{glossaries}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
  {%
    \@gls@docloadedtrue
  }%
  {%
    \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
  }
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
  {true,false,showtargets,showaccsupp}[true]{%
    \ifcase\gls@debug@nr\relax
      % debug=true
      \@gls@debugtrue
      \renewcommand*{\GlossariesWarning}[1]{%
        \PackageWarning{glossaries}{##1}%
      }%
      \renewcommand*{\GlossariesWarningNoLine}[1]{%
        \PackageWarningNoLine{glossaries}{##1}%
      }%
      \let\gls@showtarget\@gobble
      \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \or

```

```

% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
  \nfss@text{\glsshowtargetfont [#1]}%
\else
  \ifinner
    {\glsshowtargetfont [#1]}%
  \else
    \glsshowtargetouter{#1}%
  \fi
\fi
}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*\@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%

```

```

\endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{}
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}{
  {\newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%

```

```

\renewcommand*{\@glossarysec}{#1}}
\newcommand*{\@glossarysecstar}{*}
\newcommand*{\@glossaryseclabel}{}
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{}%
    \label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{}%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
  \renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\@glo@seeautonumberlist{%
    \def\@glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\gls@esclocationstrue
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}[=v4.49]}
\@gls@declareoption{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\theH@glossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \ifx\@glo@sort\@glsdefaultsort
      \let\@glo@sort\@glo@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort
    \glsprestandardsort{\@glo@sort}{##1}{##2}%
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
  }%
  \def\@gls@setsort##1{%
  \let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi

```

```

\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*{\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
{}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi

```



```

}%
\let\@glo@check@sortallowed@gobble
}
\newcommand*\@gls@setupsort@none{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*\glsdefmain{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*\gls@tr@set@main@toctitle{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*\glsdefaulttype{main}
\newcommand*\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
\ifbool{glscompatible-3.07}%
{}%
{}%
\providecommand*\printacronyms[1]{}%

```

```

        \printglossary[type=\acronymtype,#1]}%
    }%
\fi
}
\newcommand*{\@gls@do@acronymsdef}{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*{\acronymtype}{acronym}%
    \newcommand*{\gls@tr@set@acronym@toctitle}{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*{\@gls@acronymlists}{}
\newcommand*{\@addtoacronymlists}[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*{\DeclareAcronymList}[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@gls@acronymlists}{#1}%
}

```

```

\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\@gls@sanitizedesc}{%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
}

```

```

\@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
\ifdefvoid\@glo@sort
{}%
{}%
\expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
\bgroup
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egroup
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent@\@secondoftwo
\let\@text@composite@x\@secondoftwo
\let\@tabacckludge\@secondoftwo
\expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let'\@firstofone
\let'\@firstofone
\let\~\@firstofone
\let\"@firstofone
\let\u\@firstofone
\let\t\@firstofone
\let\d\@firstofone
\let\r\@firstofone
\let=\@firstofone
\let.\@firstofone
\let~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%

```

```

\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifglS@sanitize@symbol
\glssetnoexpandfield{symbol}%
\glssetnoexpandfield{symbolplural}%
\else
\glssetexpandfield{symbol}%
\glssetexpandfield{symbolplural}%
\fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
\ifglssanitizesort
\glssetnoexpandfield{sortvalue}%
\renewcommand*{\@glS@noidx@setsanitizesort}{%
\glssanitizesorttrue
\glssetnoexpandfield{sortvalue}%
}%
\else
\glssetexpandfield{sortvalue}%
}

```

```

        \renewcommand*\@gls@noidx@setsanitizesort}{%
            \glssanitizesortfalse
            \glssetexpandfield{sortvalue}%
        }%
    \fi
}
\glssanitizesorttrue
\glssetnoexpandfield{sortvalue}%
\newcommand*\@gls@noidx@setsanitizesort}{%
    \glssanitizesortfalse
    \glssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glssanitizesort}{#1}%
    \ifglssanitizesort
        \glssetnoexpandfield{sortvalue}%
    \else
        \glssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
        deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glssetexpandfield{name}%
            \glssetexpandfield{symbol}%
            \glssetexpandfield{symbolplural}%
            \glssetexpandfield{desc}%
            \glssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
        \let\glsifusetranslator\@secondoftwo
    }%
    {%
        \@ifpackageloaded{babel}%
        {%
            \IfFileExists{translator.sty}%
            {%
                \RequirePackage{translator}%
                \let\glsifusetranslator\@firstoftwo
            }%
        }%
    }%
}

```

```

        {}%
    }%
    {}%
}
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
      \glstranslatetrue
      \renewcommand*\@gls@usetranslator{%
        \@ifpackageloaded{polyglossia}%
        {%
          \let\glsifusetranslator\@secondoftwo
        }%
        {%
          \@ifpackageloaded{babel}%
          {%
            \IfFileExists{translator.sty}%
            {%
              \RequirePackage{translator}%
              \let\glsifusetranslator\@firstoftwo
            }%
            {}%
          }%
          {}%
        }%
      }%
    \or
      \glstranslatefalse
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \or
      \glstranslatetrue
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \fi
  }
}
\glstranslatefalse

```



```

\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
  \glstranslatetrue
  \let\glsifusetranslator\@firstoftwo
}%
{%
  \for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
  {
    \@ifpackageloaded{\gls@thissty}%
    {%
      \glstranslatetrue
      \@endfortrue
    }%
    {}%
  }
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifbool{glsacrdescription}%
{}%
{%
  \renewcommand*{\@gls@sanitizedesc}{}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{}

```

```

\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{%
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\languagename}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘restoremakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}
}

```

```

}%
{%
  \ifdefequal\@makeglossary\@gobble
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makeglossaries\space has already been used)}%
  }%
  {%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been restored}%
    \let\@domakeglossaries\@firstofone
  }%
}%
}
\newcommand*\@do@write@glslabels{%
  \AtEndDocument{\@do@write@glslabels}%
  \let\@do@write@glslabels\relax
}
\newcommand*\@@do@write@glslabels{%
  \newwrite\@glsl@labelsfile
  \immediate\openout\@glsl@labelsfile=\jobname.glslabels
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@glsl@fulllist{,\@ignored@glossaries}}%
  \forallglstentries[\@glsl@fulllist]{\@glstentry}%
  {%
    \ifdefempty{\@glstentry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@glst@thisentryname{}%
      \else
        \letcs{\@glst@thisentryname}{glo@\glstetoklabel{\@glstentry}@name}%
        \@onelevel@sanitize\@glst@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@glstentry\do@write@glslabelsname{\@glst@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}
\newcommand*\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@glsl@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname\@gobble}
  \@glsl@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglslautomake

```

```

\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
  \ifnum\gls@automake@nr=1\relax
    \glsautomakefalse
  \else
    \glsautomaketrue
  \fi
  \ifglsautomake
    \renewcommand*\@gls@doautomake}{%
      \PackageError{glossaries}{You must use
      \string\makeglossaries\space with automake=true}
      {%
        Either remove the automake=true setting or
        add \string\makeglossaries\space to your document preamble.%
      }%
    }%
  \else
    \renewcommand*\@gls@doautomake}{}%
  \fi
}
\glsautomakefalse
\newcommand*\@gls@doautomake{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*\@gls@writefiles{\@gls@writefiles}%
  \else
    \let\@gls@writefiles\@empty
  \fi
}
\gls savewritesfalse
\let\@gls@writefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \booltrue{glscompatible-3.07}%
  }%
  }%
}
\boolfalse{glscompatible-2.07}
\let\@gls@original@makeglossary\makeglossary
\let\@gls@original@glossary@glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
  glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
  instead. If you \MessageBreak need the original definition of
  \string\makeglossary\space use \MessageBreak the package options

```

```

kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict))%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
\GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict))%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\@gls@debug@val\@gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\@gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict))%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%

```

```

\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*\{printsymbols}[1][\]{\printglossary[type=symbols,##1]}%
\newcommand*\{gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*\{gls@do@numbersdef}{%
\newcommand*\{gls@numbersdef}{%
\newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
\newcommand*\{printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
\newcommand*\{gls@tr@set@numbers@toctitle}{%
\translatelet{\glossarytoctitle}{Numbers (glossaries)}%
}%
}%
\@gls@declareoption{index}{%
\ifx\@gls@do@indexdef\@empty
\let\@gls@do@indexdef\@gls@indexdef
\fi
}
\@gls@declareoption{noglossaryindex}{%
\let\@gls@do@indexdef\relax
}
\newcommand*\{gls@do@indexdef}{%
\newcommand*\{gls@indexdef}{%
\newglossary[ilg]{index}{ind}{idx}{\indexname}%
\newcommand*\{printindex}[1][\]{\printglossary[type=index,##1]}%
\newcommand*\{newterm}[2][\]{%
\newglossaryentry{##2}%
{type={index},name={##2},description={\nopostdesc},##1}}
\let\@gls@do@indexdef\relax
}%
}%
\@for\CurrentOption :=\@declaredoptions\do{%
\ifx\CurrentOption\@empty
\else
\@expandtwoargs
\in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
\ifin@
\@use@ption
\expandafter \let\csname ds@\CurrentOption\endcsname\@empty
\fi
\fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.49]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%

```

```

nomain,noglossaryindex}
\newcommand*\setupglossaries[1]{%
  \renewcommand*\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglssubentrycounter
    \let\@gls@dosubentrycounterdef\relax
  \else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
  \@gls@do@acronymsdef
  \@gls@do@numbersdef
  \@gls@do@symbolssdef
  \@gls@do@indexdef
  \@gls@doentrycounterdef
  \@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%
  {%
    \ifcsundef{chapter}{}%
    {%
      \let\@gls@old@chapter\@chapter
      \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
      \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
    }%
  }%
  {}
\newcommand*\@gls@onlypremakeg{}
\newcommand*\@onlypremakeg[1]{%
  \ifx\@gls@onlypremakeg\@empty

```

```

        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
    \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
    \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle[1]{%
    \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}
\providecommand*\symbolname{Symbol}
\providecommand*\pagelistname{Page List}
\providecommand*\glsymbolsgroupname{Symbols}
\providecommand*\glsnumbersgroupname{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glsup{glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
    \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
    \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
    \RequirePackage{tracklang}
    \@gls@usetranslator
    \ifpackageloaded{translator}
    {%
        \ifboolexpr
        {
            test {\ifdefstring{\trans@languages}{English}}
            and not
            test {\ifdefstring{bbl@loaded}{english}}
        }
    {%
        \let\glsifusetranslator\@secondoftwo
    }
    }

```



```

}%
{%
\usedictionary{glossaries-dictionary}%
\renewcommand*{\addglossarytocaptions}[1]{%
\ifcsundef{captions#1}{}%
{%
\expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
\expandafter\toks@\expandafter{\@gls@tmp
\renewcommand*{\glossaryname}{\translate{Glossary}}}%
}%
\expandafter\edef\csname captions#1\endcsname{\the\toks@}%
}%
}%
}%
{}%
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
{%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
}%
}%
}%
{}%
\glsifusetranslator
{%
\renewcommand*{\glssettoctitle}[1]{%
\ifcsdef{gls@tr@set@#1@toctitle}%
{%
\csuse{gls@tr@set@#1@toctitle}%
}%
{%
\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
}%
}%
\renewcommand*{\glossaryname}{\translate{Glossary}}%
\renewcommand*{\acronymname}{\translate{Acronyms}}%
\renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
\renewcommand*{\descriptionname}{%
\translate{Description (glossaries)}}%
\renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
\renewcommand*{\pagelistname}{%
\translate{Page List (glossaries)}}%

```

```

        \renewcommand*{\glssymbolsgroupname}{%
          \translate{Symbols (glossaries)}}%
        \renewcommand*{\glsnumbersgroupname}{%
          \translate{Numbers (glossaries)}}%
      }{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*{\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gls@istfilebase}{#1}%
  \ifglxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*{\gls@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{}
\newcommand*{\glscompositor}{.}
\newcommand*{\glsSetCompositor}[1]{%
  \renewcommand*{\glscompositor}{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*{\@glsAlphaCompositor}{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphaCompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxindywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*{\gls@suffixF}{}
\newcommand*{\glsSetSuffixF}[1]{%
  \renewcommand*{\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*{\gls@suffixFF}{}
\newcommand*{\glsSetSuffixFF}[1]{%
  \renewcommand*{\gls@suffixFF}{#1}}

```

```

}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*\glsnumberformat}[1]{\glsnumberformat{#1}}%
}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*\glossarypostamble{}
\newcommand*\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand*\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand*\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\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*{@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
            }%
        }%
    }%
}

```



```

\@for\@gls@ctr:=#1\do{%
  \edef\@do@addcounter{%
    \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
    {%
      \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
        \noexpand\@gls@ctr}%
    }%
  }%
  \@do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
  \glsnoxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglsxindy
\newcommand*\@glsaddxdyattribute[2]{%
  \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
  \expandafter\toks@\expandafter{\@xdylocref}%
  \edef\@xdylocref{\the\toks@ ^^J}%
  (markup-locref
:open \string"\glstildechar n%
  \expandafter\string\csname glsX#2X#1\endcsname
  \string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string"))%
  \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
    \setentrycounter{##1}{#2}\csname #1\endcsname{##2}%
  }%
}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
  \edef\@xdyattributelist{#1}%
  \else
  \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
\@for\@this@counter:=\@xdycounters\do{%
  \protected@edef\gls@do@addxdyattribute{%
    \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
  }
  \gls@do@addxdyattribute
}

```

```

    }%
    \let\GlsAddXdyCounters\@disabled@gl saddx dycounters
  }
  \@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \gl snoxindywarning\GlsAddXdyAttribute}
\fi
\ifgl sxindy
\newcommand*{\@gl s@addpredefinedattributes}{%
  \GlsAddXdyAttribute{gl snumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{gl shypernumber}
  \GlsAddXdyAttribute{hyperm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{gl signore}
}
\else
  \let\@gl s@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifgl sxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^~J
      (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \gl snoxindywarning\GlsAddXdyAlphabet}
\fi
\ifgl sxindy
  \newcommand*{\@gl s@xdy@locationlist}{%
    roman-page-numbers,%

```

```

    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      \string"roman-numbers-lowercase\string"%
    }%
\else
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      :sep \string"\@gls@roman\string"%
    }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@gls@alphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxsindy

```



```

\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][[]]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\glsopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsopenbrace"
#1
:sep "\glsclosebrace\glsopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnoxywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"}
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnoxywarning\GlsSetXdyLocationClassOrder}

```

```

\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string")}%
    }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{-}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][ ]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }

```

```

\AtBeginDocument{%
  \ifx\gls@codepage\empty
    \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxindywarning\GlsSetXdyCodePage}
\fi
\ifglsxindy
  \ifgls@xindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glsnumbers\string^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\empty\else#2\fi}%
}
\newcommand*\forglentries[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forglentries[\@@this@glo@]{#2}{#3}%
  }%
}
\newcommand\ifglossaryexists{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

```

```

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
  }%
}

```

```

\protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
\expandafter\forglentries\expandafter
  [\csname glo@\@gls@thislabel @type\endcsname]
{\glo@label}%
{%
  \letcs\glo@parent{glo@\glo@label @parent}%
  \ifdefequal\@gls@thislabel\glo@parent
  {%
    \def\do@glshaschildren{#2}%
    \@endfortrue
  }%
  }%
  \do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*\{ifglshasdesc}[3]{%
  \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*\{ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{\no@post@desc}%
  {#2}%
  {#3}%
}
\newrobustcmd*\{ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}
\newrobustcmd*\{ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}

```

```

    }%
  }
  \newrobustcmd*{\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@glsetoklabel{#1}@short}%
    \ifdefempty\@glo@short
      {#3}%
    {%
      \ifdefequal\@glo@short\@gls@default@value
        {#3}%
        {#2}%
      }%
    }
  }
  \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}@#1}%
          \ifdefempty\@glo@thisvalue
            {%
              #4%
            }%
          {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
              {%
                #4%
              }%
            {%
              \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
              }%
            }%
          }%
        }%
      }%
    }
  }

```



```

\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakegnewglossary
\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{\@gls@nonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@gls@nextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*{\@gls@savenonumberlist}[1]{%
\newcommand*{\@gls@initnonumberlist}{%
\newcommand*{\@gls@storenonumberlist}[1]{%

```

```

\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
      {%
        \cslet{glo@\glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
      }%
    }%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@gls@noname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}
}

```

```

\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*{\@glsdefaultplural}{}
\newcommand*{\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
    Maybe makeglossaries + rerun required}%
  \else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
  \fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%
  }
}

```

```

        \@@gls@noexpand@field{#2}{#3}{#4}%
    }%
}
\newcommand{\@@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {%
    \ifglsentryexists{#1}%
    {%
      \xifinlist{#1}{\gls@docdef@labellist}%
      {%
        \GlossariesWarning{Entry ‘#1’ has already been defined in the
          document environment (ignoring new definition)}%
      }%
      {%
        \xifinlist{#1}{\gls@glsdefs@labellist}%
        {%
          \gls@defglossaryentry{#1}{#2}%
          \gls@docdef@addlabel{#1}%
          \@gls@writedef{#1}%
        }%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in
            the preamble (in which case remove the duplicate definition)
            or you have just updated to glossaries.sty
            v4.47 (in which case delete ‘\jobname.glsdefs’)}%
        }%
      }%
    }%
  }%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}

```

```

        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
    }%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
    \gls@enablesavenonumberlist
    \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
    \makeatletter
    \InputIfFileExists{\jobname.glsdefs}{-}{-}%
    \@gls@restreat
    \undef\@gls@restreat
    \gls@defdocnewglossaryentry
}
\newcommand*\glsdefs@newdocentry}[2]{%
    \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
        (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
        \gls@glsdefs@addlabel{#1}%
        \gls@defglossaryentry{#1}{#2}%
    }%
}
\newcommand*\@gls@writedef}[1]{%
    \immediate\write\@gls@deffile
    {%
        \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
        \expandafter\@gobble\string\{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
        \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
        \ifdef\glo@value
        {%
            \@onelevel@sanitize\glo@value
            \immediate\write\@gls@deffile
            {%
                \expandafter\@firstoftwo\@gls@map
                =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
                \glspercentchar
            }%
        }%
        }%
    }%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
}

```

```

    \expandafter\@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```



```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
{}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
{}%
}%
{}%
}

```

```

\expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2} []}}%
\csdef{@gls@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}{%
  {%
    \PackageError{glossaries}%
    {Can't define '\string#6' as helper command
     '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}%
    }%
}%
  {%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
\csdef{@Gls@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}{%
  {%
    \PackageError{glossaries}%
    {Can't define '\string#7' as helper command
     '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}%
    }%
}%
  {%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
\csdef{@GLS@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%

```

```

    {%
      \PackageError{glossaries}{Key ‘#1’ already exists}{}%
    }%
  }
\newcommand{\glsfieldxdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \protected@csxdef{glo@\glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfielddedef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \protected@csedef{glo@\glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \expandafter\def\csname glo@\glo@label @#2\endcsname{#3}%
    }%
  }%
}

```

```

    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newcommand{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\ifglsfielddfeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal
    \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}

```

```

}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand*{\glswritedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\glo@desc}%
\gls@assign@field{\glo@desc}{#1}{descplural}{\glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
\bgroup
\let\org@newglossaryentryprehook\@newglossaryentryprehook
\long\def\@newglossaryentryprehook{%
\long\def\glo@desc{#3\leavevmode\nskip\nopostdesc}%
\org@newglossaryentryprehook
}%
\renewcommand*{\gls@assign@desc}[1]{%
\global\cslet{glo@\glsdetoklabel{#1}@desc}{\glo@desc}%
\global\cslet{glo@\glsdetoklabel{#1}@descplural}{\glo@desc}%
}
\gls@defglossaryentry{#1}{#2}%
\egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
\ifglsentryexists{#1}{}%
{\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
\let\GlsSetQuote\gls@nosetquote
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\let\glslabel\glo@label
\let\glo@name\gls@name
\let\glo@desc\gls@desc
\let\glo@descplural\gls@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@initnonnumberlist
\def@glo@useri{}%
\def@glo@userii{}%
\def@glo@useriii{}%
\def@glo@useriv{}%
\def@glo@userv{}%
\def@glo@uservi{}%
\def@glo@short{}%
\def@glo@shortpl{}%
\def@glo@long{}%
\def@glo@longpl{}%
@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef@glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\glo@label}{type}{\glo@type}%
\protected@edef@glo@type{\glsentrytype{\glo@label}}%
\ifcsundef{glolist@\glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{}%
\ifignoredglossary@glo@type
{%
  \ifx@glo@desc@glsnodesc
  \let@glo@desc@empty
  \fi
}%
{}%
{}%

```

```

\protected@edef\glo@list@{\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
  \glo@list@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@gls@noname
        \expandafter\let\expandafter\@glo@name
        \csname glo@\@glo@parent @name\endcsname
        \ifx\@glo@plural\@gls@default@value
          \expandafter\let\expandafter\@glo@plural
          \csname glo@\@glo@parent @plural\endcsname
        \fi
      \fi
    }%
  }%
  \PackageError{glossaries}%
  {%
    Invalid parent ‘\@glo@parent’
    for entry ‘\@glo@label’ - parent doesn’t exist%
  }%
  {%
    Parent entries must be defined before their children%
  }%
  \def\@glo@parent{}%
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\gls@pluralsuffix}%

```

```

    {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glotype@\@glo@type @counter}%
  {%
    \def\@glo@defaultcounter{\glscounter}%
  }%
  {%
    \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
  }%
  \gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
  \gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
  \gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
  \gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
  \gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
  \gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
  \gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
  \gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
  \gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
  \gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
  \gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
  \@glsnoname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
  {%
    \csxdef{glo@\@glo@label @numberlist}{%
      \noexpand\@gls@missingnumberlist{\@glo@label}}%
  }%
  {}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
  \@glsnodesc
  \let\@glo@desc\@gls@default@value

```



```

\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
  \gls@assign@field\expandafter
  {\csname glo@\@glo@label @symbol\endcsname}
  {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
  \noexpand\let\expandafter\noexpand
    \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
  \noexpand\let\expandafter\noexpand
    \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
  \csdef{glo@\@glo@label @index}{}%
}
{%
  \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
\ifdefvoid\@glo@see{%
{%
  \protected@edef\@do@glssee{%
    \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
    \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
  \do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%
\newcommand*{\@newglossaryentryprehook}{%
\newcommand*{\@newglossaryentryposthook}{%
\newcommand*{\@newglossaryentry@defcounters}{%
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%

```



```

\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@%glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\AtBeginDocument{%
\@ifpackageloaded{amsmath}%
{\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
{}}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
\ifmeasuring@
\else
#1%
\fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
{}}%
}
\newcommand*\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}

```

```

}
\newcommand*\glsunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}
\newcommand*\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%

```

```

    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}}%
}
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}}%
}
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\def\@cgls###1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglformat{##2}{##3}%
        \glsunset{##2}%
    \else
        \@gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cglspl@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glsunset{##2}%
    \else
        \@glspl@{##1}{##2}[##3]%
    \fi
}%
\def\@cGls@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cGlsformat{##2}{##3}%
        \glsunset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cGlspl@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cGlsplformat{##2}{##3}%
        \glsunset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csgdef{glo@\glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsused{\@glsentry}%

```

```

        {\immediate\write\@auxout
         {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
        }%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][]{%
  \new@ifnextchar[{\@cgl@#1}{#2}]{\@cgl@#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@cgl@#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2][]{%
  \new@ifnextchar[{\@cGls@#1}{#2}]{\@cGls@#1}{#2}[]}%
}
\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@cGls@#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][]{%
  \new@ifnextchar[{\@cglspl@#1}{#2}]{\@cglspl@#1}{#2}[]}%
}
\def\@cglspl@#1#2[#3]{%
  \GlossariesWarning{\string\cglspl\space is defaulting to
    \string\glspl\space since you haven't enabled entry counting}%
  \@cglspl@#1}{#2}[#3]%
}
\newcommand*{\cglsplformat}[2]{%
  \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\newcommand*{\@cGlspl}[2][]{%
  \new@ifnextchar[{\@cGlspl@#1}{#2}]{\@cGlspl@#1}{#2}[]}%
}
\def\@cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@cGlspl@#1}{#2}[#3]%
}

```

```

}
\newcommand*{\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*{\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*{\glstextformat}[1]{#1}
\newcommand*{\glsentryfmt}{%
  \@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*{\@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }%
        }%
      }%
    }%
  }%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
  }%
}

```



```

{%
  \ifbool{glscompatible-3.07}%
  {%
    \protected@edef\@glo@etext{%
      #1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    \xmakefirstuc\@glo@etext
  }%
  {%
    #1{\Glsentryfirstplural{\glslabel}}%
    {\glsentrydescplural{\glslabel}}%
    {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
  }%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      #2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
    {%
      #1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%

```



```

    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    \ifglsused\glslabel
    {%
      \Glsentryplural{\glslabel}\glsinsert
    }%
    {%
      \Glsentryfirstplural{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \mfirstucMakeUppercase
      {\glsentryplural{\glslabel}\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\glsentryfirstplural{\glslabel}\glsinsert}%
    }%
  }%
}
  }%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glsentrytext{\glslabel}\glsinsert
    }%
  }%
}

```

```

        {%
        \glsentryfirst{\glslabel}\glsinsert
        }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \Glsentrytext{\glslabel}\glsinsert
    }%
    {%
    \Glsentryfirst{\glslabel}\glsinsert
    }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
    }%
    {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
    }%
    }%
    }%
    }%
    {%
    \glscustomtext\glsinsert
    }%
}
\newcommand*{\glsgenacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%
\glsifplural
{%
\glscapscase
{%
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\mfirstucMakeUppercase
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
}%
}%
}%
\glscapscase
{%

```



```

\protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
\xmakefirstuc\gls@text
}
\newcommand*\genplacrfullformat}[2]{%
\glsentrylongpl{#1}#2\space
(\protect\firstacronymfont{\glsentryshortpl{#1}})}%
}
\newcommand*\Genplacrfullformat}[2]{%
\protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
\xmakefirstuc\gls@text
}
\newcommand*\glsdisplayfirst}[4]{#1#4}
\newcommand*\glsdisplay}[4]{#1#4}
\newcommand*\defglsdisplay}[2][\glsdefaulttype]{%
\GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
Use \string\defglsentryfmt\space instead}%
\expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
\edef\@gls@doentrydef{%
\noexpand\defglsentryfmt[#1]{%
\noexpand\ifcsdef{gls@#1@displayfirst}}%
{%
\noexpand\@gls@default@entryfmt
{\noexpand\csuse{gls@#1@displayfirst}}%
{\noexpand\csuse{gls@#1@display}}%
}%
{%
\noexpand\@gls@default@entryfmt
{\noexpand\glsdisplayfirst}%
{\noexpand\csuse{gls@#1@display}}%
}%
}%
}%
\@gls@doentrydef
}
\newcommand*\defglsdisplayfirst}[2][\glsdefaulttype]{%
\GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
Use \string\defglsentryfmt\space instead}%
\expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
\edef\@gls@doentrydef{%
\noexpand\defglsentryfmt[#1]{%
\noexpand\ifcsdef{gls@#1@display}}%
{%
\noexpand\@gls@default@entryfmt
{\noexpand\csuse{gls@#1@displayfirst}}%
{\noexpand\csuse{gls@#1@display}}%
}%
{%
\noexpand\@gls@default@entryfmt
{\noexpand\csuse{gls@#1@displayfirst}}%
{\noexpand\glsdisplay}%
}
}
}

```

```

    }%
  }%
}%
\@gls@doentrydef
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
{%
  \def\@gls@counter{#1}%
}%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
  \glslinkvar{#1}{#2}{#1}%
  \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
  you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3] []{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }
}

```

```

    }{%
      \glstextformat{#3}%
    }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{@gls@link@checkfirsthyper}{%
  \ifglsused{glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \ifglsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}%
}
\newcommand*{@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{glslabel}%
  \@do@wrglossary{#2}%
  \ifKV@glslink@hyper

```



```

        \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \else
        \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \fi
    \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
    \def\@gls@Hcounter{}%
    \ifthenelse{\equal{\@gls@counter}{equation}}%
    {
        \ifcsundef{xatlevel@}%
        {%
            \edef\theglsentrycounter{\expandafter\noexpand
                \csname the\@gls@counter\endcsname}%
        }%
        {%
            \ifx\xatlevel@\@empty
                \edef\theglsentrycounter{\expandafter\noexpand
                    \csname the\@gls@counter\endcsname}%
            \else
                \savecounters@
                \advance\c@equation by 1\relax
                \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
                \ifcsundef{theH\@gls@counter}%
                {%
                    \def\@gls@Hcounter{\theglsentrycounter}%
                }%
                {%
                    \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
                }%
                \protected@edef\theHglsentrycounter{\@gls@Hcounter}%
                \restorecounters@
            \fi
        }%
    }%
    {%
        \edef\theglsentrycounter{\expandafter\noexpand
            \csname the\@gls@counter\endcsname}%
    }%
    \ifx\@gls@Hcounter\@empty
        \ifcsundef{theH\@gls@counter}%
        {%
            \def\theHglsentrycounter{\theglsentrycounter}%
        }%
        {%
            \protected@edef\theHglsentrycounter{\expandafter\noexpand
                \csname theH\@gls@counter\endcsname}%
        }%
    }%
}

```

```

\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1\relax
\def\@glo@prefix{()}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1\relax
\def\@glo@prefix{})%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
}

```

```

    \gls@dosubst
  }%
  \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
    \fi
}

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\!\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%

```

```

\ifx\null#3\null
  \def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
  \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else
      \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
    \fi
  \fi
  \@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkesc{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
      \def\@gls@checkesc{\@gls@checkesc#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@backslashchar}%
  \ifx\null#3\null

```

```

        |def|@@gls@checkesc{|@gls@checkesc#2\\|null}%
    |else
        |def|@@gls@checkesc{|@gls@checkesc#2\#3|null}%
    |fi
    |fi
|@@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@xdycheckquote{\relax}%
    |else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \string"\string"}%
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
    |fi
  |else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \string"}%
    \ifx\null#3\null
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
    |else
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
    |fi
  |fi
  \@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
  ##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
  {\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand|else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
  \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand|fi
\noexpand|else

```



```

\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    \@backslashchar\noexpand\null}%
\noexpand\else
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    ##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength@gls@tmplen
\newcommand*\@glsdohypertarget}[2]{%
  \@glsshowtarget{#1}%
  \settoheight{\@gls@tmplen}{#2}%
  \raisebox{\@gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*\@glsdohyperlink}[2]{%
  \@glsshowtarget{#1}%
  \hyperlink{#1}{#2}%
}
\newcommand*\@glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink@glsdonohyperlink
}%
{%
  \let\@glslink@glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget@glsdohypertarget
}
\newcommand*\@glsdisablehyper{%
  \KV@glslink@hyperfalse
  \let\@glslink@glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand*\@glsenablehyper{%
  \KV@glslink@hypertrue

```

```

\let\@glslink\glsdohyperlink
\let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*\gls{\@gls@hyp@opt\@gls}
\newcommand*\@gls}[2] [] {%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} []}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*\Gls{\@gls@hyp@opt\@Gls}
\newcommand*\@Gls}[2] [] {%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*\GLS{\@gls@hyp@opt\@GLS}

```

```

\newcommand*{\@GLS}[2] [] {%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \gls post link hook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] [] {%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \gls post link hook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2] [] {%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\glsifcaps\@secondofthree
\let\glsifcustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\newcommand*{\@GLSpl}[2][]{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
}
\def\@GLSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsifcaps\@thirdofthree
    \let\glsifcustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][]{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@firstofthree
    \def\glsifcustomtext{#3}%
    \def\glsinsert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else

```

```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link{#1}{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2][{}]{%
    \new@ifnextchar[{\@glsstext@{#1}{#2}}{\@glsstext@{#1}{#2}[]]}
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][{}]{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]]}
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glsstext}{\@gls@hyp@opt\@Glsstext}
\newcommand*{\@Glsstext}[2][{}]{%
    \new@ifnextchar[{\@Glsstext@{#1}{#2}}{\@Glsstext@{#1}{#2}[]]}
\def\@Glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][{}]{%
    \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]]}
\def\@glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][{}]{%
    \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]]}
\def\@Glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][{}]{%
    \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]]}
\def\@GLSfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
}

```

```

\newrobustcmd*\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*\@glsplural}[2] []{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*\@Glsplural}[2] []{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*\@GLSplural}[2] []{%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*\@glsfirstplural}[2] []{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*\@Glsfirstplural}[2] []{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*\@GLSfirstplural}[2] []{%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*\@Glsname}[2] []{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*\@GLSname}[2] []{%

```

```

    \new@ifnextchar [ {\@GLSname@{#1}{#2}} {\@GLSname@{#1}{#2} [] }
\def \@GLSname@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase {\glsentryname{#2}#3}} %
}
\newrobustcmd* {\glsdesc} {\@gls@hyp@opt \@glsdesc}
\newcommand* {\@glsdesc} [2] [] {%
    \new@ifnextchar [ {\@glsdesc@{#1}{#2}} {\@glsdesc@{#1}{#2} [] }
\def \@glsdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentrydesc{#2}#3} %
}
\newrobustcmd* {\Glsdesc} {\@gls@hyp@opt \@Glsdesc}
\newcommand* {\@Glsdesc} [2] [] {%
    \new@ifnextchar [ {\@Glsdesc@{#1}{#2}} {\@Glsdesc@{#1}{#2} [] }
\def \@Glsdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\Glsentrydesc{#2}#3} %
}
\newrobustcmd* {\GLSdesc} {\@gls@hyp@opt \@GLSdesc}
\newcommand* {\@GLSdesc} [2] [] {%
    \new@ifnextchar [ {\@GLSdesc@{#1}{#2}} {\@GLSdesc@{#1}{#2} [] }
\def \@GLSdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase {\glsentrydesc{#2}#3}} %
}
\newrobustcmd* {\glsdescplural} {\@gls@hyp@opt \@glsdescplural}
\newcommand* {\@glsdescplural} [2] [] {%
    \new@ifnextchar [ {\@glsdescplural@{#1}{#2}} {\@glsdescplural@{#1}{#2} [] }
\def \@glsdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentrydescplural{#2}#3} %
}
\newrobustcmd* {\Glsdescplural} {\@gls@hyp@opt \@Glsdescplural}
\newcommand* {\@Glsdescplural} [2] [] {%
    \new@ifnextchar [ {\@Glsdescplural@{#1}{#2}} {\@Glsdescplural@{#1}{#2} [] }
\def \@Glsdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\Glsentrydescplural{#2}#3} %
}
\newrobustcmd* {\GLSdescplural} {\@gls@hyp@opt \@GLSdescplural}
\newcommand* {\@GLSdescplural} [2] [] {%
    \new@ifnextchar [ {\@GLSdescplural@{#1}{#2}} {\@GLSdescplural@{#1}{#2} [] }
\def \@GLSdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase {\glsentrydescplural{#2}#3}} %
}
\newrobustcmd* {\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*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
    \new@ifnextchar [ {\@Glsymbolplural@{#1}{#2}} {\@Glsymbolplural@{#1}{#2} [] }
}
\def \@Glsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
    \new@ifnextchar [ {\@GLSsymbolplural@{#1}{#2}} {\@GLSsymbolplural@{#1}{#2} [] }
}
\def \@GLSsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
    \new@ifnextchar [ {\@glsuseri@{#1}{#2}} {\@glsuseri@{#1}{#2} [] }
}
\def \@glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
    \new@ifnextchar [ {\@Glsuseri@{#1}{#2}} {\@Glsuseri@{#1}{#2} [] }
}
\def \@Glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
    \new@ifnextchar [ {\@GLSuseri@{#1}{#2}} {\@GLSuseri@{#1}{#2} [] }
}
\def \@GLSuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
    \new@ifnextchar [ {\@glsuserii@{#1}{#2}} {\@glsuserii@{#1}{#2} [] }
}
\def \@glsuserii@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
}

```



```

\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\Glsuseriiii}{\@gls@hyp@opt\@Glsuseriiii}
\newcommand*{\@Glsuseriiii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriiii@{#1}{#2}}{\@Glsuseriiii@{#1}{#2} []}]
\def\@Glsuseriiii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriiii{#2}#3}%
}
\newrobustcmd*{\Glsuseriiiii}{\@gls@hyp@opt\@Glsuseriiiii}
\newcommand*{\@Glsuseriiiii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriiiii@{#1}{#2}}{\@Glsuseriiiii@{#1}{#2} []}]
\def\@Glsuseriiiii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriiiii{#2}#3}}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuserivv}{\@gls@hyp@opt\@Glsuserivv}
\newcommand*{\@Glsuserivv}[2] [] {%
  \new@ifnextchar[{\@Glsuserivv@{#1}{#2}}{\@Glsuserivv@{#1}{#2} []}]
\def\@Glsuserivv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserivv{#2}#3}%
}
\newrobustcmd*{\Glsuserivvi}{\@gls@hyp@opt\@Glsuserivvi}
\newcommand*{\@Glsuserivvi}[2] [] {%
  \new@ifnextchar[{\@Glsuserivvi@{#1}{#2}}{\@Glsuserivvi@{#1}{#2} []}]
\def\@Glsuserivvi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserivvi{#2}#3}}%
}
\newrobustcmd*{\Glsuserivvii}{\@gls@hyp@opt\@Glsuserivvii}
\newcommand*{\@Glsuserivvii}[2] [] {%

```

```

\new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\newcommand*{\@Glsuserv}[2] []{%
\new@ifnextchar [{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}]
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] []{%
\new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] []{%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\newcommand*{\@Glsuservi}[2] []{%
  \new@ifnextchar [{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}]
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\@ns@acrshort}[2] []{%
  \new@ifnextchar [{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls@scaps\@firstofthree
    \let\glsinsert\@empty
    \def\gls@customtext{%
      \acronymfont{\glsentryshort{#2}#3}%
    }%
    \@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
}

```

```

\glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@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\glsifcaps\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}]%
}
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo

```

```

\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] []{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glentryshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2] []{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```

```

}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\Acr long}{\@gls@hyp@opt\@ns@Acr long}
\newcommand*{\ns@Acr long}[2][{}]{%
  \new@if next char[{\@Acr long{#1}{#2}}{\@Acr long{#1}{#2}}[{}]}%
}
\def\@Acr long#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\ACR long}{\@gls@hyp@opt\@ns@ACR long}
\newcommand*{\ns@ACR long}[2][{}]{%
  \new@if next char[{\@ACR long{#1}{#2}}{\@ACR long{#1}{#2}}[{}]}%
}
\def\@ACR long#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
  }%
}

```

```

    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \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{%
      \glentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\AcrLongpl}{\@gls@hyp@opt\ns@AcrLongpl}
\newcommand*{\ns@AcrLongpl}[2][]{%
  \new@ifnextchar[{\@AcrLongpl{#1}{#2}}{\@AcrLongpl{#1}{#2}[]}%
}
\def\@AcrLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRLongpl}{\@gls@hyp@opt\ns@ACRLongpl}
\newcommand*{\ns@ACRLongpl}[2][]{%
  \new@ifnextchar[{\@ACRLongpl{#1}{#2}}{\@ACRLongpl{#1}{#2}[]}%
}
\def\@ACRLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```

```

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsifcaps\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ???
  }%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  }%
  \else

```

```

\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\xmakefirstuc{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newrobustcmd*\Glsentrydesc}[1]{%
\@Gls@entry@field{#1}{desc}%
}
\newcommand*\glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newrobustcmd*\Glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newcommand*\glsentrytext}[1]{\@Gls@entry@field{#1}{text}}
\newrobustcmd*\Glsentrytext}[1]{%
\@Gls@entry@field{#1}{text}%
}
\newcommand*\glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newrobustcmd*\Glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newcommand*\glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}
\newrobustcmd*\Glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}

```



```

}
\newcommand*\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newcommand*\glscapitalisewords}[1]{%
  \@capitalisewords{#1}%
}
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*\glsentrytitlecase}[2]{%
    \texorpdfstring
      {\@glsentrytitlecase{#1}{#2}}%
      {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
}

```

```

\newrobustcmd*\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
}
\newcommand*\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*\Glsentryshort}[1]{%
  \@Gls@entry@field{#1}{short}%
}
}
\newcommand*\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newrobustcmd*\Glsentryshortpl}[1]{%
  \@Gls@entry@field{#1}{shortpl}%
}
}
\newcommand*\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newrobustcmd*\Glsentrylong}[1]{%
  \@Gls@entry@field{#1}{long}%
}
}
\newcommand*\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newrobustcmd*\Glsentrylongpl}[1]{%
  \@Gls@entry@field{#1}{longpl}%
}
}
\newcommand*\glsentryfull}[1]{%

```

```

\acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
\acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
\acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
\acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\@gls@entry@field{#1}{numberlist}%
}%
}
\@ifpackageloaded{hyperref} {%
\newcommand*{\glsdisplaynumberlist}[1]{%
\GlossariesWarning
{%
\string\glsdisplaynumberlist\space
doesn't work with hyperref.^^JUsing
\string\glsentrynumberlist\space instead%
}%
\glsentrynumberlist{#1}%
}%
}%
{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\bgroup
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\let\@org@glsnumberformat\glsnumberformat
\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
\csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
\protected@edef\@gls@thislist{%
\@gls@thislist
\noexpand\@gls@numlist@sep
##1%
}%
\let\@gls@numlist@sep\@gls@numlist@nextsep

```

```

        \def\@gls@numlist@nextsep{\glsnumlistsep}%
        \@gls@donext@def
        \def\@gls@donext@def{%
            \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
        }%
    }%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
    \def\@glo@label{#2}%
    \@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][{%
    \@gls@adjustmode
    \glsdoifexists{#2}%
    {%
        \def\@glsnumberformat{glsnumberformat}%
        \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
        \setkeys{glossadd}{#1}%
        \@gls@saveentrycounter
        \@gls@setsort{#2}%
        \@do@wrglossary{#2}%
    }%
}
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
\newrobustcmd*{\glsaddall}[1][{%
    \protected@edef\@glo@type{\@glo@types}%
    \setkeys{glossadd}{#1}%
    \forallglsentries[\@glo@type]{\@glo@entry}{%
        \glsadd[#1]{\@glo@entry}%
    }%
}
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glo@entry}%
    {%
        \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
    }%
}
}
\newcommand*{\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{}}

```

```

\edef\glsclosebrace{\expandafter@gobble\string\}}
\edef\glsbackslash{\expandafter@gobble\string\}
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\%}
\edef\glsstildechar{\string~}
\ifglsexindy
  \newcommand*{\@glsfirstletter}{A}
\fi
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits@GlsSetXdyFirstLetterAfterDigits}
\ifglsexindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsexindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder@GlsSetXdyNumberGroupOrder
}
\ifglsexindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsexindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%

```

```

\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\@gls@addpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{{(require
\string"\@xdystyle.xdy\string")}}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
}%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
:sep "}"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
}
}
}

```

```

        ^^J\space\space\space
        :min-range-length \@glsminrange^^J%
    )
    }%
} %
} %
} %
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glsseeformat\string"
: close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
: open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{ }\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter@gobble\string\#2}%
}%
}%
}%
} %
} %
} %
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glspercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree}%
\write\glswrite{(markup-letter-group-list

```

```

:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string"}}%
\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffiF
\@onelevel@sanitize\gls@suffiFF
\ifx\gls@suffiF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffiF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffiFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffiFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string"}}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax

```



```

\ifundef{\glswrite}{\newwrite\glswrite}{\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string}
\write\glswrite{item_0 \string\string%\string\n\string}
\write\glswrite{item_1 \string\string%\string\n\string}
\write\glswrite{item_2 \string\string%\string\n\string}
\write\glswrite{item_01 \string\string%\string\n\string}
\write\glswrite{item_x1
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{item_12 \string\string%\string\n\string}
\write\glswrite{item_x2
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\}\string}
\write\glswrite{heading_suffix
  \string\string}\string\relax
  \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string\glsymbols\string}
\write\glswrite{numhead_positive \string\glsnumbers\string}
\write\glswrite{page_compositor \string\glscompositor\string}
\@gls@escbsdq\gls@suffixF

```

```

\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{ }
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{ %
  \IfTrackedLanguage{german}%
{ %
  \def\@gls@extramakeindexopts{-g}%
} %
} %
} %
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#####2#1#####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%

```

```

\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@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
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null}%
\noexpand\fi

```

```

        \noexpand\fi
        \noexpand\@gls@checkescquote
    }%
}
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
\newcommand*\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
    \ifglssavewrites
      \expandafter\newtoks\csname glo@#1@filetok\endcsname
    \else
      \expandafter\newwrite\csname glo@#1@file\endcsname
      \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%
    \PackageError{glossaries}%
    {Glossary type ‘#1’ not defined}%
    {New glossaries must be defined before using \string\makeglossaries}%
  }%
}
\newcommand*\@glsopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file
  \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*\@closegls}[1]{%
  \closeout\csname glo@#1@file\endcsname
}
\ifglxindy
\newcommand*\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
    {\let\@gls@order\@empty}%
  }

```

```

\ifcsundef{@xdy@#1@language}%
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@istfilebase\space
-C \@gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%

```

```

        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@istfilebase\space
-C \@gls@codepage\space
-t \jobname.\csuse{@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
\let\warn@noprntglossary\@makeglossaries@warn@noprntglossary
\ifgls@savenumberlist
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake

```



```

\renewcommand*{\@gls@doautomake}{%
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
{%
\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
\newcommand*\makenoidxglossaries{%
\@domakeglossaries
{%
\renewcommand{\@gls@noref@warn}[1]{%
\GlossariesWarning{Empty glossary for
\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%
\let\@gls@checkmkidxchars\@gobble
\glsesclocationsfalse
\let\@do@@wrglossary\gls@noidxglossary
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
\let\gls@checkseeallowed\relax
\renewcommand{\@do@seeglossary}[2]{%
\protected@edef\@gls@label{\glsdetoklabel{##1}}%
\protected@write\@auxout{}{%
\string\@gls@reference
{\csname glo@\@gls@label @type\endcsname}%
{\@gls@label}}%
{%
\string\glsseeformat##2}%
}
}

```

```

    }%
  }%
}%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*\@printgloss@setsort{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
  \string\makeglossaries}%
}%
\renewcommand*\@gls@entrynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*\@gls@displaynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
  }%

```

```

    }%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{
  ??\glsdoifexists{##1}%
  {
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
}%
\renewcommand*\@glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {
    ??\glsdoifexists{##1}%
    {
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\@glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\@glsnoidxnumberlistloophandler}[1]{%
  #1%
}

```

```

\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@nofwarn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\@gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\@gls@noidxdisplayloc
    {\@gls@counterprefix}%
    {\@gls@counter}%
    {\@gls@numberformat}%
    {\@gls@locref}%
    }%
  }%
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@gls@writefiles}{%
  \forallglossaries{\@gls@type}{%
    \ifcsundef{glo@\@gls@type @filetok}%
    {%
      \def\@gls@tmp{}%
    }%
    {%
      \edef\@gls@tmp{\expandafter\the
        \csname glo@\@gls@type @filetok\endcsname}%
    }%
    \ifx\@gls@tmp\@empty
      \ifx\@gls@type\@gls@defaulttype
        \GlossariesWarningNoLine{Glossary '\@gls@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '\@gls@type' has no
          entries}%
      \fi
    \else
      \@gls@openfile{\glswrite}{\@gls@type}%
      \immediate\write\glswrite{%

```

```

        \expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    \immediate\closeout\glswrite
\fi
}%
}
\newcommand*\@gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*\@gls@wrglossary}[2]{%
    \ifgls@savewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
                }%
            {%
                \ifignoredglossary{#1}{}%
                {%
                    \GlossariesWarning{No file defined for glossary ‘#1’}%
                }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*\@gls@writeentry}[2]{%
    \ifgls@indexonlyfirst
        \ifgls@used{#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@addprotectedpagefmt}[1]{%
  \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
  \csedef\gls#1page{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
  \protected@eappto\@wrglossarynumberhook{%
    \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
    \expandonce{\csname#1\endcsname}%
    \noexpand\def\expandonce{\csname#1\endcsname}{%
      \noexpand\@wrglossary@pageformat
      \expandonce{\csname gls#1page\endcsname}%
      \expandonce{\csname org@gls#1\endcsname}%
    }%
  }%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglsentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@wrglossary
}

```

```

\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theHglentrycounter\theglsentrycounter\relax
\def\@glo@counterprefix{}%
\else
\protected@edef\@glsHlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newcommand*{\@do@wrglossary}{%
\ifglswrallowprimitivemods
\expandafter\@glo@checkmkidxrangechar\@glsnumberformat\@nil

```

```

\def\@glo@range{}%
\expandafter\if\@glo@prefix(\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix)\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@gls@getcounterprefix[2]{%
\edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing~Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter~Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been~J
referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref

```



```

\ifglxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :tkey (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\glsseelastsep}{\space\andname\space}
\newcommand*\glsseesep}{, }
\newrobustcmd*\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*\gls@save@numberlist}[1]{%
  \ifglssavenumberlist

```

```

\toks@{#1}%
\protected@edef\do@writeaux@info{%
  \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\do@writeaux@info
\protected@write\auxout{}{\do@writeaux@info}%
\fi
}
\newcommand*\warn@noprntglossary{}{}%
\ifcsundef{printglossary}{}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*\printglossaries{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*\printnoidxglossaries{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*\@printgloss@setsort{}{}
\newcommand*\@gls@preglossaryhook{}{}
\newcommand*\printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle

```

```

\def@glossarystyle{%
  \ifx@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def@gls@dotoc@title{\gls@set@toc@title{\@glo@type}}%
\let@org@glossary@entry@numbers@glossary@entry@numbers
\bgroup
  \@print@gloss@set@sort
  \setkeys{print@gloss}{#1}%
  \@print@gloss@check@exists{\@glo@type}%
  {%
    \ifx@glossary@title@org@glossary@title
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
        \glossary@title
    \fi
    \let\current@glossary@\@glo@type
    \let@org@glossary@entry@numbers@glossary@entry@numbers
    \let@gls@non@next@pages@gls@non@next@pages
    \let@gls@next@pages@gls@next@pages
    \let\nopost@desc@\nopost@desc
    \gls@dotoc@title
    \@glossary@style
    \let@gls@org@glossary@entry@field@gloss@entry
    \let@gls@org@glossary@sub@entry@field@sub@gloss@entry
    \renewcommand{\gloss@entry}[1]{%
      \protected@xdef\gls@current@entry@label{\gls@detok@label{##1}}%
      \gls@org@glossary@entry@field{##1}%
    }%
    \renewcommand{\sub@gloss@entry}[2]{%
      \protected@xdef\gls@current@entry@label{\gls@detok@label{##2}}%
      \gls@org@glossary@sub@entry@field{##1}{##2}%
    }%
    \@gls@preglossary@hook
    #2%
  }%
\egroup
\global\let@glossary@entry@numbers@\org@glossary@entry@numbers
\global\let\warn@no@print@glossary\relax
}
\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%

```

```

}%
{\null}%
\ifglxindy
  \ifcsundef{@xdy@\@glo@type @language}%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
        }%
      }%
    }%
  }%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
            @language\endcsname}}%
        }%
      }%
    }%
  }%
  \@do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
      }%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
      }%
    }%
  }%
  \@do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \protected@edef\@glo@type{#2}%
  \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
  \csdef{\@glsref@#2}{%
    \for\@this@label:=\@glo@sortinglist\do{%
      \xifinlistcs{\@this@label}{\@glsref@#2}%
    }%
  }%
}

```

```

    {}%
    {%
      \listcsxadd{@glsref@#2}{\@this@label}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@label}%
    {%
      \glo@addchildren{#2}{\@this@label}%
    }%
    {}%
  }%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \glo@addchildren{#1}{\@this@childlabel}%
    }%
    {}%
  }%
}
\egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    {}%
    \expandafter\@glo@sortedinsert
      \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {}%
    {}%
    {}%
    \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
  }%
}
{}%
\@glo@sortedinsert{\@glo@sortinglist}{#1}%

```

```

}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
}

```

```

    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=word]}{}}%
    }%
  }
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\csuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
}

```

```

    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=nocase]}{}%
    }%
  }
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
      }%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
    \end{theglossary}%
  }%
}

```



```

    \glossarypostamble
  }%
  {%
    \@gls@noref@warn{\@glo@type}%
  }%
}
\def\glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%
    \@onelevel@sanitize\@gls@firsttok
    \expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
  }%
}
\def\@glo@grabfirst#1#2\@nil{%
  \ifdefempty\@glo@thislettergrp
  {%
    \def\@glo@thislettergrp{glssymbols}%
  }%
  {%
    \count@=\ucode‘#1\relax
    \ifnum\count@=0\relax
      \def\@glo@thislettergrp{glssymbols}%
    \else
      \ifdefstring\@glo@sorttype{case}%
      {%
        \count@=#1\relax
      }%
      {%
      }%
    \edef\@glo@thislettergrp{\the\count@}%
    \fi
  }%
}
\newcommand{\@gls@noidx@do}[1]{%
  \global\letcs{\@gls@loclist}{glo\@glsdetoklabel{#1}@loclist}%
  \ifglshasparent{#1}%
  {%
    \gls@level=\csuse{glo\@glsdetoklabel{#1}@level}\relax
    \ifdefvoid{\@gls@loclist}
    {%
      \subglossentry{\gls@level}{#1}{}%
    }%
    {%
      \subglossentry{\gls@level}{#1}%
    }%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}

```

```

    }%
  }%
}%
{%
\letcs{\@gls@sort}{glo@glsdetoklabel{#1}@sort}%
\expandafter\glo@grabfirst\@gls@sort{}{}\@nil
\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%
\ifdefempty{\@gls@currentlettergroup}{}%
{%
\global\let\@glo@thislettergrp\@glo@thislettergrp
\glsgroupskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%
\global\let\@gls@currentlettergroup\@glo@thislettergrp
\ifdefvoid{\@gls@loclist}
{%
\glossentry{#1}{}%
}%
{%
\glossentry{#1}%
{%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%
}
\newcommand*{\glsnoidxloclist}[1]{%
\def\@gls@noidxloclist@sep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
\@gls@noidxloclist@sep
#1%
\def\@gls@noidxloclist@sep{\delimN}%
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
}

```

```

    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{%
      \ifinlistcs{#2}{\@glsref@#1}%
      {}%
      {\listcsgadd{\@glsref@#1}{#2}}%
      \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
      {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
      {}%
      \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
    }%
  }
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
  \def\glossarytoctitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
  \ifcsundef{\@glsstyle@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \def\@glossarystyle{\setglossentrycompatibility
      \csname @glsstyle@#1\endcsname}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
  \ifcase\@gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar{*}%
    \renewcommand*\@@glossaryseclabel{}%
  \or

```

```

\renewcommand*\@@glossarysecstar{}%
\renewcommand*\@@glossaryseclabel{}%
\or
\renewcommand*\@@glossarysecstar{}%
\renewcommand*\@@glossaryseclabel{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar{*}%
\renewcommand*\@@glossaryseclabel{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*\@glo@no@assign@sortkey[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*\@glo@assign@sortkey[1]{%
\def\@glo@sorttype{#1}%
}
\newcommand*\@glsnonextpages{%
\gdef\glossaryentrynumbers##1{%
\glsresetentrylist
}%
}
\newcommand*\@glsnextpages{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}

```

```

\newcommand*\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*\glsnonextpages}{}
\newcommand*\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

```

\newcommand*\glsentryitem}[1]{%
  \ifglsentrycounter
    \glsstepentry{#1}\glsentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*\glsesubentryitem}[1]{%
  \ifglsesubentrycounter
    \glsstepsubentry{#1}\glsesubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheglossdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\providecommand*\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@glossentry
}
\newcommand*\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\newcommand*\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}

```

```

    }%
  }
  \newcommand*\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \newcommand*\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \providecommand*\compatiblesubglossentry}[3]{%
    \toks@{#3}%
    \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}}%
    {\noexpand\glsnamefont
     {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
    {\the\toks@}%
    }%
    \@do@subglossentry
  }
  \newcommand*\setglossentrycompatibility{%
    \let\glossentry\compatibleglossentry
    \let\subglossentry\compatiblesubglossentry
  }
  \setglossentrycompatibility
  \newcommand*\glossaryentryfield}[5]{%
    \GlossariesWarning
    {Deprecated use of \string\glossaryentryfield.^.^J
     I recommend you change to \string\glossentry.^.^J
     If you've just upgraded, try removing your gls auxiliary
     files^^J and recompile}%
    \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
  \newcommand*\glossarysubentryfield}[6]{%
    \GlossariesWarning
    {Deprecated use of \string\glossarysubentryfield.^.^J
     I recommend you change to \string\subglossentry.^.^J
     If you've just upgraded, try removing your gls auxiliary
     files^^J and recompile}%
  }

```

```

\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
  \dtl@ifsingle{#1}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
  {%
    \ifboolexpr{test{\ifstrequal{#1}{glsymbols}}
      or test{\ifstrequal{#1}{glsnumbers}}}%
    {%
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
    }%
    {%
      \def#2{#1}%
    }%
  }%
}
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\newcommand*{\glsgetgrouplabel}[1]{%
\ifthenelse{\equal{#1}{\glsymbolsgroupname}}{glsymbols}{%
\ifthenelse{\equal{#1}{\glsnumbersgroupname}}{glsnumbers}{#1}}%
\newcommand*{\setentrycounter}[2][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```



```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \@string\glossarystyle.^^J
    I recommend you switch to \@string\setglossarystyle\space unless
    you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\@glsnumber#1\nohyperpage{}}\@nil}
}

```

```

\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glshypernumber#3\@nil
  \fi
}
\def\@delimR#1\delimR #2\delimR #3\{%
\ifx\#2\%
  \@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\{%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
  \else
    \hyperlink{\glsentrycounter\@glo@counterprefix\the\toks@}%
      {\the\toks@}%
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glshypernumber{#1}}}

```

```

\newcommand*\hypersf}[1]{\textsf{\glshypernumber{#1}}}
\newcommand*\hypertt}[1]{\texttt{\glshypernumber{#1}}}
\newcommand*\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
\newcommand*\hypermd}[1]{\textmd{\glshypernumber{#1}}}
\newcommand*\hyperit}[1]{\textit{\glshypernumber{#1}}}
\newcommand*\hypersl}[1]{\textsl{\glshypernumber{#1}}}
\newcommand*\hyperup}[1]{\textup{\glshypernumber{#1}}}
\newcommand*\hypersc}[1]{\textsc{\glshypernumber{#1}}}
\newcommand*\hyperemph}[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }%
}
\newcommand{\newacronym}[4][[]]{
\newcommand*\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey}{short}
\newcommand*\glsshortpluralkey}{shortplural}
\newcommand*\glslongkey}{long}
\newcommand*\glslongpluralkey}{longplural}
\newrobustcmd*\acrfull}{\@gls@hyp@opt\@ns@acrfull}
\newcommand*\ns@acrfull[2][[]]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
    {\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace(#2)}
\newrobustcmd{\glspace}{\space}
\newrobustcmd*\Acrfull}{\@gls@hyp@opt\@ns@Acrfull}
\newcommand*\ns@Acrfull[2][[]]{%

```

```

\new@ifnextchar [{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2} []}%
}
\def\@Acrfull#1#2[#3]{%
    \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\Acrfullfmt[3]{%
    \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*\@ACRfull{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2] []{%
    \new@ifnextchar [{\@ACRfull{#1}{#2}}%
        {\@ACRfull{#1}{#2} []}%
}
}
\def\@ACRfull#1#2[#3]{%
    \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*\ACRfullfmt[3]{%
    \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfullpl{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2] []{%
    \new@ifnextchar [{\@acrfullpl{#1}{#2}}%
        {\@acrfullpl{#1}{#2} []}%
}
}
\def\@acrfullpl#1#2[#3]{%
    \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullplfmt[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*\@Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2] []{%
    \new@ifnextchar [{\@Acrfullpl{#1}{#2}}%
        {\@Acrfullpl{#1}{#2} []}%
}
}
\def\@Acrfullpl#1#2[#3]{%
    \ACRfullplfmt{#1}{#2}{#3}%
}
}

```

```

\newcommand*\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acrentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
\protected@edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\expandonce{\acronymentry{##2}}},%
    sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
    text={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    \GenericAcronymFields,%
    \the\glskeylisttok
  }%
}%
\@do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%

```

```

\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields{description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*\acronymsort}[2]{##1}
\newcommand*\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{##1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
    }%
  }%
}
\newcommand*\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{##2}%
    \csdef{@glsacr@styledefs@#1}{##3}%
  }%
}
\newcommand*\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{##2}%
    \csdef{@glsacr@styledefs@#1}{##3}%
  }%
  {%

```

```

    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{%
  }%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\glsacsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\glsacsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\glsacsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%

```

```

}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglsashaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*\acronymfont}[1]{\textsc{##1}}%

```



```

    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  }
  \newacronymstyle{long-sm-short}%
  {%
    \GlsUseAcrEntryDispStyle{long-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-short}%
    \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
  }
  \newacronymstyle{sc-short-long}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  }
  \newacronymstyle{sm-short-long}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
  }
  \newacronymstyle{long-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
  \newacronymstyle{long-sp-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-sp-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sp-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
  }

```

```

}
\newacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{

```

```

\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylong{##2}##3\space

```

```

        (\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
    (\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glsHYperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \firstacronymfont{\Glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
    (\glsentrylong{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%

```

```

}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
}

```

```

\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\acrlong
  \let\aclp\acrlongpl
  \let\Acl\Acrlong
  \let\Aclp\Acrlongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\newcommand*{\@gl@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@the\gl@labeltok @plural\endcsname\@gl@@default@acr@plural
    \csxdef{glo@the\gl@labeltok @plural}{\expandonce\@glo@shortpl}%
  \fi
  \expandafter\ifx\csname glo@the\gl@labeltok @firstpl\endcsname\@gl@@default@acr@firstplural
    \csxdef{glo@the\gl@labeltok @firstpl}{%
      \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
    \fi
  \expandafter\ifx\csname glo@the\gl@labeltok @descplural\endcsname\@gl@@default@acr@descplural
    \csxdef{glo@the\gl@labeltok @descplural}{\expandonce\@glo@longpl}%
    \fi
}
\newcommand*{\@gl@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@gl@@default@acr@firstplural}{%
  \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
}
\newcommand*{\@gl@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
  \@for\@gl@type:=\@gl@sacronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gl@type}%
  }%
}
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@gl@sacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDefaultAcronymDisplayStyle{\@glo@type}%
  \fi
  \gl@keylisttok{##1}%
  \gl@labeltok{##2}%
  \gl@shorttok{##3}%
  \gl@longtok{##4}%
  \newacronymhook

```



```

\DefaultNewAcronymDef
}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
\footnote{\glslink[#1]{#2}{#3}}%
}
\newcommand*{\acrnolinkfootnote}[3]{%
\footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\acronymfont{\glsgenentryfmt}%
}%
{%
\firstacronymfont{\glsgenentryfmt}%
\ifgls hassymbol{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
{%
\glsifplural
{\glsentrysymbolplural{\glslabel}}%
{\glsentrysymbol{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
}
}
}

```

```

    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    symbol={\the\glslongtok},%
    symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{footnote,smallcaps,description}%
    {footnote-sc-desc}{short-sc-footnote-desc}%
  \else
    \ifglsacrsmaller
      \@gls@deprecated@acr@warning{footnote,smaller,description}%
      {footnote-sm-desc}{short-sm-footnote-desc}%
    \else
      \@gls@deprecated@acr@warning{footnote,description}%
      {footnote-desc}{short-footnote-desc}%
    \fi
  \fi
\renewcommand{\newacronym}[4][]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \DescriptionFootnoteNewAcronymDef
}%

```

```

\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\newcommand*\SetDescriptionDUAAcronymDisplayStyle[1]{%
  \defglsentryfmt[##1]{\glsgenentryfmt}%
}
\newcommand*\DescriptionDUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}

```

```

}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\SetDescriptionDUAAcronymStyle{%
\ifglsacrsmallcaps
\PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
\ifglsacrsmaller
\PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
\@gls@deprecated@acr@warning{dua,description}{dua-desc}{long-noshort-desc}%
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*\SetDescriptionAcronymDisplayStyle[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsgenentryfmt}\gls@org@insert
}%
}%
\glsgenentryfmt
\ifglshassymbol{\glslabel}%
{%
\glsifplural

```

```

        {%
        \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
        }%
        {%
        \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
        }%
        \space(\protect\firstacronymfont
        {\gls caps case
        {\@glo@symbol}
        {\@glo@symbol}
        {\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
    }%
    }%
    {\gls custom text\gls insert}%
}
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\gls shorttok}{\the\gls longtok}},%
sort={\the\gls shorttok},%
first={\the\gls longtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\gls shorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\gls shorttok},%
shortplural={\the\gls shorttok\noexpand\acrpluralsuffix},%
long={\the\gls longtok},%
longplural={\the\gls longtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\gls keylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%

```

```

\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
\ifglsacrsmallcaps
\@gls@deprecated@acr@warning{smallcaps,description}%
{long-sc-short-desc}{long-short-sc-desc}%
\else
\ifglsacrsmaller
\@gls@deprecated@acr@warning{smaller,description}%
{long-sm-short-desc}{long-short-sm-desc}%
\else
\@gls@deprecated@acr@warning{description}%
{long-short-desc}{long-short-desc}%
\fi
\fi
\renewcommand{\newacronym}[4][ ]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand{\acronymfont}[1]{\textsc{##1}}
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
\ifglsacrsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty

```

```

\ifglsused{\glslabel}%
{%
  \acronymfont{\glsgetentryfmt}\gls@org@insert
}%
{%
  \firstacronymfont{\glsgetentryfmt}\gls@org@insert
  \ifglsashaslong{\glslabel}%
  {%
    \expandafter\protect\expandafter\acrfootnote\expandafter
    {\@gls@link@opts}{\@gls@link@label}%
    {%
      \glsifplural
        {\glsentrylongpl{\glslabel}}%
        {\glsentrylong{\glslabel}}%
    }%
  }%
  }%
  {}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\the\glslongtok},%
    descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}

```

```

}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*\SetFootnoteAcronymStyle{%
  \ifglsacrsmallcaps
    \gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
  \else
    \ifglsacrsmaller
      \gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
    \else
      \gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
    \fi
  \fi
\renewcommand{\newacronym}[4] []{%
  \ifx\glsacronymlists\@empty
    \def\glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\glo@type}%
    \SetFootnoteAcronymDisplayStyle{\glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \gslongtok{##4}%
  \newacronymhook
  \FootnoteNewAcronymDef
}%
\@for\gls@type:=\glsacronymlists\do{%
  \SetFootnoteAcronymDisplayStyle{\gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
  can’t both be set}{}%
\fi
}%
\DeclareRobustCommand*\glsdoparenifnotempty}[2]{%

```



```

\protected@edef\gls@tmp{#1}%
\ifdefempty\gls@tmp
{}%
{%
  \ifx\gls@tmp\@gls@default@value
  \else
  \space (#2{#1})%
  \fi
}%
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\gls@customtext
{%
  \let\gls@org@insert\gls@insert
  \let\gls@insert\@empty
  \ifglsused{\gls@label}%
  {%
    \acronymfont{\gls@genentryfmt}\gls@org@insert
  }%
  {%
    \gls@genentryfmt
    \ifgls@shassymbol{\gls@label}%
    {%
      \gls@ifplural
      {%
        \def\@glo@symbol{\gls@entrysymbolplural{\gls@label}}%
      }%
      {%
        \def\@glo@symbol{\gls@entrysymbol{\gls@label}}%
      }%
      \space
      (\gls@scapscase
      {\firstacronymfont{\@glo@symbol}}%
      {\firstacronymfont{\@glo@symbol}}%
      {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
    }%
    {}%
  }%
  }%
  {\gls@customtext\gls@insert}%
}%
}
\newcommand*{\SmallNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\gls@labeltok}%
{%
  type=\acronymtype,%
  name={\noexpand\acronymfont{\the\gls@shorttok}},%
  sort={\the\gls@shorttok},%

```

```

text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
  \ifglsacrsmallcaps
    \gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
  \else
    \gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
  \fi
\renewcommand{\newacronym}[4][]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi
}

```

```

\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont}[1]{\textsc{##1}}
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \renewcommand*\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
}%

```

```

\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \@gls@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDUADisplayStyle{\@gls@type}%
  }%
}
\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle

```

```

\ifglsacrdescription
  \ifglsacrfootnote
    \SetDescriptionFootnoteAcronymStyle
  \else
    \ifglsacrdua
      \SetDescriptionDUAAcronymStyle
    \else
      \SetDescriptionAcronymStyle
    \fi
  \fi
\else
  \ifglsacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glsacrsmallcaps}\OR
      \boolean{glsacrsmaller}}{%
      \SetSmallAcronymStyle
    }%
    \ifglsacrdua
      \SetDUASyle
    \fi
  }%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    }
  }
}

```

```

        user1={\the\glsshorttok},%
        user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
        user3={\the\glslongtok},%
        user4={\the\glslongtok\noexpand\acrpluralsuffix},%
        \CustomAcronymFields,%
        \the\glskeylisttok
    }%
}%
\do@newglossaryentry
}
\newcommand*{\SetCustomStyle}{%
\renewcommand{\newacronym}[4][{}]{%
\ifx\@glsacronymlists\@empty
\def\@gls@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@gls@type}%
\SetCustomDisplayStyle{\@gls@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\CustomNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetCustomDisplayStyle{\@gls@type}%
}%
}
\RequirePackage{glossary-hypernav}[=v4.49]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
}

```

```

\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showgloftype}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@ftype\endcsname
}
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouser}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@user\endcsname
}
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showglongame}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@short\endcsname
}

```

```

}
\newcommand*\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*\showglooclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@oclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \csshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*\showglossaries}{%
  \show\@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.49]
\fi

```

8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*\PassOptionsToPackage{\CurrentOption}{glossaries}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}

```



```

{%
\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The 'glossaries-accsupp'
package has been loaded\MessageBreak
after the 'glossaries-extra' package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the 'accsupp' option when you load\MessageBreak
'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
before loading 'glossaries-extra'}%
\fi
}
{}
\def\compatibleglossentry#1#2{%
\toks@{#2}%
\protected@edef\@do@glossentry{%
\noexpand\accsuppglossaryentryfield{#1}%
{ \noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\toks@{#3}%
\protected@edef\@do@subglossentry{%
\noexpand\accsuppglossarysubentryfield{\number#1}%
{#2}%
{ \noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@subglossentry
}
\RequirePackage{glossaries}[v4.49]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}
}
}
{}
\newcommand{\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 @useriaccess\endcsname{%
  \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriiiaccess\endcsname{%
  \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userivaccess\endcsname{%
  \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}
\newcommand*\@glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*\@glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*\@glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*\@glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*\@glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*\@glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*\@glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*\@glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*\@glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*\@glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*\@glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*\@glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

```

}
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}

```

```

\newcommand*\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
\newcommand*\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*\@glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@access}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
\newrobustcmd*\@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*{\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{glo@\glsdetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{%
    }%
    {%
      \csname gls#1accessdisplay\endcsname{#2}{#3}%
    }%
  }
}
\renewcommand*{\@@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glsymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}
          {\glsinsert}%
        }%
      }%
    }%
    #1{\glsfirstpluralaccessdisplay
      {\glsentryfirstplural{\glslabel}}{\glslabel}}%
      {\glsdescriptionpluralaccessdisplay
        {\glsentrydescplural{\glslabel}}{\glslabel}}%
      {\glsymbolpluralaccessdisplay
        {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
      {\glsinsert}%
    }%
  }%
}

```

```

{%
  \ifglused\glslabel
  {%
    #2{\glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
{%
  \ifglused\glslabel
  {%
    \MakeUppercase{%
      #2{\glspluralaccessdisplay
        {\glsentryplural{\glslabel}}{\glslabel}}%
        {\glsdescriptionpluralaccessdisplay
          {\glsentrydescplural{\glslabel}}{\glslabel}}%
        {\glssymbolpluralaccessdisplay
          {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
    }%
    {%
      \MakeUppercase{%
        #1{\glsfirstpluralaccessdisplay
          {\glsentryfirstplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glssymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
          {\glsinsert}}%
        }%
      }%
    }%
  }%
  \glscapscale
  {%
    \ifglused\glslabel
    {%

```



```

}%
{%
  \MakeUppercase{%
    #1{\glsfirstaccessdisplay
      {\glsentryfirst{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glsymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
  }%
}%
}%
}%
{%
  \ifglsused{\glslabel}%
  {%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
}
\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glsfirstpluralaccessdisplay
            {\glsentryfirstplural{\glslabel}}{\glslabel}%

```

```

    \glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}%
    \glsinsert
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}%
    \glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
    \glsinsert
  }%
}

```

```

    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay
      {\mfirstucMakeUppercase{\glsentrytext{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
    {%
      \glsfirstaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirst{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}
}
}
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\Glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \mfirstucMakeUppercase
          {\acronymfont
            {\glsshortpluralaccessdisplay
              {\glsentryshortpl{\glslabel}}{\glslabel}}%
          }%
        }%
      }%
    }%
  }%
}

```

```

        \glsinsert}%
    }%
}%
{%
    \glscapscase
    {%
        \acronymfont
        {\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \acronymfont
        {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \mfirstucMakeUppercase
        {\acronymfont{%
            \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
            \glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \mfirstucMakeUppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
{%
    \glscapscase
    {%
        \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \mfirstucMakeUppercase
        {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
}

```

```

    }%
  }%
}%
{%
  \glscustomtext
}%
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
   {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
   {\protect\firstacronymfont{\Glsentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glsaps@case\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
  }%
}

```



```

}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay
{\MakeUppercase{\glstryshort{#2}}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\glstrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax

```

```

\let\glsifplural\@secondoftwo
\let\glsifcaps\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glslongaccessdisplay{%
    \MakeUppercase{\glsentrylong{#2}}}{#2}#3}%
  }%
\@gls@link[#1]{#2}{\csname gls@#1\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\@GLS@text@#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\@GLS@first@#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\@GLS@plural@#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\@glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsymbolplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glstentrysymbolplural{#2}}{#2}#3}%
  }
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glstentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glstentrysymbolplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glstentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glstentryuseri{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glstentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@ii#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glstentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glstentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@iii#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glstentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glstentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@iv#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

```

```

\renewcommand*\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}
\newcommand*\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
}

```

```

\renewcommand*\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%
  \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{\Glsentryshort{##1}}}{##1}##2\space
  (\glsLongAccessDisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*\GenPlacrFullFormat}[2]{%
  \glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glsLongPluralAccessDisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*\GenPlacrFullFormat}[2]{%
  \glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
  (\glsLongPluralAccessDisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%

```

```

        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
    }
\renewacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }

```



```

}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
          {\mfirstucMakeUppercase{\glsinsert}}%
      }%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \glsgenentryfmt
  }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%

```

```

\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\Glsentryfullpl}[1]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}%
\renewcommand*\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}

```

```

}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}{##2}}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}{##2}}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshort{##2}}{##2}}##3\space
        (\glslongaccessdisplay{\glentrylong{##2}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}{##2}}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}{##2}}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshortpl{##2}}{##2}}##3\space
        (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}}%
\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}{##1}}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}{##1}}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%

```

```

}
\renewacronymstyle{footnote-sc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
  }
  \newcommand*\glsdefaultshortaccess}[2]{#1}
  \renewcommand*\newacronymhook{%
    \protected@edef\@gls@keylist{%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      \the\glskeylisttok}%
    \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
  }
  \renewcommand*\DefaultNewAcronymDef{%
    \edef\@do@newglossaryentry{%
      \noexpandnewglossaryentry{\the\glslabeltok}%
      {%
        type=\acronymtype,%
        name={\the\glsshorttok},%
        description={\the\glslongtok},%
        descriptionaccess=\relax,%
        text={\the\glsshorttok},%
        access={\noexpand\@glo@textaccess},%
        sort={\the\glsshorttok},%
        short={\the\glsshorttok},%
        shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
        shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
        long={\the\glslongtok},%
        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        first={\noexpand\glslongaccessdisplay
          {\the\glslongtok}{\the\glslabeltok}\space
          (\noexpand\glsshortaccessdisplay
            {\the\glsshorttok}{\the\glslabeltok})},%
        plural={\the\glsshorttok\acrpluralsuffix},%
        firstplural={\noexpand\glslongpluralaccessdisplay
          {\noexpand\@glo@longpl}{\the\glslabeltok}\space
          (\noexpand\glsshortpluralaccessdisplay
            {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
        firstaccess=\relax,%
        firstpluralaccess=\relax,%
        textaccess={\noexpand\@glo@shortaccess},%
        \the\glskeylisttok
      }%
    }%
  }
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\gl@labeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\gl@shorttok}},%
    sort={\the\gl@shorttok},%
    text={\the\gl@shorttok},%
    short={\the\gl@shorttok},%
    shortplural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
    shortaccess={\gl@defaultshortaccess{\the\gl@longtok}{\the\gl@shorttok}},%
    long={\the\gl@longtok},%
    longplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
    access={\noexpand\@glo@textaccess},%
    plural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
    symbol={\the\gl@longtok},%
    symbolplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
    firstpluralaccess=\relax,
    textaccess={\noexpand\@glo@shortaccess},%
    \the\gl@keylisttok
  }%
}%
\let\org@gl@assign@firstpl\gl@assign@firstpl
\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}

```

```

}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}

```



```

\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
access={\noexpand\@glo@textaccess},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
\@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\renewcommand*{\SmallNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
access={\noexpand\@glo@symbolaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%

```

```

text={\noexpand\@glo@short},%
textaccess={\noexpand\@glo@shortaccess},%
plural={\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\@glo@firstplural},%
symbol={\the\glsshorttok},%
symbolaccess={\the\glslongtok},%
symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglongnameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglotextaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglopluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%

```

```

\expandafter\show\csname glo@glstetoklabel{#1}@firstaccess\endcsname
}
\newcommand*\showglofirstpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*\showglosymbolaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*\showglosymbolpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolpluralaccess\endcsname
}
\newcommand*\showglodescaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@descaccess\endcsname
}
\newcommand*\showglodescpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*\showgloshortaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@shortaccess\endcsname
}
\newcommand*\showgloshortpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*\showglolongaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@longaccess\endcsname
}
\newcommand*\showglolongpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@longpluralaccess\endcsname
}
}

```

8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
}%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak

```

```

        Language modules need to be installed separately.\MessageBreak
        Please check on CTAN for a bundle called\MessageBreak
        ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
} %
} %
{} %

```

8.21 Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2021/11/01 v4.49 (NLCT)]
\ifglxsindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J}%
    (markup-locref
    :open \string"\string~n\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter@gobble\string\{\string" ^^J
    :close \string"\expandafter@gobble\string\}\string" ^^J
    :attr \string"#1\string")}}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}

```

```

\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxindy
  \renewcommand*{\GlsAddXdyLocation}[2]{%
    \edef\xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
    }%
    \edef\xdyuserlocationnames{%
      \@xdyuserlocationnames^^J\space\space\space
      \string"#1\string"%
    }
  }
\fi
\renewcommand{\do@wrglossary}[1]{%
\ifglxindy
  \expandafter\@glo@check@mkidxrangear\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix)\relax
    \def\@glo@range{:close-range}%
  \fi
  \fi
  \protected@edef\@glslocref{\theglentrycounter}%
  \@gls@checkmkidchars\@glslocref
  \glossary[\csname glo@#1@type\endcsname]{%
    (indexentry :tkey (\csname glo@#1@index\endcsname)
      :locref \string"\@glslocref\string" %
      :attr \string"\@glo@suffix\string" \@glo@range
    )
  }%
\else
  \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
  \glossary[\csname glo@#1@type\endcsname]{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar\@glo@numfmt}{\theglentrycounter}}%
  \fi
}
\def\@set@glo@numformat#1#2#3{%
  \expandafter\@glo@check@mkidxrangear#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[]#{#2}%
  \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidchars#1%
}

```

```

\ifglxindy
\def\writeist{%
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package in compatible-2.07 mode}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
      \protected@write\glswrite{{(require
        \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
  \@onelevel@sanitize\@gls@roman
  \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}%
  \@onelevel@sanitize\@tmp
  \ifx\@tmp\@gls@roman
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (\string"roman-numbers-lowercase\string")
      :min-range-length \@glsminrange)}%
  \else
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (:sep "\@gls@roman")
      :min-range-length \@glsminrange)}%
  \fi
  \write\glswrite{(define-location-class
    \string"Roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-uppercase\string")
    :min-range-length \@glsminrange)}%
  \write\glswrite{(define-location-class
    \string"arabic-page-numbers\string"^^J\space\space\space
    (\string"arabic-numbers\string")
    :min-range-length \@glsminrange)}%
  \write\glswrite{(define-location-class
    \string"alpha-page-numbers\string"^^J\space\space\space
    (\string"alpha\string")
    :min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter@gobble
  \string%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list

```

```

:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}

```



```

\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2021/11/01 v4.49 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
  \ifcsundef{@glscompstyle@#1}%
  {%
    \csdef{@glscompstyle@#1}{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
  }%
}
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
    \ifglshaschildren{##1}%
    {%
      \glsresetsubentrycounter
      \glsinlineparentchildseparator
      \def\gls@inlinesubsep{}%
      \def\gls@inlinepostchild{\glsinlinepostchild}%
    }%
    {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]

```

```

        ##3\glspostdescription\space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
  \glsentryitem{##2}%
  \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
    \mbox{}\par\nobreak\@afterheading
    ##3\glspostdescription\space ##5}%
  \renewcommand*\glossarysubentryfield}[6]{%
    \par
    \glsentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##1}\glstarget{##1}{##2}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
  \renewcommand*\glossarysubentryfield}[6]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##2}%
      \glstarget{##2}{##3}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
}%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscompstyle@listdotted}%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glsentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}

```

```

}%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \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]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
    \tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
    \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%

```

```

\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\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}{##2}%
    \ifx\relax##4\relax
      \else
        \space{##4}%
      \fi
    \fi
  }%

```

```

\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else

```

```

        \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
        \hangindent\glstreeindent
        \parindent\glstreeindent
    \fi
    \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
        \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
    \else
        (##4)\space
    \fi
    ##3\glspostdescription \space ##5\par
    \def\@gls@prevlevel{0}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
    \ifnum##1=1\relax
        \glssubentryitem{##2}%
    \fi
    \ifnum\@gls@prevlevel=##1\relax
    \else
        \@ifundefined{@glswidestname\romannumeral##1}{%
            \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
            \settowidth{\gls@tmplen}{\textbf{%
                \csname @glswidestname\romannumeral##1\endcsname\space}}}%
        \ifnum\@gls@prevlevel<##1\relax
            \setlength\glstreeindent\gls@tmplen
            \addtolength\glstreeindent\parindent
            \parindent\glstreeindent
        \else
            \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
                \settowidth{\glstreeindent}{\textbf{%
                    \@glswidestname\space}}}%
                \settowidth{\glstreeindent}{\textbf{%
                    \csname @glswidestname\romannumeral\@gls@prevlevel
                    \endcsname\space}}}%
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
        \fi
    \fi
    \hangindent\glstreeindent
    \makebox[0pt][r]{\makebox[\gls@tmplen][l]{%
        \textbf{\glstarget{##2}{##3}}}%
    \ifx##5\relax\relax
    \else
        (##5)\space
    \fi
    ##4\glspostdescription\space ##6\par
    \def\@gls@prevlevel{##1}%
}%
}%
\compatglossarystyle{almtreeegroup}{%

```



```

\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{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{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{super3col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%

```

```

\csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
\csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
\csuse{@glscompstyle@super4col}%
}%

```

8.23 Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
{%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
}%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-\CurrentTrackedLanguage’ or similar}%
}%
}%
}%
{}%

```

8.24 Rollback v4.49 (glossaries-prefix-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2021/11/01 v4.49 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[=v4.49]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%

```

```

{prefixfirst}{prefixfirst},%
{prefixfirstplural}{prefixfirstplural},%
{prefix}{prefix},%
{prefixplural}{prefixplural}%
}
\appto@newglossaryentryprehook{%
\def@glo@entryprefix{}%
\def@glo@entryprefixplural{}%
\let@glo@entryprefixfirst@gls@default@value
\let@glo@entryprefixfirstplural@gls@default@value
}
\appto@newglossaryentryposthook{%
\gls@assign@field{}{\@glo@label}{prefix}{\@glo@entryprefix}%
\gls@assign@field{}{\@glo@label}{prefixplural}{\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefix\endcsname}{\@glo@label}{prefixfirst}%
{\@glo@entryprefixfirst}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefixplural\endcsname}{\@glo@label}%
{prefixfirstplural}{\@glo@entryprefixfirstplural}%
}
\newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@glsdetoklabel{#1}@prefixfirst}}
\newcommand*{\glsentryprefixfirstplural}[1]{%
\csuse{glo@glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*{\glsentryprefix}[1]{\csuse{glo@glsdetoklabel{#1}@prefix}}
\newcommand*{\glsentryprefixplural}[1]{\csuse{glo@glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*{\Glsentryprefixfirst}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc@glo@text
}
\newrobustcmd*{\Glsentryprefixfirstplural}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc@glo@text
}
\newrobustcmd*{\Glsentryprefix}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc@glo@text
}
\newrobustcmd*{\Glsentryprefixplural}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc@glo@text
}
\newcommand*{\ifglshasprefix}[3]{%
\ifcsemtty{glo@glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
\ifcsemtty{glo@glsdetoklabel{#1}@prefixplural}%
{#3}%
}

```

```

    {#2}%
}
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcempty{glo@glstdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@glstdetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsextr@record
  {\@glsextr@record{#1}{#2}{glslink}}%
  {}%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
  {\@pgls@{#1}{#2}}%
  {\@pgls@{#1}{#2}[]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{%
    }%
    }%
    \ifglshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{%
    }%
    \@glss@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@glshyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
  {\@pglsp1@{#1}{#2}}%
  {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%

```

```

        \ifglshasprefixplural{#2}{\glstentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
            {\glstentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2}[#3]%
}%
}
\newrobustcmd{\PglS}{\@gls@hyp@opt\PglS}
\newcommand*{\@PglS}[2][{}]{%
    \new@ifnextchar[%
        {\@PglS@{#1}{#2}}%
        {\@PglS@{#1}{#2}[{}]}%
    }
\def\@PglS@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%
        \ifglsused{#2}%
        {%
            \ifglshasprefix{#2}%
            {%
                \Glsentryprefix{#2}%
                \glsprefixsep
                \@gls@{#1}{#2}[#3]%
            }%
            {\@Gls@{#1}{#2}[#3]}%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {%
                \Glsentryprefixfirst{#2}%
                \glsprefixsep
                \@gls@{#1}{#2}[#3]%
            }%
            {\@Gls@{#1}{#2}[#3]}%
        }%
    }%
}
\newrobustcmd{\PglSpl}{\@gls@hyp@opt\PglSpl}
\newcommand*{\@PglSpl}[2][{}]{%
    \new@ifnextchar[%
        {\@PglSpl@{#1}{#2}}%
        {\@PglSpl@{#1}{#2}[{}]}%
    }
\def\@PglSpl@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%

```

```

\ifglused{#2}%
{%
  \ifglshasprefixplural{#2}%
  {%
    \Glentryprefixplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glsp1@{#1}{#2}[#3]}%
}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glsp1@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@gls@hyp@opt\PGLS}
\newcommand*{\@PGLS}[2][\%
  \new@ifnextchar[\%
  {\@PGLS@{#1}{#2}}%
  {\@PGLS@{#1}{#2}[]}%
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glentryprefix{#2}\glsprefixsep}}}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glentryprefixfirst{#2}\glsprefixsep}}}%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
}
\newrobustcmd{\PGLSp1}{\@gls@hyp@opt\PGLSp1}
\newcommand*{\@PGLSp1}[2][\%
  \new@ifnextchar[\%
  {\@PGLSp1@{#1}{#2}}%
  {\@PGLSp1@{#1}{#2}[]}%
}
\def\@PGLSp1@#1#2[#3]{%

```



```

\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
  \ifglsused{#2}%
  {%
    \ifglsshasprefixplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%
  }%
  {%
    \ifglsshasprefixfirstplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{%
  }%
  \@GLSp1@{#1}{#2}{#3}%
}%
}

```

8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2021/11/01 v4.49 (NLCT)]
\newcommand*\glsnavhyperlink}[3][\@glo@type]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*\glsnavhypertarget}[3][\@glo@type]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{\string\@gls@hypergroup{#1}{#2}}%
  \@gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
  \expandafter\@gls@list\csname @gls@hypergroup@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}}%
  \if@endfor
  \else
  \GlossariesWarningNoLine{Navigation panel
    for glossary type ‘#1’^^Jmissing group ‘#2’}%
  \gdef\gls@hypergroup@rerun{%
    \GlossariesWarningNoLine{Navigation panel
      has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
\newcommand*\@gls@hypergroup}[2]{%
\@ifundefined{\@gls@hypergroup@list@#1}{%
  \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
}{%

```

```

\expandafter\let\expandafter\@gls@tmp
  \csname @gls@hypergroupplist@#1\endcsname
\expandafter\xdef\csname @gls@hypergroupplist@#1\endcsname{%
  \@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergroupplist@\@glo@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergroupplist@\@glo@type\endcsname
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glsymbolnav}{%
  \glsnavhyperlink{glsymbols}{\glsgetgrouptitle{glsymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}

```

8.26 Rollback v4.49 (glossary-inline-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2021/11/01 v4.49 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{}%
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{}%
  }%
  {\glspostinline}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgrupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \glsentryitem{##1}%
  }%
}

```

```

\glsinlinenameformat{##1}{%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
}%
{%
  \ifglsdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\ifglschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \gls@inlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\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*\glslistsubglossentry}[3]{%
    \glsentryitem{##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\hspace}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*\glossentry}[2]{%

```

```

    \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]{%
}

```

8.28 Rollback v4.49 (glossary-long-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}[lp{glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}[|lp{glsdescwidth}|]{\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longheader}{%
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
  }

```

```

\newglossarystyle{longheaderborder}{%
  \setglossarystyle{longborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{long3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

```

}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
}

```



```

\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
}

```

8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%

```

```

        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
\ifglsnogroupskip
  \renewcommand*\{glsgroupskip}\{}%
\else
  \renewcommand*\{glsgroupskip}\{glspenaltygroupskip}%
\fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*\{glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsnogroupskip
    \renewcommand*\{glsgroupskip}\{}%
  \else
    \renewcommand*\{glsgroupskip}\{glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\lp{\glsdescwidth}\lp{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glsdescwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%

```

```

\glspatchLToutput
\setglossarystyle{altlong4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
\ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
\noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
\renewcommand*{\LT@output}{%
\ifnum\outputpenalty <-\@Mi
\ifnum\outputpenalty > -\LT@end@pen
\LT@err{floats and marginpars not allowed in a longtable}\@ehc
\else
\setbox\z@\vbox{\unvbox\@cclv}%
\ifdim \ht\LT@lastfoot>\ht\LT@foot
\dimen@\pagegoal
\advance\dimen@-\ht\LT@lastfoot
\ifdim\dimen@<\ht\z@
\setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
\@makecol
\@outputpage
\setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
\fi
\fi
\global\@colroom\@colht
\global\vsizel\@colht
{\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
\fi
\else
\setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
\@makecol
\@outputpage
\global\vsizel\@colroom
\copy\LT@head
\glsLTpenaltycheck
\nobreak
\fi
}%
}

```

8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\ProvidesPackage{glossary-longragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{longragged}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}[1>{\raggedright}p{glsdescwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%}
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}%
    \glspostdescription\space ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{glsdescwidth}|}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
}
```

```

}
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline\endhead
        \hline\endfoot}%
    }
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l]>\raggedright}p{\glsdescwidth}l%
    >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l]>\raggedright}p{\glsdescwidth}l%
    >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>\raggedright}p{\glsdescwidth}l|l%
    >\raggedright}p{\glspagelistwidth}l}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}

```

8.31 Rollback v4.49 (glossary-mcols-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.49]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}{%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\end{multicols}}%
  }
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
  }
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
  }
\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%

```

```

\renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
    \let\item\glstreeitem}%
  {\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreespannav}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
}

```



```

\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%

```

```

\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
\setglossarystyle{mcolalmtree}%
\renewcommand*\glossaryheader{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
\setglossarystyle{almtree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
\newlength\glsdescwidth
\setlength{glsdescwidth}{0.6\hsize}
}{}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{ }\tabletail{ }}%
    {\begin{supertabular}{lp{glstdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{ }%
  \renewcommand*{\glsgroupheading}[1]{ }%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{ }%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|lp{glstdescwidth}|}%
    {\end{supertabular}}%
  }
\newglossarystyle{superheader}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
    \bfseries\descriptionname\tabularnewline}%
    \tabletail{ }}%
    {\begin{supertabular}{lp{glstdescwidth}}}%
    {\end{supertabular}}%
  }
\newglossarystyle{superheaderborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\hline}%
    \tabletail{\hline}}
    {\begin{supertabular}{|lp{glstdescwidth}|}%
  }

```

```

        {\end{supertabular}}%
    }
\newglossarystyle{super3col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}%
      \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
        {\end{supertabular}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
      \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} & ##2\tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
      \fi
    }
\newglossarystyle{super3colborder}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
      \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}%
      \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
        {\end{supertabular}}%
    }
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
      \tabletail{\hline}%
      \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }
}

```

```

\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{1111}}{%
    \end{supertabular}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand\glossentry[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand\subglossentry[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*\glsgroupskip{}%
  \else
    \renewcommand*\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{1111}}{%
    \end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}}%
    {\end{supertabular}}%
}
\newglossarystyle{super4colheaderborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}}%
    {\end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuper4col}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{ }\tabletail{ }%
     \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname & \bfseries\descriptionname &
     \bfseries\symbolname &
     \bfseries\pagelistname\tabularnewline}\tabletail{ }%
     \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}%
       {llp{\glsdescwidth}llp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
  \setglossarystyle{super4colheaderborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
     \bfseries\entryname &
     \bfseries\descriptionname &
     \bfseries\symbolname &
     \bfseries\pagelistname\tabularnewline\hline}%
     \tabletail{\hline}%
     \begin{supertabular}%
       {llp{\glsdescwidth}llp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}
}

```

8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{ }

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline} \tabletail{\hline}%
    \begin{supertabular}{||1>{\raggedright}p{\glsdescwidth}||}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname \tabularnewline \hline}%
}

```

```

        \tabletail{\hline}
        \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||}%
        {\end{supertabular}}%
    }
\newglossarystyle{superragged3col}{%
    \renewenvironment{theglossary}%
    {\tablehead{\tabletail}{%
        \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
        >{\raggedright}p{\glspagelistwidth}}}%
        {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} &
        ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
    \fi
}
\newglossarystyle{superragged3colborder}{%
    \setglossarystyle{superragged3col}%
    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||%
    >{\raggedright}p{\glspagelistwidth}||}%
    {\end{supertabular}}%
}
\newglossarystyle{superragged3colheader}{%
    \setglossarystyle{superragged3col}%
    \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline}\tabletail}{%
    \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{superragged3colheaderborder}{%
    \setglossarystyle{superragged3colborder}%
    \renewenvironment{theglossary}%
    {\tablehead{\hline

```



```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4col}{%
\renewenvironment{theglossary}%
{ \tablehead{} \tabletail{}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} & ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{altsuperragged4colheader}{%
\setglossarystyle{altsuperragged4col}%
\renewenvironment{theglossary}%
{ \tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4colborder}{%
\setglossarystyle{altsuper4col}%
\renewenvironment{theglossary}%
{ \tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{|l|>{\raggedright}p{\glsdescwidth}|l|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      \l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}%
    {\end{supertabular}}%
}

```

8.34 Rollback v4.49 (glossary-tree-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2021/11/01 v4.49 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
     \let\item\glstreeitem
     \let\subitem\glstreesubitem
     \let\subsubitem\glstreesubsubitem
    }%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%

```

```

\item\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
\renewcommand{\subglossentry}[3]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
\setglossarystyle{index}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{tree}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par

```

```

}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsgroupnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsgrouphypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi

```

```

\glstarget{##2}{\strut}%
\glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
\setglossarystyle{treenoname}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt
{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
\setglossarystyle{treenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsgroupnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsgroupnavigationtarget{##1}{\glsgroupnavigationtitle{##1}}}\par
\indexspace}%
}
\newrobustcmd*{\glsgroupfindwidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forallglossaries[#1]{\@gls@type}%
{%
\forallglsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
}%
{%
\settowidth{\dimen@}%
{\glstreenamefmt{\glsentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@gls@widestname}{glo\@gls@detoklabel{\@glo@label}@name}%
\fi
}%
}%
}
\newcommand*{\glsgroupsetwidest}[2][0]{%
\expandafter\def\csname @gls@widestname\romannumeral#1\endcsname{%
#2}%
}
\newcommand*{\@gls@widestname}{}
\newcommand*{\glsgroupnamebox}[2]{%
\makebox[#1][l]{#2}%
}
\newglossarystyle{almtree}{%

```

```

\renewenvironment{theglossary}%
  {\def\@gls@prevlevel{-1}%
   \mbox{}\par}%
  {\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
    \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
  \hangindent\glstreeindent
  \parindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
  \glossentrydesc{##1}\glspostdescription \space ##2\par
  \def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \@ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength\glstreeindent\gls@tmplen
        \addtolength\glstreeindent\parindent
        \parindent\glstreeindent
      \else
        \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \@glswidestname\space}}{%
              \settowidth{\glstreeindent}{\glstreenamefmt{%
                \csname @glswidestname\romannumeral\@gls@prevlevel
                  \endcsname\space}}}%
          \addtolength\parindent{-\glstreeindent}%
          \setlength\glstreeindent\parindent
        \fi
        \fi
      \hangindent\glstreeindent
      \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
        \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
        \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
        \glossentrydesc{##2}\glspostdescription\space ##3\par
        \def\@gls@prevlevel{##1}%

```

```

}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.35 Rollback v4.52 (glossaries-2022-11-03.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareCurrentRelease{v4.52}{2022-11-03}
\ProvidesPackage{glossaries}[2022/11/03 v4.52 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\newcommand{\glsentencecase}[1]{\makefirstuc{#1}}
\ExplSyntaxOn
\newcommand{\glslowercase}[1]{\text_lowercase:n{#1}}
\newcommand{\glsuppercase}[1]{\text_uppercase:n{#1}}
\ifdef\MFUsentencecase
{%
\newcommand{\glsmfuexcl}[1]{\MFUexcl{#1}}
\newcommand{\glsmfublocker}[1]{\MFUblocker{#1}}
\newcommand{\glsmfuaddmap}[2]{\MFUaddmap{#1}{#2}}
}
{

```

```

\RequirePackage{textcase}
\providecommand{\MFUsentencecase}[1]{\text_titlecase_first:n{#1}}
\providecommand{\glsmfuexcl}[1]{
  \tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
  {
    \tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
  }
}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}
\newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
\newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
  {%
    \@gls@docloadedtrue
  }%
  {%
    \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
  }
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\gls@debug@nr\relax
    % debug=true
    \@gls@debugtrue
    \renewcommand*{\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*{\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
  \let\@gls@showtarget\@gobble

```



```

\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\glsshowtarget\@gobble
\let\glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
\glsshowtargetinner{#1}%
\else
\ifinner
\glsshowtargetinner{#1}%
\else
\glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetfonttext}[1]{%
\ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
\glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}

```

```

\newcommand*\@glsshowtarget}[1]{}
\newcommand*\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{}
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}%
      }%
    \or
      \renewcommand*\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*\@gls@see@noindex}{}%
    \fi
  }
}

```

```

\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec}{section}}%
  {\newcommand*\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec}{#1}}
\newcommand*\@@glossarysecstar}{*}%
\newcommand*\@@glossaryseclabel}{%
\newcommand*\glsautoprefix}{%
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{%
    \renewcommand*\@@glossaryseclabel}{%
    \label{\glsautoprefix@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}}
}
\newcommand*\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\@glo@seeautonumberlist}{%

```

```

\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\glsesclocationsfalse
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
\@gls@declareoption{no long}{\renewcommand*\@gls@loadlong{}}
\IfFileExists{supertabular.sty}{%
\newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}{%
\newcommand*\@gls@loadsuper{}}
\@ifpackageloaded{xtab}{\renewcommand*\@gls@loadsuper{}}{}
\@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}}
\@gls@declareoption{no list}{%
\renewcommand*\@gls@loadlist}{%
\ifdefstring{\@glossary@default@style}{list}%
{\let\@glossary@default@style\relax}%
}%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}}
\@gls@declareoption{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}{%
\ifglsentrycounter
\ifundef\c@glossaryentry
{%
\ifx\@gls@counterwithin\@empty
\newcounter{glossaryentry}%

```

```

        \else
            \newcounter{glossaryentry}[\@gls@counterwithin]%
        \fi
        \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
    }%
    {}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glstentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
    \renewcommand*{\@gls@counterwithin}{#1}%
    \glstentrycountertrue
    \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
    \ifundef\c@glossarysubentry
    {%
        \ifglssubentrycounter
            \ifglstentrycounter
                \newcounter{glossarysubentry}[glossaryentry]%
            \else
                \newcounter{glossarysubentry}%
            \fi
        \fi
        \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
    }%
    {}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand{\glsindexingsetting}{\ifglsxindy xindy\else makeindex\fi}
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
    \renewcommand*{\@glo@default@sorttype}{#1}%
    \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
    \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
    \def\do@glo@storeentry{\@glo@storeentry}%
    \def\@gls@defsorcount##1{%
    \def\@gls@defsorcount##1##2{%
        \ifx\@glo@sort\@glsdefaultsort
            \let\@glo@sort\@glo@name
        \fi
        \let\glsdosanitizesort\@gls@sanitizesort
    }
}

```

```

\glsprestandardsort{\@glo@sort}{##1}{##2}%
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt [1]{%
\ifnum#1<100000 0\fi
\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glsortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%

```

```

\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
  \expandafter\global\expandafter
  \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
  \expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
    \expandafter\glssortnumberfmt
    {\csname glossary@\@glo@type @sortcount\endcsname}}%
  \glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
  \def\glsindexingsetting{none}%
  \def\do@glo@storeentry##1{%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
  }%
  \def\@gls@setsort##1{%
  \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
  {Option sort=none not allowed with \string##1}%
  {(Use sort=def instead)}}%
}
\newcommand*{\@gls@setupsort@clear}{%
  \def\glsindexingsetting{none}%
  \def\do@glo@storeentry##1{%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
  \def\@gls@setsort##1{%
  \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
  {Option sort=clear not allowed with \string##1}%
  {(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
  \if@gls@docloaded
    \newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
  \else
    \newglossary{main}{gls}{glo}{\glossaryname}%
  \fi
  \newcommand*{\gls@tr@set@main@toctitle}{%
    \translatelet{\glossarytoctitle}{Glossary}%
  }%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
  \let\glsdefaulttype\relax
  \renewcommand*{\glsdefmain}{%
}

```

```

\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
  \ifglsacronym
    \renewcommand{\@gls@do@acronymsdef}{%
      \DeclareAcronymList{acronym}%
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
      \renewcommand*{\acronymtype}{acronym}%
      \newcommand*\gls@tr@set@acronym@toctitle}{%
        \translatelet{\glossarytoctitle}{Acronyms}%
      }%
    }%
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
\AtBeginDocument{%
  \ifglsacronym
    \providecommand*\printacronyms[1][1]{%
      \printglossary[type=\acronymtype,#1]%
    }%
  \fi
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*{\acronymtype}{acronym}%
    \newcommand*\gls@tr@set@acronym@toctitle}{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*\@gls@acronymlists{}
\newcommand*\@addtoacronymlists[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*\DeclareAcronymList[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\glsIfListOfAcronyms[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%

```



```

        \def\@gls@before{##1}\def\@gls@after{##2}}%
\gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
\ifx\@gls@after\@nnil
    #4%
\else
    #3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
    \glsIfListOfAcronyms{#1}%
    {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists}[1]{%
    \renewcommand*\@gls@acronymlists{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
    \DeclareAcronymList{#1}%
}
\newcommand*\@gls@counter{page}
\define@key{glossaries.sty}{counter}{%
    \renewcommand*\@gls@counter{#1}%
}
\newcommand*\@gls@nohyperlist{}
\newcommand*\@GlsDeclareNoHyperList}[1]{%
    \ifdefempty\@gls@nohyperlist
    {%
        \renewcommand*\@gls@nohyperlist{#1}%
    }%
    {%
        \appto\@gls@nohyperlist{,#1}%
    }%
}
\define@key{glossaries.sty}{nohypertypes}{%
    \GlsDeclareNoHyperList{#1}%
}
\newcommand*\@GlossariesWarning}[1]{%
    \PackageWarning{glossaries}{#1}%
}
\newcommand*\@GlossariesWarningNoLine}[1]{%
    \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand*\@glosortentrieswarning{%
    \typeout{Using TeX to sort glossary entries---this may
    take a while}%
}
\@gls@declareoption{nowarn}{%
    \if@gls@debug
        \GlossariesWarning{Warnings can't be suppressed in debug mode}%
    \else

```

```

        \renewcommand*\GlossariesWarning}[1]{}%
        \renewcommand*\GlossariesWarningNoLine}[1]{}%
        \renewcommand*\glosortentrieswarning}{}%
        \renewcommand*\@gls@missinglang@warn}[2]{}%
    \fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for ‘#1’.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  ‘glossaries-#2’ or similar}%
}
\@gls@declareoption{noLANGwarn}{%
  \renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheGlossdefined}{%
  \GlossariesWarning{Overriding ‘theGlossary’ environment}%
}
\@gls@declareoption{noRedefwarn}{%
  \renewcommand*\@gls@warnonglossdefined}{}%
  \renewcommand*\@gls@warnontheGlossdefined}{}%
}
\newcommand*\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*\@gls@sanitizename}{%
  \glssetnoexpandfield{name}
}
\newcommand*\@gls@sanitizesymbol}{%
  \glssetnoexpandfield{symbol}
}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}

```

```

}
\newcommand*{\@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{%}
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let^\@firstofone
  \let"@@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let.\@firstofone
  \let~\@firstofone

```

```

\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
}

```

```

\define@boolkey[glS]{sanitize}{symbol}[true]{%
  \GlossariesWarning{sanitize={symbol} package option deprecated}%
  \ifglS@sanitize@symbol
    \glSsetnoexpandfield{symbol}%
    \glSsetnoexpandfield{symbolplural}%
  \else
    \glSsetexpandfield{symbol}%
    \glSsetexpandfield{symbolplural}%
  \fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesorttrue
      \glSsetnoexpandfield{sortvalue}%
    }%
  \else
    \glSsetexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesortfalse
      \glSsetexpandfield{sortvalue}%
    }%
  \fi
}
\glSsanitizesorttrue
\glSsetnoexpandfield{sortvalue}%
\newcommand*{\@glS@noidx@setsanitizesort}{%
  \glSsanitizesortfalse
  \glSsetexpandfield{sortvalue}%
}
\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
  \setbool{glSsanitizesort}{#1}%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
  \else
    \glSsetexpandfield{sortvalue}%
  \fi
  \GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
  \ifthenelse{\equal{#1}{none}}{%
    {%
      \GlossariesWarning{sanitize package option deprecated}%
      \glSsetexpandfield{name}%
      \glSsetexpandfield{symbol}%
      \glSsetexpandfield{symbolplural}%
      \glSsetexpandfield{desc}%
      \glSsetexpandfield{descplural}%
    }%
  }%
}

```

```

}%
{%
  \setkeys[glS]{sanitize}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@glS@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glSifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glSifusetranslator\@firstoftwo
      }%
      {}%
    }%
    {}%
  }%
}
\newcommand{\glSifusedtranslatorDict}[3]{%
  \glSifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@glS@declareoption{notranslate}{%
  \glstranslatefalse
  \let\@glS@usetranslator\relax
  \let\glSifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\glS@translate@val\glS@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\glS@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@glS@usetranslator{%
      \@ifpackageloaded{polyglossia}%
      {%
        \let\glSifusetranslator\@secondoftwo
      }%
      {}%
    }%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%

```

```

        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
    }%
    {}%
} %
{}%
} %
} %
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{}%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\endfortrue
}%
{}%
}
}
\newcommand{\@gls@requested@languages}{}
\define@key{glossaries.sty}{languages}{%
\renewcommand{\@gls@requested@languages}{#1}%
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifglsacrfootnote

```

```

    \@gls@removedoption{footnote}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
  \ifglsacrdescription
    \@gls@removedoption{description}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%
  \ifglsacrsmallcaps
    \@gls@removedoption{smallcaps}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
  \ifglsacrsmaller
    \@gls@removedoption{smaller}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
  \ifglsacrdua
    \@gls@removedoption{dua}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
  \def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}[glsnumbers][true]{}
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\languagename}%
\newcommand*{\gls@xdy@set@lang}[1]{%
  \ifcsundef{\@xdy@#1@language}%
    {%
      \ifcsundef{\@xdy@\glsdefaulttype @language}
        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@\glsdefaulttype @language}}}%
    }%
  {\letcs\@gls@langmod{\@xdy@#1@language}}%
}
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifdefvoid\inputencodingname
  {\def\gls@codepage{utf8}}%
  {\def\gls@codepage{\inputencodingname}}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[][%
  \glsxindytrue

```



```

\setkeys[glS]{xindy}{#1}%
}
\@glS@declareoption{xindygloss}{%
  \glSxindytrue
}
\@glS@declareoption{xindynoglsnumbers}{%
  \glSxindytrue
  \glSxindy@glSnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{\if@filesW #1\fi}
\@glS@declareoption{disablemakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘disablemakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘disablemakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
  }%
  \renewcommand{\@domakeglossaries}[1]{%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been disabled}%
  }%
}
}
\@glS@declareoption{restoremakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
  }%
  \PackageInfo{glossaries}{\string\makeglossaries\space and
    \string\makenoidxglossaries\space have been restored}%
  \let\@domakeglossaries\@firstofone
}
}
\newcommand*{\@do@write@glSlabels}{%

```

```

\AtEndDocument{\do@write@glslabels}%
\let\do@write@glslabels\relax
}
\newcommand*{\do@write@glslabels}{%
  \newwrite\@gls@labelsfile
  \immediate\openout\@gls@labelsfile=\jobname.glslabels
  \protected@edef\@gls@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
  \forallglsentries[\@gls@fulllist]{\@glsentry}%
  {%
    \ifdefempty{\@glsentry}{}%
    {%
      \ifx\do@write@glslabelsname@gobble
        \def\@gls@thisentryname{%
          \else
            \letcs{\@gls@thisentryname}{glo@\@glsdetoklabel{\@glsentry}@name}%
            \@onelevel@sanitize\@gls@thisentryname
          \fi
          \immediate\write\@gls@labelsfile
            {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
        }%
      }%
    \immediate\closeout\@gls@labelsfile
  }
}
\newcommand*{\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@gls@declareoption{writeglslabels}{\do@write@glslabels
    \let\do@write@glslabelsname@gobble}
  \@gls@declareoption{writeglslabelnames}{\do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglsautomake
  \newcommand{\gls@automake@nr}{1}
  \newcommand{\gls@automake@makegloss}{%
    \newcommand{\gls@automake@option}{automake=\gls@automake@val}
    \newcommand*{\@gls@do@automake@err}{%
      \PackageError{glossaries}{You must use
        \string\makeglossaries\space with \gls@automake@option}
      {%
        Either remove the \gls@automake@option\space setting or
        add \string\makeglossaries\space to your document preamble.%
      }%
    }%
  }
}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
  \ifcase\gls@automake@nr\relax
    \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
  \fi
}

```

```

\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or
\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
\let\@gls@doautomake\@gls@do@automake@err
\else
\renewcommand*\@gls@doautomake{}%
\fi
}
\glsautomakefalse
\newcommand*\@gls@doautomake{}
\AtEndDocument{\@gls@doautomake}
\@gls@declareoption{automakegloss}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakegloss}%
\def\gls@automake@val{makegloss}%
\def\gls@automake@nr{3}%
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\let\@gls@doautomake\@gls@do@automake@err
}
\@gls@declareoption{automakeglosslite}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%
\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\@gls@doautomake\@gls@do@automake@err
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
\ifgls savewrites
\renewcommand*\@gls@writefiles{\@gls@writefiles}%
\else
\let\@gls@writefiles\@empty

```

```

\fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
  \ifbool{glscompatible-3.07}%
  {%
    \@gls@removedoption{compatible-3.07}%
  }%
  {}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \@gls@removedoption{compatible-2.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
\gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
{%
  \ifcase\gls@kernelglossredefs@nr\relax
  \def\glossary{\@gls@override@glossary}%
  \def\makeglossary{%
    \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
    \makeglossaries
  }%
\or
  \let\glossary\gls@original@glossary

```

```

\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
  \gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{%
\newcommand*{\@gls@symbolsdef}{%
  \newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
  \newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]]{%
  \newcommand*{\gls@tr@set@symbols@toctitle}{%
    \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
  }%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
  \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
  \newcommand*{\newterm}[2][%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
}
\define@choicekey{glossaries.sty}{mfirstuc}{%
  [\gls@mfirstuc@val\gls@mfirstuc@nr]%
  {expanded,unexpanded}%
}

```

```

{%
  \ifcase\gls@mfirstuc@nr
  % expanded
  \renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
  \or
  % unexpanded
  \renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
  \fi
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
  \@expandtwoargs
  \in@ {\, \CurrentOption ,}{, \@classoptionslist, \@curroptions,}%
  \ifin@
  \@use@ption
  \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
  \fi
  \fi
}
\ProcessOptionsX
\ifglsautomake
\ifdef\directlua{\RequirePackage{shellesc}}{}
\fi
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*\setupglossaries}[1]{%
  \renewcommand*\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
  \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
  \def\@gls@setupshortcuts{%
    \ifglsacrshortcuts
    \DefineAcronymSynonyms
    \fi
  }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsubentrycounter
  \let\@gls@doentrycounterdef\relax
  \else
  \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglsubentrycounter

```

```

        \let\@gls@dosomeentrycounterdef\relax
    \else
        \let\@gls@dosomeentrycounterdef\@gls@define@glossarysubentrycounter
    \fi
    \setkeys{glossaries.sty}{#1}%
    \@gls@setacrstyle
    \@gls@setupshortcuts
    \@gls@do@acronymsdef
    \@gls@do@numbersdef
    \@gls@do@symbolssdef
    \@gls@do@indexdef
    \@gls@doentrycounterdef
    \@gls@dosomeentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
    \ifcsundef{chapter}{}%
    {%
        \let\@gls@old@chapter\@chapter
        \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
        \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
    }%
}%
{}
\newcommand*{\@gls@onlypremakeg}{%
\newcommand*{\@onlypremakeg}[1]{%
    \ifx\@gls@onlypremakeg\@empty
        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
    \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
    \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
    \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}
\providecommand*{\pagelistname}{Page List}

```

```

\providecommand*\glssymbolsgroupname}{Symbols}
\providecommand*\glsnumbersgroupname}{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}[2019/11/30]
  \ifvoid\@gls@requested@languages{}%
  {%
    \@for\@gls@thislang:=\@gls@requested@languages\do{%
      \TrackIfKnownLanguage{\@gls@thislang}{}%
      {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
    }%
  }%
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp
            \renewcommand*\glossaryname}{\translate{Glossary}}%
          }%
          \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
        }%
      }%
    }%
  }%
}{}%

```



```

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glssymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
}%
}{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{\nopostdesc}
\newcommand*\glspar{\par}
\newcommand*\setStyleFile[1]{%
  \renewcommand*\gls@istfilebase{#1}%
}

```

```

\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase{\jobname}
\newcommand*\@istfilename[1]{ }
\newcommand*\glscompositor{.}
\newcommand*\glsSetCompositor[1]{%
  \renewcommand*\glscompositor{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphacompositor{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*\gls@suffixF{ }
\newcommand*\glsSetSuffixF[1]{%
  \renewcommand*\gls@suffixF{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*\gls@suffixFF{ }
\newcommand*\glsSetSuffixFF[1]{%
  \renewcommand*\gls@suffixFF{#1}}
}
\ifcsundef{hyperlink}%
{ %
  \newcommand*\glsnumberformat[1]{#1}%
}%
{ %
  \newcommand*\glsnumberformat[1]{\glsnumber{#1}}%
}
\newcommand{\glslocationstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%

```

```

{\csgdef{@glossary preamble@#1}{#2}}%
{\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossary postamble}{}
\newcommand*{\glossary section}[2][\@gls@title]{%
\def\@gls@title{#2}%
\ifcsundef{phantom section}%
{%
\@glossary section{#1}{#2}%
}%
{%
\@pglossary section{#1}{#2}%
}%
\glsglossary mark{\glossary toctitle}%
}
\ifcsundef{glossary mark}%
{%
\newcommand{\glsglossary mark}[1]{\glossary mark{#1}}
}%
{%
\@ifclassloaded{memoir}
{%
\newcommand{\glsglossary mark}[1]{%
\ifglsucmark
\markboth{\memUchead{#1}}{\memUchead{#1}}%
\else
\markboth{#1}{#1}%
\fi
}
}%
{%
\newcommand{\glsglossary mark}[1]{%
\ifglsucmark
\@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
\else
\@mkboth{#1}{#1}%
\fi
}
}
}
\providecommand{\glossary mark}[1]{%
\ifglsucmark
\@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
\else
\@mkboth{#1}{#1}%
\fi
}
\newcommand*{\setglossary section}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossary section}[2]{%

```

```

\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname[#1]{#2}%
}%
{%
  \csname\@glossarysec\endcsname*{#2}%
  \@gls@toc{#1}{\@glossarysec}%
}%
\@glossaryseclabel
}
\newcommand*\@pglossarysection}[2]{%
  \glsclearpage
  \phantomsection
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname{#2}%
  }%
  {%
    \@gls@toc{#1}{\@glossarysec}%
    \csname\@glossarysec\endcsname*{#2}%
  }%
  \@glossaryseclabel
}
\newcommand*\gls@docclearpage{%
  \ifthenelse{\equal{\@glossarysec}{chapter}}%
  {%
    \ifcsundef{cleardoublepage}%
    {%
      \clearpage
    }%
    {%
      \ifcsdef{if@openright}%
      {%
        \if@openright
          \cleardoublepage
        \else
          \clearpage
        \fi
      }%
    }%
    \cleardoublepage
  }%
}
\newcommand*\glsclearpage{\gls@docclearpage}
\newcommand*\@gls@toc}[2]{%
  \ifglstoc
  \ifglsnumberline

```

```

        \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
    \else
        \addcontentsline{toc}{#2}{#1}%
    \fi
\fi
}
\newcommand*\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@do@ifinlist{%
        \def\@gls@listsuffix{##2}%
        \ifx\@gls@listsuffix\@empty
            #4%
        \else
            #3%
        \fi
    }%
    \@do@ifinlist,#2,#1,\end@do@ifinlist
}
\ifglsxindy
    \newcommand*\@xdycounters{\glscounter}
    \newcommand*\GlsAddXdyCounters[1]{%
        \@for\@gls@ctr:=#1\do{%
            \edef\@do@addcounter{%
                \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
                {%
                    \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
                        \noexpand\@gls@ctr}%
                }%
            }%
            \@do@addcounter
        }
    }
    \@onlypremakeg\GlsAddXdyCounters
\else
    \newcommand*\GlsAddXdyCounters[1]{%
        \glsnoxindywarning\GlsAddXdyAttribute

```

```

}
\fi
\newcommand*\@disabled@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\@glssaddxdyattribute{#1}{\@this@counter}%
    }
    \gls@do@addxdyattribute
  }%
  \let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
  \glsnoindeywarning\GlsAddXdyAttribute}
\fi
\ifglxindy
\newcommand*{\@gls@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}

```

```

\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\GlsAddXdyAttribute{glsignore}
}
\else
  \let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ~^J
      (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxindy
  \newcommand*{\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp

```

```

\ifx\@gls@tmp\@gls@roman
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      \string"roman-numbers-lowercase\string"%
    }%
\else
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
      :sep \string"\@gls@roman\string"%
    }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@gls@Alphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
  \newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyLocation}[3][[]]{%
    \def\@gls@tmp[#1]{%
      \ifx\@gls@tmp\@empty
        \edef\@xdyuserlocationdefs{%
          \@xdyuserlocationdefs ^^J%
          (define-location-class \string"#2\string"^^J\space\space
            \space(:sep \string"{}\glsopenbrace\string" #3
              :sep \string"\glsclosebrace\string"))
        }%
      \else

```



```

\edef\@xdyuserlocationdefs{%
  \@xdyuserlocationdefs ^^J%
  (define-location-class \string"#2\string"^^J\space\space
  \space(:sep "\glsoopenbrace"
    #1
    :sep "\glsclosebrace\glsoopenbrace" #3
    :sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#2\string"%
}
\@onlypremakeg\GlsAddXdyLocation
\else
  \newcommand*\GlsAddXdyLocation[2]{%
    \glsnoinindywarning\GlsAddXdyLocation}
\fi
\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]{%
    \glsnoinindywarning\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]{%
    \glsnoinindywarning\GlsAddSortRule}

```

```

\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\def\@xdy@language#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage\@empty
      \ifpackage@loaded{fontspec}{\def\@gls@codepage{utf8}}}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifglx@xindy@glsnumbers
    \def\@xdy@lettergroups{(define-letter-group
      \string"glxnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"

```

```

        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
        \@xdynumbergrouporder)}
\else
  \def\@xdylettergroups{}
\fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries}[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}

\newcommand*\forglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}

\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@thisglo@}%
  {%
    \forglsentries[\@@thisglo@]{#2}{#3}%
  }%
}

\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glo@type@#1@out}{#3}{#2}%
}

\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}

\newcommand*\glsdetoklabel}[1]{#1}
\newcommand*\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}

\newcommand*\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}

\newcommand*\glsdoifexists}[2]{%

```

```

\ifglentryexists{#1}{#2}{%
  \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
    has not been defined}{You need to define a glossary entry before you
    can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@glentries\expandafter
      [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs@glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel@glo@parent
        {%
          \def\do@glshaschildren{#2}%
          \@endfortrue
        }%
      }%
    }%
  }%
  \do@glshaschildren
}

```

```

    }%
  }
  \newcommand{\ifglshasparent}[3]{%
    \glsdoifexists{#1}%
    {%
      \ifcsemtyp{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
    }%
  }
  \newcommand*{\ifglshasdesc}[3]{%
    \ifcsemtyp{glo@\glsdetoklabel{#1}@desc}%
    {#3}%
    {#2}%
  }
  \newcommand*{\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
  }
  \newrobustcmd*{\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
      \ifdefequal\@glo@symbol\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newrobustcmd*{\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
      \ifdefequal\@glo@long\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newrobustcmd*{\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
      \ifdefequal\@glo@short\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@\glsdetoklabel{#2}@#1}{#3}{#4}}
  \newrobustcmd*{\ifglshasfield}[4]{%

```

```

\glsdoifexists{#2}%
{%
  \letcs{\@glo@thisvalue}{glo\glsdetoklabel{#2}@#1}%
  \ifdef\@glo@thisvalue
  {%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  \ifdequal\@glo@thisvalue\@gls@default@value
  {%
    #4%
  }%
  {%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \@gls@fetchfield{\@gls@thisfield}{#1}%
  \ifdef\@gls@thisfield
  {%
    \letcs{\@glo@thisvalue}{glo\glsdetoklabel{#2}@ \@gls@thisfield}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  \ifdequal\@glo@thisvalue\@gls@default@value
  {%
    #4%
  }%
  {%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \GlossariesWarning{Unknown entry field '#1'}%
  #4%
}%
}%
}
\newcommand*{\glscurrentfieldvalue}{ }
\newcommand*{\@glo@types}{,}
\newcommand*{\@gls@provide@newglossary}{%

```

```

\protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
\let\@gls@provide@newglossary\relax
}
\newcommand*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
\protected@edef\@glo@tmp{#1}%
\ifx\@glo@tmp\@gls@defaulttype
\GlossariesWarning{\string\defglsentryfmt\space attempting to set
a display style for \string\glsdefaulttype\space but there
doesn't seem to be a default glossary. If you have used
'nomain', you need to define an alternative glossary}%
\else
\csgdef{gls@\@glo@tmp @entryfmt}{#2}%
\fi
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
\ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
\doifglossarynoexistsordo{#2}%
{%
\ifundef\glsdefaulttype
{%
\gdef\glsdefaulttype{#2}%
}{}%
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname

```

```

\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
}

```



```

\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*\@gls@nohyperlist}{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\@ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%

```

```

    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning
  {‘\gls@xr@key’ key doesn’t have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
  [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]}%
  {%
    \ifcase\gls@nonumberlist@nr\relax
      \def\@glo@prefix{\glsnonextpages}%
      \@gls@savenonumberlist{true}%
    \else
      \def\@glo@prefix{\glsnextpages}%
      \@gls@savenonumberlist{false}%
    \fi
  }
\newcommand*{\gls@noidx@nonumberlist}[1]{%
  \cslet\glo@\@glo@label @prenumberlist}\@glo@prefix
}%
\newcommand{\glsnoidxprenumberlist}[1]{%
  \csuse{glo@\glsdetoklabel{#1}@prenumberlist}}
\newcommand*{\@gls@savenonumberlist}[1]{%
}
\newcommand*{\@gls@initnonumberlist}{%
}

```

```

\newcommand*{\@gls@storenonumberlist}[1]{%
\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
    {%
      \cslet{glo@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
    }%
    {}%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@gls@name}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You

```

```

haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
    Maybe makeglossaries + rerun required}%
  \else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
  \fi
}
\newcommand*\@gls@defaultsort}{\@glo@name}
\newcount\gls@level
\newcommand*\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand*\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%
  }

```

```

        \@@gls@noexpand@field{#2}{#3}{#4}%
    }%
}
\newcommand{\@@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {%
    \ifglsentryexists{#1}%
    {%
      \xifinlist{#1}{\gls@docdef@labellist}%
      {%
        \GlossariesWarning{Entry ‘#1’ has already been defined in the
          document environment (ignoring new definition)}%
      }%
      {%
        \xifinlist{#1}{\gls@glsdefs@labellist}%
        {%
          \gls@defglossaryentry{#1}{#2}%
          \gls@docdef@addlabel{#1}%
          \@gls@writedef{#1}%
        }%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in
            the preamble (in which case remove the duplicate definition)
            or you have just updated to glossaries.sty
            v4.47 (in which case delete ‘\jobname.glsdefs’)}%
        }%
      }%
    }%
  }%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}

```

```

        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
    }%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
    \gls@enablesavenonumberlist
    \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
    \makeatletter
    \InputIfFileExists{\jobname.glsdefs}{-}{-}%
    \@gls@restreat
    \undef\@gls@restreat
    \gls@defdocnewglossaryentry
}
\newcommand*\glsdefs@newdocentry}[2]{%
    \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
        (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
        \gls@glsdefs@addlabel{#1}%
        \gls@defglossaryentry{#1}{#2}%
    }%
}
}
\newcommand*\@gls@writedef}[1]{%
    \immediate\write\@gls@deffile
    {%
        \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
        \expandafter\@gobble\string\{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
        \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
        \ifdef\glo@value
        {%
            \@onelevel@sanitize\glo@value
            \immediate\write\@gls@deffile
            {%
                \expandafter\@firstoftwo\@gls@map
                =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
                \glspercentchar
            }%
        }%
        }%
    }%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
}

```

```

    \expandafter\@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```



```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
}%
}%
}%

```

```

\expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#6' as helper command
        '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @Gls@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@Gls@user@#1@}{##1}{##2}}%
          {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#7' as helper command
        '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @GLS@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@GLS@user@#1@}{##1}{##2}}%
          {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
\glsmfuaddmap{#3}{#4}%

```

```

    \glsmfuaddmap{#5}{#6}%
    \glsmfublocker{#7}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newrobustcmd{\glsfieldxdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csxdef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldedef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csedef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldgdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldddef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%

```

```

\ifcsdef{glo@\@glo@label @#2}%
{%
  \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
}%
{%
  \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}
\newrobustcmd{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*\glsunexpandedfieldvalue[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}

```

```

    }%
  }%
}
\newcommand{\ifglsfieldcseq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key '#2' doesn't exist}{}%
    }%
  }%
}
\newcommand*{\gls.writedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \@org@newglossaryentryprehook
    }%
    \renewcommand*{\gls@assign@desc}[1]{%
      \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
      \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
    }
    \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@glsname
  \let\@glo@desc\@glsnodesc

```

```

\let\@glo@descplural\@gls@default@value
\let\@glo@type\@gls@default@value
\let\@glo@symbol\@gls@default@value
\let\@glo@symbolplural\@gls@default@value
\let\@glo@text\@gls@default@value
\let\@glo@plural\@gls@default@value
\let\@glo@first\@gls@default@value
\let\@glo@firstplural\@gls@default@value
\let\@glo@sort\@gls@default@value
\let\@glo@counter\@gls@default@value
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\@empty
  \fi
}%

```

```

    {%
  }%
  \protected@edef\@glo@list@\csname glo@list@\@glo@type\endcsname}%
  \expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
    \@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
    {%
      \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
      \def\@glo@parent{}%
      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
    }%
    {%
      \ifglsentryexists{\@glo@parent}%
      {%
        \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
          \@glo@parent}%
        \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
        \advance\gls@level by 1\relax
        \ifx\@glo@name\@gls@name
          \expandafter\let\expandafter\@glo@name
            \csname glo@\@glo@parent @name\endcsname
          \ifx\@glo@plural\@gls@default@value
            \expandafter\let\expandafter\@glo@plural
              \csname glo@\@glo@parent @plural\endcsname
          \fi
        \fi
      }%
    }%
    \PackageError{glossaries}%
    {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn’t exist%
    }%
    {%
      Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%

```

```

\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glo@type@\@glo@type @counter}%
{%
  \def\@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@gls@name
  \let\@gls@name\@gls@name
  \let\@gls@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\glo@label @numberlist}%
{%
  \csxdef{glo@\glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
  \@gls@storenonumberlist{\@glo@label}%
  \def\@glo@@desc{\@glo@first}%
  \ifx\@glo@desc\@glo@@desc
    \let\@glo@desc\@glo@first
  \fi
\ifx\@glo@desc\@gls@nodesc

```



```

    \@glsnodesc
    \let\@glsdesc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
    \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
    \gls@assign@field\expandafter
    {csname glo@\@glo@label @symbol\endcsname}
    {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
    \noexpand\let\expandafter\noexpand
        \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
    \noexpand\let\expandafter\noexpand
        \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
    \csdef{glo@\@glo@label @index}{}%
}
{%
    \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
    \ifdefvoid\@glo@see{%
        {%
            \protected@edef\@do@glssee{%
                \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
                \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
            \@do@glssee
        }%
        \@glo@autoseehook
    }%
}
\newcommand*{\@glo@autoseehook}{-}
\newcommand*{\@newglossaryentryprehook}{-}
\newcommand*{\@newglossaryentryposthook}{-}
\newcommand*{\@newglossaryentrydefcounters}{-}

```

```

\newcommand*\glsmoveentry}[2]{%
  \protected@edef\glo@thislabel{\glsdetoklabel{#1}}%
  \protected@edef\glo@type{\csname glo@\glo@thislabel @type\endcsname}%
  \def\glo@list{,%}
  \forlslentries[\glo@type]{\glo@label}%
  {%
    \ifdefequal\@glo@thislabel\glo@label
      {}{\protected@eappto\glo@list{\glo@label,}}%
    }%
  \cslet\glolist@\glo@type{\glo@list}%
  \csdef\glo@\@glo@thislabel @type}{#2}%
}
\ifglxindy
  \newcommand*\@glossaryentryfield{\string\glossentry}
\else
  \newcommand*\@glossaryentryfield{\string\glossentry}
\fi
\ifglxindy
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\else
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\fi
\newcommand{\glo@storeentry}[1]{%
  \protected@edef\glo@esclabel{#1}%
  \gls@checkmkidxchars\glo@esclabel
  \protected@edef\glo@sort{\csname glo@#1@sort\endcsname}%
  \gls@checkmkidxchars\glo@sort
  \gls@checkmkidxchars\glo@prefix
  \protected@edef\glo@parent{\csname glo@#1@parent\endcsname}%
  \ifglxindy
    \ifx\glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string\@glo@sort\string" %
        \string\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string\@glo@sort\string" %
        \string\@glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\glo@parent\@empty
      \@onelevel@sanitize\@glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \glo@sort\gls@actualchar\@glo@prefix

```

```

        \@glossaryentryfield{\@glo@esclabel}%
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
        \@glo@sort\@gls@actualchar\@glo@prefix
        \@glossarysubentryfield
        {\csname glo@#1@level\endcsname}\@glo@esclabel}%
      }%
    \fi
  \fi
}
\newcommand*\@gls@if@measuring}[2]{%
  \ifmeasuring@
    #1%
  \else
    #2%
  \fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand*\gls@if@measuring}[2]{#2}
\AtBeginDocument{%
  \@ifpackageloaded{amsmath}%
  {\let\gls@if@measuring\@gls@if@measuring}%
  }%
  \let\gls@org@refstepcounter\refstepcounter
  \let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring}{#1}}
\newcommand*\glsifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter}[1]{%
  \ifcsdef{c@#1}%
  {%
    \epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
    \expandafter\advance\csname c@#1\endcsname by \@ne
  }%
  }%
}
\newcommand*\gls@start@measuring}{%
  \ifx\glsifmeasuring\@firstoftwo
  \else
    \let\gls@org@target\glstarget
    \let\gls@org@link\glslink
    \def\gls@measuring@reset{}%
  \fi
  \let\glsifmeasuring\@firstoftwo
  \let\refstepcounter\gls@local@stepcounter
  \let\label\@gobble
  \let\glstarget\@secondoftwo

```

```

\let\@glslink\@secondoftwo
}
\newcommand{\gls@stop@measuring}{%
\renewcommand*\gls@ifmeasuring{\gls@if@measuring}%
\let\refstepcounter\gls@org@refstepcounter
\let\label\gls@org@label
\let\glstarget\gls@org@target
\let\@glslink\gls@org@link
\gls@measuring@reset
\def\gls@measuring@reset{%
}
\newcommand{\glsmeasureheight}[2]{%
\settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasuredepth}[2]{%
\settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasurewidth}[2]{%
\settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{##1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
}%
}
\newcommand*\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%

```

```

\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glsunset{#1}%
  }%
}%
}
\newcommand*\@glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}

```

```

}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \ifglsresetcurrcount\csgdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}
\def\@cgl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglformat{##2}{##3}%
        \glsunset{##2}%
    \else
        \@gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cgl spl@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cgl spl format{##2}{##3}%
        \glsunset{##2}%
    \else
        \@gls spl@{##1}{##2}[##3]%
    \fi
}%
\def\@cGls@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cGls format{##2}{##3}%
        \glsunset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cGls spl@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cGls spl format{##2}{##3}%
        \glsunset{##2}%
    \else
        \@Gls spl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{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}
\glsmfuaddmap{\cgl}{\cGls}
\newcommand*{\@cGls}[2][{}]{%
    \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}[]}%
}
\def\@cGls@#1#2[#3]{%
    \GlossariesWarning{\string\cGls\space is defaulting to
        \string\Gls\space since you haven't enabled entry counting}%
    \@Gls@{#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
    \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][{}]{%
    \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}[]}%
}
\def\@cglspl@#1#2[#3]{%
    \GlossariesWarning{\string\cglspl\space is defaulting to
        \string\glspl\space since you haven't enabled entry counting}%
    \@glspl@{#1}{#2}[#3]%
}
\newcommand*{\cglsplformat}[2]{%
    \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\glsmfuaddmap{\cglspl}{\cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
    \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[]}%
}
\def\@cGlspl@#1#2[#3]{%
    \GlossariesWarning{\string\cGlspl\space is defaulting to

```



```

\string\Glspl\space since you haven't enabled entry counting}%
\@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
\ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries}[2][\@gls@default]{%
\let\@gls@default\glsdefaulttype
\def\glsdefaulttype{#1}\input{#2}%
\let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glstextformat}[1]{#1}
\newcommand*\glsentryfmt{\glsgenentryfmt}
\newcommand*\glsgenentryfmt{%
\ifdefempty\glscustomtext
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
\glsentryplural{\glslabel}\glsinsert
}%
{%
\glsentryfirstplural{\glslabel}\glsinsert
}%
}%
{%
\ifglsused\glslabel
{%
\Glsentryplural{\glslabel}\glsinsert
}%
{%
\Glsentryfirstplural{\glslabel}\glsinsert
}%
}%
\ifglsused\glslabel
{%
\glsuppercase{\glsentryplural{\glslabel}\glsinsert}%
}%
{%
\glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%
}%
}%
}%
\glscapscase
}

```

```

    {%
      \ifglused\glslabel
      {%
        \glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \glsuppercase{\glsentrytext{\glslabel}\glsinsert}%
      }%
      {%
        \glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%
      }%
    }%
  }%
}
\newcommand*{\glsnacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \glsuppercase
            {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```



```

\newcommand*\genacrfullformat}[2]{%
  \glsentrylong{#1}#2\space
  (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset@glslocalunset
  \else
    \let\@gls@do@glsunset@glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\glslinkvar}[3]{#1}

```

```

\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3] []{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
  \@gls@hyp@opt\@Gls@@link
}
\newcommand*{\@Gls@@link}[3] []{%
  \@gls@@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}

```

```

}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\gls\type}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}}{}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{\glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\gls\type{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glo@linkprefix\glslabel}{\gls\textformat{#3}}%
  \else
    \glsdonohyperlink{\glo@linkprefix\glslabel}{\gls\textformat{#3}}%
  \fi
  \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
  \@gls@restore@glslocal
}
\newcommand*{\glsencapwrcontent}[1]{#1}
\newcommand*{\glo@linkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}%
  {
    \ifcsundef{xatlevel@}%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
  }%
  \ifx\xatlevel@\@empty
    \edef\theglsentrycounter{\expandafter\noexpand

```

```

        \csname the\@gls@counter\endcsname}%
\else
  \savecounters@
  \advance\c@equation by 1\relax
  \edef\theglentrycounter{\csname the\@gls@counter\endcsname}%
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\@gls@Hcounter{\theglentrycounter}%
  }%
  {%
    \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
  }%
  \protected@edef\theHglentrycounter{\@gls@Hcounter}%
  \restorecounters@
\fi
}%
}%
{%
  \edef\theglentrycounter{\expandafter\noexpand
    \csname the\@gls@counter\endcsname}%
}%
\ifx\@gls@Hcounter\@empty
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\theHglentrycounter{\theglentrycounter}%
  }%
  {%
    \protected@edef\theHglentrycounter{\expandafter\noexpand
      \csname theH\@gls@counter\endcsname}%
  }%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
  \expandafter\@glo@check@mkidxrangear#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[#4]{#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1(\relax
  \def\@glo@prefix{()}%
  \if\relax#2\relax
    \def\@glo@suffix{glsnumberformat}%
  \else
    \def\@glo@suffix{#2}%
  \fi
\else
  \if#1)\relax

```

```

\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffi{x{glsnumberformat}}%
\else
\def\@glo@suffi{x{#2}}%
\fi
\else
\def\@glo@prefix{ }\def\@glo@suffi{x{#1#2}}%
\fi
\fi}
\newcommand*\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\ifgls@esclocations
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostsubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostsubst
}%
\fi
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
\ifgls@xindy
\@gls@escbsdq{#1}%
\else
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkquote#1\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescquote#1\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescactual#1\@nil\?\?\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%

```



```

\expandafter\@gls@checkactual#1\@nil??\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkbar#1\@nil||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesclevel#1\@nil!!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\edef\@do@gls@checkesc{%
  \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
  \@backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
  \ifx\null#3\null
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
    \def\@@gls@checkquote{\relax}%
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
    \def\@@gls@checkquote{\@gls@checkquote#3\null}%
  \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@quotechar}%
    \ifx\null#3\null
      \def\@@gls@checkquote{\@gls@checkquote#2""\null}%
    \else
      \def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
    \fi
  \fi
  \@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%

```

```

\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2\""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2\#"#3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null

```

```

\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeschar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%
\def\@gls@checkeschar{\@gls@checkeschar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkeschar{\@gls@checkeschar#2\|\|\null}%
\else
\def\@gls@checkeschar{\@gls@checkeschar#2\|#3\null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkeslevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeslevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@gls@checkeslevel{\@gls@checkeslevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkeslevel{\@gls@checkeslevel#2\!\!\null}%
\else
\def\@gls@checkeslevel{\@gls@checkeslevel#2\!#3\null}%
\fi
\fi
\@gls@checkeslevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%

```

```

\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2|#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%

```

```

        \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
|gdef|\@gls@checkesc#1\#2\#3\null{%
|\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
|\toks@={#1}%
\ifx\null#2\null
|\ifx\null#3\null
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@}%
|\def|\@gls@checkesc{\relax}%
|\else
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\backslashchar|\@gls@quotechar|\backslashchar}%
|\def|\@gls@checkesc{\@gls@checkesc#3\null}%
|\fi
|\else
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\backslashchar}%
|\ifx\null#3\null
|\def|\@gls@checkesc{\@gls@checkesc#2\\\null}%
|\else
|\def|\@gls@checkesc{\@gls@checkesc#2\#3\null}%
|\fi
|\fi
|\@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
|\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
|\toks@={#1}%
\ifx\null#2\null
|\ifx\null#3\null
|\edef|\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
|\def\@gls@xdycheckquote{\relax}%
|\else
|\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
|string"\string"}%

```

```

        \def\@@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
    \fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string\}%
\ifx\null#3\null
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\fi
\@@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

```

```

\newlength\gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
  \glsdoshowtarget{#1}%
  {%
    \glsmeasureheight{\gls@tmplen}{#2}%
    \raisebox{\gls@tmplen}{\hypertarget{#1}{}}%
    #2%
  }%
}
\newcommand*\glsdohyperlink}[2]{%
  \glsdoshowtarget{#1}{\hyperlink{#1}{#2}}%
}
\newcommand*\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*\gls{\@gls@hyp@opt\@gls}
\newcommand*\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}]%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\gls@ifplural\@secondoftwo
    \let\gls@scapscase\@firstofthree
    \let\gls@customtext\@empty
  }
}

```

```

\def\glsinsert{#3}%
\def\glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
\new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}}[{}]}%
}
\def\@Gls@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@caps@case\@secondofthree
\let\gls@custom@text\@empty
\def\glsinsert{#3}%
\def\glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2][{}]{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}}[{}]}%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@caps@case\@thirdofthree
\let\gls@custom@text\@empty
\def\glsinsert{#3}%
\def\glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2][{}]{%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2}}[{}]}%
}
\def\@glspl@#1#2[#3]{%

```



```

\glsdoifexists{#2}%
{%
  \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
  \let\glsifplural\@firstoftwo
  \let\glsescapscase\@firstofthree
  \let\glscustomtext\@empty
  \def\glsinsert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gls@link[#1]{#2}{\@glo@text}%
  \@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\glsmfuaddmap{\glspl}{\Glspl}
\newcommand*{\@Glspl}[2][{}]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[{}]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[{}]}%
}
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}

```

```

}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3] []{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls@scaps\@firstofthree
    \def\gls@customtext{#3}%
    \def\gls@insert{}%
    \def\@glo@text{\csname gls@\gls@type @entryfmt@endcsname}%
    \@gls@link{#1}{#2}{\@glo@text}%
    \@gls@do@gls@unset{#2}%
  }%
  \gls@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3] []{%
  \@glsdisp{#1}{#2}{\gls@sentencecase{#3}}%
}
\gls@fu@addmap{\glsdisp}{\Glsdisp}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3] {%
  \glsdoifexists{#2}{%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \@gls@link{#1}{#2}{#3}%
    }%
  }%
  \gls@postlinkhook
}
\newrobustcmd*{\gls@text}{\@gls@hyp@opt\@gls@text}
\newcommand*{\@gls@text}[2] []{%
  \new@ifnextchar[{\@gls@text@{#1}{#2}}{\@gls@text@{#1}{#2} []}]
\def\@gls@text@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\gls@entrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\gls@fu@blocker{\GLStext}
\newcommand*{\@GLStext}[2] []{%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} []}]
\def\@GLStext@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\gls@supercase{\gls@entrytext{#2}#3}}%
}
\newrobustcmd*{\Gls@text}{\@gls@hyp@opt\@Gls@text}
\gls@fu@addmap{\gls@text}{\Gls@text}
\newcommand*{\@Gls@text}[2] []{%
  \new@ifnextchar[{\@Gls@text@{#1}{#2}}{\@Gls@text@{#1}{#2} []}]
\def\@Gls@text@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Gls@entrytext{#2}#3}%
}
\newrobustcmd*{\gls@first}{\@gls@hyp@opt\@gls@first}

```

```

\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]
\def\@glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\glsmfuaddmap{\glsfirst}{\Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\glsmfublocker{\GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
\def\@GLSfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\glsmfublocker{\GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\glstentryfirstplural{#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{\glstentryfirstplural{#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}{\glstentryname{#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}{\glstentryname{#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{\glstentryname{#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}{\glstentrydesc{#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}{\glstentrydesc{#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{\glstentrydesc{#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*{\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*{\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}
\glsmfuaddmap{\glssymbolplural}{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\glsmfuaddmap{\glsuseri}{\Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}]
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\glsmfuaddmap{\glsuserii}{\Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}

```

```

\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2] [] {%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\glsmfuaddmap{\glsuserv}{\Glsuserv}
\newcommand*{\@Glsuserv}[2] [] {%
  \new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}]
\def\@Glsuserv@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glsentryuser{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2][{}]{%
\new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2}}]}
\def\@GLSuserv@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuser{#2}#3}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2][{}]{%
    \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}}]}
\def\@glsuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuser{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\glsmfuaddmap{\glsuservi}{\Glsuservi}
\newcommand*{\@Glsuservi}[2][{}]{%
    \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2}}]}
\def\@Glsuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuser{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
\newcommand*{\@GLSuservi}[2][{}]{%
    \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}}]}
\def\@GLSuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuser{#2}#3}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2][{}]{%
    \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2}}]}
}
\def\@acrshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsifplural\@secondoftwo
        \let\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}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\ns@Acrshort}[2][{}]{%

```



```

\new@ifnextchar [{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}%
}
\def\@Acrshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsentryshort{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gl@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] []{%
\new@ifnextchar [{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
\new@ifnextchar [{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%

```

```

        \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
}
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] []{%
    \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\gls caps case\@secondofthree
        \let\glsinsert\@empty
        \def\gls custom text{%
            \acronymfont{\Glsentryshortpl{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\gls caps case\@thirdofthree
        \let\glsinsert\@empty
        \def\gls custom text{%
            \glsupercase{\acronymfont{\glsentryshortpl{#2}}#3}%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2] []{%
    \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```

```

\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*\@Acr long{\@gls@hyp@opt\@ns@Acr long}
\glsmfuaddmap{\Acr long}{\Acr long}
\newcommand*\@ns@Acr long}[2][{}]{%
  \new@if next char[{\@Acr long{#1}{#2}}{\@Acr long{#1}{#2}}[{}]}%
}
\def\@Acr long#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*\@ACR long{\@gls@hyp@opt\@ns@ACR long}
\glsmfublocker{\ACR long}
\newcommand*\@ns@ACR long}[2][{}]{%
  \new@if next char[{\@ACR long{#1}{#2}}{\@ACR long{#1}{#2}}[{}]}%
}
\def\@ACR long#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsupercase{\glsentrylong{#2}#3}%
    }%
  }%
}

```

```

    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapsaps\@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{%
    \glsuppercase{\glsentrylongpl{#2}#3}%
  }%
  \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gl@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\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*{\glsentryname}[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
  \fi
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
}
\newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\glsmfuaddmap{\glsentrydesc}{\Glsentrydesc}

```

```

\newcommand*\glsentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
\newcommand*\Glsentrydescplural}[1]{%
  \@Glsentryfield{#1}{descplural}%
}
\glsmfuaddmap{\glsentrydescplural}{\Glsentrydescplural}
\newcommand*\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newcommand*\Glsentrytext}[1]{%
  \@Glsentryfield{#1}{text}%
}
\glsmfuaddmap{\glsentrytext}{\Glsentrytext}
\newcommand*\glsentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
\newcommand*\Glsentryplural}[1]{%
  \@Glsentryfield{#1}{plural}%
}
\glsmfuaddmap{\glsentryplural}{\Glsentryplural}
\newcommand*\glsentrysymbol}[1]{%
  \@gls@entry@field{#1}{symbol}%
}
\newcommand*\Glsentrysymbol}[1]{%
  \@Glsentryfield{#1}{symbol}%
}
\glsmfuaddmap{\glsentrysymbol}{\Glsentrysymbol}
\newcommand*\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newcommand*\Glsentrysymbolplural}[1]{%
  \@Glsentryfield{#1}{symbolplural}%
}
\glsmfuaddmap{\glsentrysymbolplural}{\Glsentrysymbolplural}
\newcommand*\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newcommand*\Glsentryfirst}[1]{%
  \@Glsentryfield{#1}{first}%
}
\glsmfuaddmap{\glsentryfirst}{\Glsentryfirst}
\newcommand*\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newcommand*\Glsentryfirstplural}[1]{%
  \@Glsentryfield{#1}{firstpl}%
}
\glsmfuaddmap{\glsentryfirstplural}{\Glsentryfirstplural}
\newcommand*\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}

```

```

\newcommand*\glsentrytitlecase}[2]{%
  \glstexorpdfstring
  {\@glsentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}
\glsmfuaddmap{\glsentryuseri}{\Glsentryuseri}
\newcommand*\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newcommand*\Glsentryuserii}[1]{%
  \@Glsentryfield{#1}{userii}%
}
\glsmfuaddmap{\glsentryuserii}{\Glsentryuserii}
\newcommand*\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newcommand*\Glsentryuseriii}[1]{%
  \@Glsentryfield{#1}{useriii}%
}
\glsmfuaddmap{\glsentryuseriii}{\Glsentryuseriii}
\newcommand*\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newcommand*\Glsentryuseriv}[1]{%
  \@Glsentryfield{#1}{useriv}%
}
\glsmfuaddmap{\glsentryuseriv}{\Glsentryuseriv}
\newcommand*\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
}

```



```

\newcommand*{\Glsentryuser}[1]{%
  \@Glsentryfield{#1}{user}%
}
\glsmfuaddmap{\Glsentryuser}{\Glsentryuser}
\newcommand*{\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\Glsentryuseriv}[1]{%
  \@Glsentryfield{#1}{useriv}%
}
\glsmfuaddmap{\Glsentryuseriv}{\Glsentryuseriv}
\newcommand*{\Glsentryshort}[1]{\@Gls@entry@field{#1}{short}}
\newcommand*{\Glsentryshorti}[1]{%
  \@Glsentryfield{#1}{shorti}%
}
\glsmfuaddmap{\Glsentryshorti}{\Glsentryshorti}
\newcommand*{\Glsentryshortpl}[1]{\@Gls@entry@field{#1}{shortpl}}
\newcommand*{\Glsentryshortpl}[1]{%
  \@Glsentryfield{#1}{shortpl}%
}
\glsmfuaddmap{\Glsentryshortpl}{\Glsentryshortpl}
\newcommand*{\Glsentrylong}[1]{\@Gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylongi}[1]{%
  \@Glsentryfield{#1}{longi}%
}
\glsmfuaddmap{\Glsentrylongi}{\Glsentrylongi}
\newcommand*{\Glsentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuaddmap{\Glsentrylongpl}{\Glsentrylongpl}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\Glsentryshort{#1})%
}
\newcommand*{\Glsentryfulli}[1]{%
  \Glsentrylong{#1} (\Glsentryshort{#1})%
}
\glsmfuaddmap{\Glsentryfulli}{\Glsentryfulli}
\newcommand*{\Glsentryfullpl}[1]{%
  \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\Glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\Glsentryshortpl{#1})%
}
\glsmfuaddmap{\Glsentryfullpl}{\Glsentryfullpl}
\newcommand*{\Glsentrynumberlist}[1]{%
  \Glsdoifexists{#1}%
  {%
    \@Gls@entry@field{#1}{numberlist}%
  }%
}

```

```

}
\@ifpackageloaded{hyperref} {%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
{%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \bgroup
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \let\@org@glsnumberformat\glsnumberformat
        \def\glsnumberformat##1{##1}%
        \protected@edef\the@numberlist{%
          \csname glo@\@glo@label @numberlist\endcsname}%
        \def\@gls@numlist@sep{%
          \def\@gls@numlist@nextsep{}%
          \def\@gls@numlist@lastsep{}%
          \def\@gls@thislist{}%
          \def\@gls@donext@def{}%
          \renewcommand\do[1]{%
            \protected@edef\@gls@thislist{%
              \@gls@thislist
              \noexpand\@gls@numlist@sep
              ##1%
            }%
            \let\@gls@numlist@sep\@gls@numlist@nextsep
            \def\@gls@numlist@nextsep{\glsnumlistsep}%
            \@gls@donext@def
            \def\@gls@donext@def{%
              \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
            }%
          }%
          \expandafter \glsnumlistparser \expandafter{\the@numberlist}%
          \let\@gls@numlist@sep\@gls@numlist@lastsep
          \@gls@thislist
        \egroup
      }%
    }
  }
\newcommand*\glsnumlistsep}{, }
\newcommand*\glsnumlistlastsep}{ \& }
\newcommand*\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%

```

```

\def\@glo@label{#2}%
\@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd[2] [] {%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\glsmfuexcl{\glsadd}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}
\newrobustcmd*\glsaddall[1] [] {%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore[1]{}
\edef\glsopenbrace{\expandafter@gobble\string\{ }
\edef\glsclosebrace{\expandafter@gobble\string\} }
\edef\glsbackslash{\expandafter@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter}{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter}{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter}{#1}}

```

```

    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsxindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsxindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoxywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
  \def\writeist{%
    \ifundef{\glswrite}{\newwrite\glswrite}{}%
    \@gls@addpredefinedattributes
    \openout\glswrite=\istfilename
    \write\glswrite{;; xindy style file created by the glossaries
      package}%
    \write\glswrite{;; for document '\jobname' on
      \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\@xdystyle:=\@xdyrequiredstyles\do{%
      \ifx\@xdystyle\@empty

```

```

\else
  \protected@write\glswrite{}{(require
    \string"\xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
  ; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}{(define-location-class
    \string"\@gls@classI\string"^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"

```

```

        :close \string"{}\string"}}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{ }\string\providecommand*%
      \expandafter\string
      \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
      {%
        \string\setentrycounter
        [\expandafter\@gobble\string\#1]{\@this@ctr}%
        \expandafter\string
        \csname\@this@attr\endcsname
        {\expandafter\@gobble\string\#2}%
      }%
    }%
  }%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string"\glspercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffiX
\@onelevel@sanitize\gls@suffiXF

```

```

\ifx\gls@suffiF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffiFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  '\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string

```

```

    \glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
    \end{theglossary}\string\glossarypostamble\string\n
    \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
    \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
    \string"\string\relax \string\glsresetentrylist\string\n
    \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
    \string"\string\relax \string\glsresetentrylist\string\n
    \string"}
\write\glswrite{delim_0 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
    \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
    \string"\string}\string\relax
    \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"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}}
\@onlypremake\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%

```

```

\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
  \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
    \noexpand\def\noexpand\@gls@checkedmkidx{%
      \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
      #1#1\noexpand\null
      \noexpand\expandafter\noexpand\@gls@updatechecked
      \noexpand\@gls@checkedmkidx{####1}%
      \noexpand\def\noexpand\@gls@checkedmkidx{%
        \noexpand\expandafter\noexpand\@gls@checkesquote####1\noexpand\@nil
        \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
        \noexpand\null
        \noexpand\expandafter\noexpand\@gls@updatechecked
        \noexpand\@gls@checkedmkidx{####1}%
        \noexpand\def\noexpand\@gls@checkedmkidx{%
          \noexpand\expandafter\noexpand\@gls@checkesactual####1\noexpand\@nil
          \noexpand\?\noexpand\?\noexpand\null
          \noexpand\expandafter\noexpand\@gls@updatechecked
          \noexpand\@gls@checkedmkidx{####1}%
          \noexpand\def\noexpand\@gls@checkedmkidx{%
            \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
            \noexpand?\noexpand?\noexpand\null
            \noexpand\expandafter\noexpand\@gls@updatechecked
            \noexpand\@gls@checkedmkidx{####1}%
            \noexpand\def\noexpand\@gls@checkedmkidx{%
              \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
              \noexpand|\noexpand|\noexpand\null
              \noexpand\expandafter\noexpand\@gls@updatechecked
              \noexpand\@gls@checkedmkidx{####1}%
              \noexpand\def\noexpand\@gls@checkedmkidx{%
                \noexpand\expandafter\noexpand\@gls@checkeschar####1\noexpand\@nil
                \noexpand\|\noexpand\|\noexpand\null
                \noexpand\expandafter\noexpand\@gls@updatechecked
                \noexpand\@gls@checkedmkidx{####1}%
                \noexpand\def\noexpand\@gls@checkedmkidx{%
                  \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
                  \noexpand!\noexpand!\noexpand\null
                  \noexpand\expandafter\noexpand\@gls@updatechecked
                  \noexpand\@gls@checkedmkidx{####1}%
                  \noexpand\def\noexpand\@gls@checkedmkidx{%
                    \noexpand\expandafter\noexpand\@gls@checkescllevel####1\noexpand\@nil
                    \noexpand\!\noexpand\!\noexpand\null
                    \noexpand\expandafter\noexpand\@gls@updatechecked
                    \noexpand\@gls@checkedmkidx{####1}%
                    \noexpand\def\noexpand\@gls@checkedmkidx{%
                      \noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
                      \@backslashchar\@backslashchar\noexpand\null
                      \noexpand\expandafter\noexpand\@gls@updatechecked
                      \noexpand\@gls@checkedmkidx{####1}%
                    }%
                  }%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}

```

```

}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
  \noexpand\def\noexpand\@gls@checkescquote####1%
    \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
    ####3\noexpand\null{%
      \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
      \noexpand\toks@={####1}%
      \noexpand\ifx\noexpand\null####2\noexpand\null
        \noexpand\ifx\noexpand\null####3\noexpand\null
          \noexpand\edef\noexpand\@gls@checkedmkidx{%
            \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
          \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
          \noexpand\else
            \noexpand\edef\noexpand\@gls@checkedmkidx{%
              \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
              \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                \csname#1\endcsname}\noexpand\@gls@quotechar
                \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                  \csname#1\endcsname}\noexpand\@gls@quotechar}%
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####3\noexpand\null}%
            \noexpand\fi
          \noexpand\else
            \noexpand\edef\noexpand\@gls@checkedmkidx{%
              \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
              \noexpand\@gls@quotechar\noexpand\string
                \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
            \noexpand\ifx\noexpand\null####3\noexpand\null
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
                \expandonce{\csname#1\endcsname}\noexpand\null}%
            \noexpand\else
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
                ####3\noexpand\null}%
            \noexpand\fi
          \noexpand\fi
        \noexpand\@gls@checkescquote
      }%
    }%
  }%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%

```

```

\@gls@addpredefinedattributes
\let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
\GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
}
\newcommand*{\@makeglossary}[1]{%
\ifglossaryexists{#1}%
{%
\ifglssavewrites
\expandafter\newtoks\csname glo@#1@filetok\endcsname
\else
\expandafter\newwrite\csname glo@#1@file\endcsname
\expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
\fi
\@gls@renewglossary
\writeist
}%
{%
\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@glsopenfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglsexindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%
\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space

```

```

        \@gls@codepage
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}
}
\else
\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-1 }}%
        {\let\@gls@order\@empty}%
        \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
            -s \listfilename\space
            -t \jobname.\csuse{@glotype@#1@log}
            -o \jobname.\csuse{@glotype@#1@in}
            \jobname.\csuse{@glotype@#1@out}}}%
        }%
        \@gls@dothiswrite
    }%
    {
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \gls@xdy@set@lang{#1}%
            \ifdefvoid\gls@codepage
            {\let\@gls@codepage\@empty}%
            {\def\@gls@codepage{-C \gls@codepage\space}}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
                -I xindy
                \@gls@order
                -L \@gls@langmod\space
                -M \gls@istfilebase\space

```

```

        \@gls@codepage
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-1 }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\@warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@types}{\@glo@types}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{%
{\@gls@automake@immediate{\@gls@type}}%
}%
}%
\glsautomakefalse

```

```

\renewcommand*{\@gls@doautomake}{}%
\else
\ifdefempty\gls@automake@makegloss{%
{%
\IfFileExists{\jobname.aux}%
{\immediate\write18{\gls@automake@makegloss\space \jobname}}{%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
}%
\fi
}
\newcommand*{\makeglossaries}{%
\ifglxindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef\glswrite{\newwrite\glswrite}{%
\protected@write\@auxout{\string\providecommand\string\@glsorder[1]}{}}
\protected@write\@auxout{\string\providecommand\string\@istfilename[1]}{}}
\ifundef\@gls@extramakeindexopts
}{%
{%
\protected@write\@auxout{\string\providecommand
\string\@gls@extramakeindexopts[1]}{}}
\protected@write\@auxout{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}}{}}{}}{%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][{}]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser

```

```

        {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodedeflistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
        \protected@edef\gls@tmp{\@gls@automake@types}%
        \@for\@gls@type:=\gls@tmp\do{%
            \ifdefempty{\@gls@type}{%
                {\@gls@automake{\@gls@type}}%
            }%
        }%
    }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprntglossary{%
    \ifdefstring{\@glo@types}{,}%
    {%
        \GlossariesWarningNoLine{No glossaries have been defined}%
    }%
    {%
        \GlossariesWarningNoLine{No \string\printglossary\space
            or \string\printglossaries\space
            found. ^^J(Remove \string\makeglossaries\space if you
            don't want any glossaries.) ^^JThis document will not
            have a glossary}%
    }%
}%
\AtEndDocument{%
    \warn@nomakeglossaries
    \warn@noprntglossary
}
\newcommand*\makenoidxglossaries{%
    \def\glsindexingsetting{noidx}%
    \@domakeglossaries
    {%
        \renewcommand{\@gls@noref@warn}[1]{%
            \GlossariesWarning{Empty glossary for
                \string\printnoidxglossary[type={##1}].
                Rerun may be required (or you may have forgotten to use
                commands like \string\gls)}%
        }%
        \let\@gls@checkmkidxchars\@gobble
        \glsesclocationsfalse
        \let\@do@wrglossary\gls@noidxglossary
    }%
}

```



```

\let@gls@getgrouptitle@gls@noidx@getgrouptitle
\let@gls@checkseeallowed@relax
\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef@gls@label{\glsdetoklabel{##1}}%
  \protected@write@auxout{}{%
    \string@gls@reference
      {\csname glo@%gls@label @type\endcsname}%
      {\@gls@label}%
      {%
        \string@glsseeformat##2}%
      }%
  }%
}%
\AtBeginDocument
{%
  \write@auxout{\string\providecommand\string@gls@reference[3]{}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries@relax
\let\makeglossaries@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let@glo@assign@sortkey@glo@assign@sortkey
  \def@glo@sorttype@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
  \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%

```

```

\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries

```

```

\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type=#1] work}%
}
\newcommand*\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@gls@locref}}%
    }%
  }%
}
\AtEndDocument{%
  \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]{%
  \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
\ifglsesclocations
  \@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*\gls@addprotectedpagefmt}[1]{%
  \ifglsesclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gls#1\endcsname}%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}
\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\gls@addprotectedpagefmt\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{\theglslentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglslentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph
    \let\gls@orgRoman\@Roman
    \ifglswrallowprimitivemods
      \def\gls@the##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
      \def\the{\expandafter\gls@the}%
      \def\gls@number##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
      \def\number{\expandafter\gls@number}%
      \def\romannumeral##1{%
        \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
    \fi
    \def\@arabic##1{%
      \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
    \def\@Roman##1{%

```

```

        \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
    \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
    \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\gls@locref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\@xp@gls@getcounterprefix{\@gls@locref}{\@gls@Hlocref}%
\fi
\protected@edef\@gls@label{\gls@detoklabel{#1}}%
\@do@@wrglossary
}
\newcommand*{\@do@@wrglossary}{%
\ifglxindy
\expandafter\@glo@check@mkidxrangechar\@gls@numberformat\@nil
\def\@glo@range{%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgroup
\gls@wrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%

```

```

        \noexpand\egroup
        \noexpand\@gls@getcounterprefix{#1}{#2}%
    }%
    \do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
    \def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
    \ifx\@gls@thisloc\@gls@thisHloc
        \def\@glo@counterprefix{}%
    \else
        \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
            \def\@glo@tmp{##2}%
            \ifx\@glo@tmp\@empty
                \def\@glo@counterprefix{}%
            \else
                \def\@glo@counterprefix{##1}%
            \fi
        }%
        \@gls@get@counterprefix#2.#1\end@getprefix
        \ifx\@glo@counterprefix\@empty
            \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
                prefixing^^Jlocation ‘#1’. You need to modify the
                definition of \string\theH\@gls@counter^^Jotherwise you
                will get the warning: "‘name{\@gls@counter.#1}’ has been^^J
                referenced but does not exist"%
        \fi
    \fi
}
\newcommand{\@do@seeglossary}[2]{%
    \def\@gls@xref{#2}%
    \@onelevel@sanitize\@gls@xref
    \@gls@checkmkidxchars\@gls@xref
    \ifglsxindy
        \gls@glossary{\csname glo@#1@type\endcsname}{%
            (indexentry
                :key (\csname glo@#1@index\endcsname)
                :xref (\string"\@gls@xref\string")
                :attr \string"see\string"
            )
        }%
    \else
        \gls@glossary{\csname glo@#1@type\endcsname}{%
            \string\glossaryentry{\csname glo@#1@index\endcsname
                \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
        \fi
    }
}
\def\@gls@fixbraces#1#2#3\@nil{%
    \if#2[\relax
        \@gls@fixbraces#1#2#3\@end@fixbraces
    \else

```



```

    \def#1{#2#3}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee[3][\seename]{%
  \@do@seeglossary{#2}{#1}{#3}}
\newcommand*\@glssee[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat[1]{%
  \ifglshashshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
\newcommand*\gls@save@numberlist[1]{%
  \ifglssavenumberlist
    \toks@{#1}%
    \protected@edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{\@do@writeaux@info}%
  \fi
}
\newcommand*\warn@noprntglossary{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored[2]{%
  \@ifglossaryexists{#1}%

```

```

{#2}%
{%
  \ifignoredglossary{#1}%
  {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary ‘\@glo@type’. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{\@glo@type}%
}

```

```

{%
\ifx\glossarytitle\org@glossarytitle
\else
\expandafter\let\csname @glotype@\@glo@type @title\endcsname
\glossarytitle
\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\let\nopostdesc\@nopostdesc
\gls@dotoc@title
\@glossarystyle
\let\gls@org@glossaryentryfield\glossentry
\let\gls@org@glossarysubentryfield\subglossentry
\renewcommand{\glossentry}[1]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
\gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%
\@gls@preglossaryhook
#2%
}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
\makeatletter
\@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
\IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
{}%
{\null}%
\ifglsxindy
\gls@xdy@set@lang{\@glo@type}%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\@gls@langmod}}%
}%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%

```

```

        \string\providecommand\string\@gls@codepage[2]{}%
        \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\gls@codepage}}%
    }%
}%
\do@auxoutstuff
\fi
\renewcommand*\@warn@nomakeglossaries{%
    \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*\@glo@sortentries[2]{%
    \glosortentrieswarning
    \def\@glo@sortinglist{}%
    \def\@glo@sortinghandler{#1}%
    \protected@edef\@glo@type{#2}%
    \forlistcsloop{\@glo@do@sortentries}{\glsref@#2}%
    \csdef{\glsref@#2}{%
    \@for\@this@label:=\@glo@sortinglist\do{%
        \xifinlistcs{\@this@label}{\glsref@#2}%
        {}%
        {%
            \listcsxadd{\glsref@#2}{\@this@label}%
        }%
        \ifcsdef{\@glo@sortingchildren@\@this@label}%
        {%
            \@glo@addchildren{#2}{\@this@label}%
        }%
        {}%
    }%
}%
}
\newcommand*\@glo@addchildren[2]{%
    \bgroup
    \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
    \@for\@this@childlabel:=\@glo@childlist\do
    {%
        \xifinlistcs{\@this@childlabel}{\glsref@#1}%
        {}%
        {%
            \listcsxadd{\glsref@#1}{\@this@childlabel}%
        }%
        \ifcsdef{\@glo@sortingchildren@\@this@childlabel}%
        {%
            \@glo@addchildren{#1}{\@this@childlabel}%
        }%
        {}%
    }%
}%
\egroup

```

```

}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{glo@sortingchildren@\@glo@parent}{}%
    }%
    {}%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{\glsref{\@glo@type}}%
    {%
    }%
    {%
      \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
  }%
  {}%
  \@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtlinsertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce@gls@sort@B}%
    {\expandonce@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce@gls@sort@B}%
    {\expandonce@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%

```

```

        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@word}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=word]}{-%
    }%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=letter]}{-%
    }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \ifcsdef{\@glo@sorthandler@\glsorder}%
        {%
            \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
        }%
        {%
            \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{-%
        }%
    }%
    {%
}

```

```

    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=standard]}{}}%
  }%
}
\newcommand*\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=case]}{}}%
  }%
}
\newcommand*\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}}%
  }%
}
\newcommand*\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \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}%
      {%
        \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{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%

```

```

\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{'sort' key not permitted with
  \string\printglossary}%
  {The 'sort' key may only be used with \string\printnoidxglossary}%
}
\newcommand*\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}
\newcommand*\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*\glsnonextpages}{}
\newcommand*\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglsstepsubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglsstepsubentry}%
  \fi

```

```

}
\newcommand*\glsrefentry}[1]{%
\ifglentrycounter
\ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
\ifglssubentrycounter
\ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
\gls{#1}%
\fi
\fi
}
\newcommand*\glsentrycounterlabel}{%
\ifglentrycounter
\theglossaryentry.\space
\fi
}
\newcommand*\glssubentrycounterlabel}{%
\ifglssubentrycounter
\theglossarysubentry)\space
\fi
}
\newcommand*\glsentryitem}[1]{%
\ifglentrycounter
\glsstepentry{#1}\glsentrycounterlabel
\else
\glsresetsubentrycounter
\fi
}
\newcommand*\glssubentryitem}[1]{%
\ifglssubentrycounter
\glsstepsubentry{#1}\glssubentrycounterlabel
\fi
}
\ifcsundef{theglossary}%
{%
\newenvironment{theglossary}{-}{-}%
}%
{%
\@gls@warnontheGLOSSDEFINED
\renewenvironment{theglossary}{-}{-}%
}
\newcommand*\glossaryheader{}
\newcommand*\glsstarget}[2]{\@glsstarget{\glo@linkprefix#1}{#2}}
\newcommand*\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*\glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
\expandafter\glsnamefont\expandafter{\glo@name}%

```

```

    }%
  }
  \newcommand*{\Glossentryname}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsnamefont{\Glsentryname{#1}}%
    }%
  }
  \glsmfuaddmap{\glossentryname}{\Glossentryname}
  \newcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrydesc{#1}%
    }%
  }
  \newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
  \newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*{\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
  \newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
  \newcommand*{\glsgroupskip}{=}
  \newcommand*{\glsgroupheading}[1]{=}
  \newrobustcmd*{\glsgetgrouptitle}[1]{%
    \@gls@getgrouptitle{#1}{\@gls@grptitle}%
    \@gls@grptitle
  }
  \ExplSyntaxOn
  \newcommand*{\@gls@getgrouptitle}[2]{%
    \def#2{#1}%
    \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
    { \ifcsundef{#1groupname}{-}{\letcs#2{#1groupname}} }
  }
  \ExplSyntaxOff
  \newcommand*{\@gls@noidx@getgrouptitle}[2]{%

```

```

\DTLifint{#1}%
{\edef#2{\char#1\relax}}%
{%
  \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
}%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { \glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { \glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][ ]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%

```

```

    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\xp@glsnumber{#1}}
}
\newrobustcmd{\xp@glsnumber}[1]{%
  \glsnumber#1\nohyperpage{}}\@nil
}
\def\@glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
    \else
      \gls@delimR#1\delimR\delimR\%
    \fi
  \ifx\#2\%
    \else
      #2%
    \fi
  \ifx\#3\%
    \else
      \glsnumber#3\@nil
    \fi
}
\def\gls@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
  \gls@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil
}

```



```

\endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else
\bgroup
\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\glsentrycounter\@glo@counterprefix
\glswrglosslocationtarget{\@gls@x}}%
\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{\noexpand\glswrglosslocationtextfmt{\the\toks@}}%
}%
\@gls@x
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm}[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf}[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt}[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf}[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd}[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit}[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl}[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup}[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc}[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph}[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\@gls@label]{%
\def\@gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
\ifcsundef{xspace}%
{%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
}%
}%
}%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
\noexpand\gls{#1}\noexpand\xspace}%
}%
}

```

```

}%
}
\newcommand{\newacronym}[4][\%
\gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand{\gls@newacronym}[4]{%
\renewcommand{\gls@newacronym}[4]{%
\PackageError{glossaries}{incompatible acronym style:
rollback may be required}%
{something has gone wrong with \string\newacronym. Try
\string\usepackage{glossaries}[=v4.49]}%
}%
\setacronymstyle{long-short}%
\newacronym[#1]{#2}{#3}{#4}%
}
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
\acrfullformat{#1}{#3}{#4}[#5]}{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace{#2}}
\newrobustcmd*{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
\acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
\acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
\acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
\acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
\acrlinkfullformat{\@Acrlongpl}{\@Acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
\acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*{ns@acrfull}[2][\%
\new@ifnextchar[{\@acrfull{#1}{#2}}%
{\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%

```

```

\acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][{}]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][{}]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}[]}%
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][{}]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2}[]}%
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok

```

```

\newtoks\glslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\glspluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genacrfullformat{##2}{##3}}}%
}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genplacrfullformat{##2}{##3}}}%
}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%

```

```

\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{##1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
}%
}%
}
\newcommand*{\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*{\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%

```

```

{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglsashaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
}

```

```

\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglsahaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{
\GlsUseAcrEntryDispStyle{short-long}%
}%
{
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{
\GlsUseAcrEntryDispStyle{short-long}%
}%
{
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{
\GlsUseAcrEntryDispStyle{long-short}%
}%
{
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
}

```



```

\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext

```

```

{%
  \ifglshaslong{\glslabel}%
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \glsentrylongpl{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylongpl{\glslabel}\glsinsert
      }%
      {%
        \glsupercase
        {\glsentrylongpl{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \glsentrylong{\glslabel}\glsinsert
    }%
    {%
      \Glsentrylong{\glslabel}\glsinsert
    }%
    {%
      \glsupercase
      {\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}{%
      \glsupercase{\glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
    }%
  }%
  \renewcommand*{\acrfullplfmt}[3]{%

```

```

\glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\glsentryfull}[1]{%
\glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
\Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
\glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\Glsentrylong{##1}}%
}

```

```

}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%
\renewcommand*\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymfont}[1]{\textsc{##1}}%

```

```

\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
\let\acs\acrshort
\let\Acs\Acrshort
\let\acsp\acrshortpl
\let\Acsp\Acrshortpl

```

```

\let\acl\acrlong
\let\aclp\acrlongpl
\let\Acl\Acrlong
\let\Aclp\Acrlongpl
\let\acf\acrfull
\let\acfp\acrfullpl
\let\Acf\Acrfull
\let\Acfp\Acrfullpl
\let\ac\gls
\let\Ac\Gls
\let\acp\glspl
\let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
  \footnote{\glslink[#1]{#2}{#3}}%
}
\newcommand*{\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}[=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@glstetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
}
\newcommand*{\showglofirst}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showglotype}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouseri}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@useri\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showglongame}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@desc\endcsname
}
}

```

```

\newcommand*\showgloDESCplural}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@DESCplural\endcsname
}
\newcommand*\showgloSort}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Sort\endcsname
}
\newcommand*\showgloSymbol}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Symbol\endcsname
}
\newcommand*\showgloSymbolplural}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Symbolplural\endcsname
}
\newcommand*\showgloShort}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Short\endcsname
}
\newcommand*\showgloLong}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Long\endcsname
}
\newcommand*\showgloIndex}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Index\endcsname
}
\newcommand*\showgloFlag}[1]{%
  \expandafter\show\csname ifglo@glSDetoklabel{#1}@Flag\endcsname
}
\newcommand*\showgloLoclist}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Loclist\endcsname
}
\newcommand*\showgloField}[2]{%
  \cSShow{glo@glSDetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show\@glSACronymlists
}
\newcommand*\showglossaries}{%
  \show\@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @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 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*\glsymbolnav{%
  \glsnavhyperlink{glsymbols}{\glsgetgrouptitle{glsymbols}}%
  \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{\glsentencecase}[1]{\makefirstuc{#1}}
\ExplSyntaxOn
\newcommand{\glslowercase}[1]{ \text_lowercase:n { #1 } }
\newcommand{\glsuppercase}[1]{ \text_uppercase:n { #1 } }
\ifdef\MFUsentencecase
{%
  \newcommand{\glsmfuexcl}[1]{\MFUexcl{#1}}
  \newcommand{\glsmfublocker}[1]{\MFUblocker{#1}}
  \newcommand{\glsmfuaddmap}[2]{\MFUaddmap{#1}{#2}}
}
{
  \RequirePackage{textcase}
  \providecommand{\MFUsentencecase}[1]{ \text_titlecase_first:n { #1 } }
  \providecommand{\glsmfuexcl}[1]{
    \tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
    {
      \tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
    }
  }
  \renewcommand*\mfirstucMakeUppercase{\MakeTextUppercase}
  \newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
  \newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}
\RequirePackage{datatool-base}[=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\gls@showtarget\@gobble
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
  \or
    % debug=false
    \@gls@debugfalse
    \let\gls@showtarget\@gobble
    \let\gls@showaccsupp\@gobblethree
    \PackageInfo{glossaries}{debug mode OFF}%
  \or
    % debug=showtargets
    \@gls@debugtrue
    \renewcommand*\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
  }%

```

```

    }%
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \renewcommand{\@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}{seenoindex}{%
  [\gls@seenoindex@val\gls@seenoindex@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoindex@nr
    \renewcommand*{\@gls@see@noindex}{%
      \PackageError{glossaries}%
      {'\gls@xr@key' key may only be used after \string\makeglossaries\space
      or \string\makenoidxglossaries}%
      {You must use \string\makeglossaries\space
      or \string\makenoidxglossaries\space before defining
      any entries that have a '\gls@xr@key' key}%
    }%
  \or
    \renewcommand*{\@gls@see@noindex}{%
      \GlossariesWarning{'\gls@xr@key' key ignored}%
    }%
  \or
    \renewcommand*{\@gls@see@noindex}{}%
  \fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@@glossarysec}{section}}%
  {\newcommand*{\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*{\@@glossarysec}{#1}}
\newcommand*{\@@glossarysecstar}{*}
\newcommand*{\@@glossaryseclabel}{}
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
  false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax

```

```

\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{}%
\or
\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{}%
\or
\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{%
\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*\@glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
\renewcommand*\@glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
\renewcommand*\@glo@seeautonumberlist}{%
\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationfalse
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
\@gls@declareoption{noalong}{\renewcommand*\@gls@loadlong}{}}
\IfFileExists{supertabular.sty}{%
\newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}{%
\newcommand*\@gls@loadsuper}{}}
\@ifpackageloaded{xtab}{\renewcommand*\@gls@loadsuper}{}}{}
\@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper}{}}{}
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}}
\@gls@declareoption{no list}{%
\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@indexing@setting}{\ifglsubentrycounter\else makeindex\fi}
\newcommand*{\@gls@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
  \renewcommand*{\@gls@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandard@sort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@gls@check@sort@allowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@gls@storeentry{\@gls@storeentry}%
  \def\@gls@defsortcount##1{}%
  \def\@gls@defsort##1##2{%
    \ifx\@gls@sort\@gls@default@sort
      \let\@gls@sort\@gls@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort
    \glsprestandard@sort{\@gls@sort}{##1}{##2}%
    \expandafter\protected\xdef\csname glo@##2@sort\endcsname{\@gls@sort}%
  }%
  \def\@gls@setsort##1{}%
  \let\@gls@check@sort@allowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\gls@sort@number@fmt[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]%
  }
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*\@acronymtype{acronym}%
    \newcommand*\gls@tr@set@acronym@toctitle{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*\@gls@acronymlists{}
\newcommand*\@addtoacronymlists[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*\DeclareAcronymList[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\SetAcronymLists[1]{%
  \renewcommand*\@gls@acronymlists{#1}%
}

```

```

}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

}
\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\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@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}}
\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}{%
  \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\@gls@labelsfile
\immediate\openout\@gls@labelsfile=\jobname.glslabels
\protected@edef\@gls@fulllist{\@gls@types}%
\ifdefempty{\@ignored@glossaries}{%
{\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
\forallglsentries[\@gls@fulllist]{\@glsentry}%
}%
\ifdefempty{\@glsentry}{%
{%
\ifx\do@write@glslabelsname\@gobble
\def\@gls@thisentryname{%

```

```

        \else
            \letcs{\@gls@thisentryname}{glo@\glsdetoklabel{\@glsentry}@name}%
            \@onelevel@sanitize\@gls@thisentryname
        \fi
        \immediate\write\@gls@labelsfile
            {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
    }%
}%
\immediate\closeout\@gls@labelsfile
}
\newcommand*\do@write@glslabelsname[1]{
\begingroup
\catcode'\^^I12
\gdef\@write@glslabelsname#1{\^^I#1}
\endgroup
\@gls@declareoption{writeglslabels}{\do@write@glslabels
\let\do@write@glslabelsname\gobble}
\@gls@declareoption{writeglslabelnames}{\do@write@glslabels
\let\do@write@glslabelsname\write@glslabelsname}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\newcommand{\gls@automake@makegloss}{}
\newcommand{\gls@automake@option}{automake=\gls@automake@val}
\newcommand*\@gls@do@automake@err}{%
\PackageError{glossaries}{You must use
\string\makeglossaries\space with \gls@automake@option}
{%
    Either remove the \gls@automake@option\space setting or
    add \string\makeglossaries\space to your document preamble.%
}%
}%
}%
\define@choicekey{glossaries.sty}{automake}%
[\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
\ifcase\gls@automake@nr\relax
\GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or

```

```

\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
\let\@gls@doautomake\@gls@do@automake@err
\else
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{%
\AtEndDocument{\@gls@doautomake}
@gls@declareoption{automakegloss}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakegloss}%
\def\gls@automake@val{makegloss}%
\def\gls@automake@nr{3}%
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\let\@gls@doautomake\@gls@do@automake@err
}
@gls@declareoption{automakeglosslite}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%
\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\@gls@doautomake\@gls@do@automake@err
}
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
\ifgls savewrites
\renewcommand*{\glswritefiles}{\@glswritefiles}%
\else
\let\glswritefiles\@empty
\fi
}
\gls 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@do@symbolsdef]{\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*\@gls@makefirstuc}[1]{\MFUsentencecase{#1}}%
    \or
      % unexpanded
      \renewcommand*\@gls@makefirstuc}[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*\@glssymbolsgroupname{Symbols}
\providecommand*\@glsnumbersgroupname{Numbers}
\newcommand*\@glspluralsuffix{s}
\newcommand*\@glsacrpluralsuffix{\@glspluralsuffix}
\newcommand*\@glsupacrpluralsuffix{\glstextup{\@glsacrpluralsuffix}}
\providecommand*\@seename{see}
\providecommand*\@andname{\&}
\newcommand*\@RequireGlossariesLang[1]{%
  \@ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\@ProvidesGlossariesLang[1]{%
  \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\@addglossarytocaptions[1]{}
\ifglstranslate

```

```

\RequirePackage{tracklang}[2019/11/30]
\ifvoid\@gls@requested@languages{%
{%
  \for\@gls@thislang:=\@gls@requested@languages\do{%
    \TrackIfKnownLanguage{\@gls@thislang}{}%
    {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
  }%
}%
\@gls@usetranslator
\@ifpackageloaded{translator}
{%
  \ifboolexpr
  {
    test {\ifdefstring{\trans@languages}{English}}
    and not
    test {\ifdefstring{bbl@loaded}{english}}
  }
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \usedictionary{glossaries-dictionary}%
    \renewcommand*{\addglossarytocaptions}[1]{%
      \ifcsundef{captions#1}{}%
      {%
        \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
        \expandafter\toks@\expandafter{\@gls@tmp
          \renewcommand*{\glossaryname}{\translate{Glossary}}}%
        }%
        \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
      }%
    }%
  }%
}%
\}
\}
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
\}
\}

```

```

\glsifusetranslator
{%
  \renewcommand*\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
      {%
        \csuse{gls@tr@set@#1@toctitle}%
      }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glsymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
  }{%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{\nopostdesc}
\newcommand*\glspar{\par}
\newcommand*\setStyleFile[1]{%
  \renewcommand*\gls@istfilebase}{#1}%
  \ifglxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase{\jobname}
\newcommand*\@istfilename[1]{%
\newcommand*\glscompositor}{.}

```

```

\newcommand*\glsSetCompositor}[1]{%
  \renewcommand*\glscompositor{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphacompositor{\glscompositor}
\ifglsxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*\gls@suffixF{}
\newcommand*\glsSetSuffixF[1]{%
  \renewcommand*\gls@suffixF{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*\gls@suffixFF{}
\newcommand*\glsSetSuffixFF[1]{%
  \renewcommand*\gls@suffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glsnumberformat[1]{#1}%
}%
{%
  \newcommand*\glsnumberformat[1]{\glshypernumber{#1}}%
}
\newcommand{\glslocationstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@\currentglossary}}
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*\glossarypostamble{}
\newcommand*\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@pglossarysection{#1}{#2}%
  }%
  \gls glossarymark{\glossarytoctitle}%
}

```

```

\ifcsundef{glossarymark}%
{%
  \newcommand{\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@docclearpage{%
\ifthenelse{\equal{\@@glossarysec}{chapter}}%
{%
\ifcsundef{cleardoublepage}%
{%
\clearpage
}%
{%
\ifcsdef{if@openright}%
{%
\if@openright
\cleardoublepage
\else
\clearpage
\fi
}%
{%
\cleardoublepage
}%
}%
}%
}%
}
\newcommand*\glscclearpage{\gls@docclearpage}
\newcommand*\@gls@toc}[2]{%
\ifglstoc
\ifglsnumberline
\addcontentsline{toc}{#2}{\protect\numberline{#1}}%
\else
\addcontentsline{toc}{#2}{#1}%
\fi
\fi
}
\newcommand*\glsnoxindywarning}[1]{%
\GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
\GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
\edef\@xdyattributes{\string"default\string"}%
\fi

```

```

\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]{%
    \glsnoxindywarning\GlsAddXdyAttribute
  }
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
  can't be used after \string\GlsAddXdyAttribute}{Move all
  occurrences of \string\GlsAddXdyCounters\space before the first
  instance of \string\GlsAddXdyAttribute}%
}
\ifglxindy
  \newcommand*\@glsaddxdyattribute[2]{%
    \edef\xdyattributes{\@xdyattributes ^^J
      \string"#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{hyper rm}
  \GlsAddXdyAttribute{hyper sf}
  \GlsAddXdyAttribute{hyper tt}
  \GlsAddXdyAttribute{hyper bf}
  \GlsAddXdyAttribute{hyper md}
  \GlsAddXdyAttribute{hyper it}
}

```

```

\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
\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"glxnumbers\string"^^J\space\space\space
        :prefixes (\string"0\string" \string"1\string"
          \string"2\string" \string"3\string" \string"4\string"
          \string"5\string" \string"6\string" \string"7\string"
          \string"8\string" \string"9\string")^^J\space\space\space
          \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
  \fi
  \newcommand*\GlsAddLetterGroup[2]{%
    \expandafter\toks@\expandafter{\@xdylettergroups}%
    \protected@edef\@xdylettergroups{\the\toks@^^J%
      (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

```

```

}

\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forallglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forallglsentries[\@@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}

```

```

}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}%
    #3%
  }%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \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}%
  {%
    \ifcsemtyp{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsemtyp{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%

```

```

    {#2}%
    {#3}%
  }
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@glstdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
    {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@glstdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
    {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@glstdetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
  {%
    \ifdefequal\@glo@short\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
}
}
\newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@glstdetoklabel{#2}@#1}{#3}{#4}}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glstdetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        }%
      }%
    \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*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
    \protected@edef\@glo@tmp{#1}%
    \ifx\@glo@tmp\@gls@defaulttype
        \GlossariesWarning{\string\defglsentryfmt\space attempting to set
            a display style for \string\glsdefaulttype\space but there
            doesn't seem to be a default glossary. If you have used
            'nomain', you need to define an alternative glossary}%
    \else
        \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
    \fi
}
}

```

```

\newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@forbidtext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{%
\newcommand*\@newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*\s@newglossary}[2]{%
\ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*\ns@newglossary}[5][glg]{%
\doifglossarynoexistsordo{#2}%
{%
\ifundef\glsdefaulttype
{%
\gdef\glsdefaulttype{#2}%
}{}%
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}{\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
\defglsentryfmt[#2]{\glsentryfmt}%
}%
}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
{\@gls@setcounter{#2}[\glscounter]}%
}%
}%
\gls@gobbleopt
}%

```

```

}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{, #1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist{#1}%
  }%
  {%
    \protected@eappto\@gls@nohyperlist{, #1}%
  }%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%

```

```

\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key{3}{%
\renewcommand*\@gls@xr@key{#1}%
}

```

```

\gls@checkseeallowed
\def#2{#3}%
\@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
\@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
\GlossariesWarning
{'\gls@xr@key' key doesn't have any effect when used in the document
environment. Move the definition to the preamble
after \string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
\def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
[\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]}%
{f%
\ifcase\@gls@nonumberlist@nr\relax
\def\@glo@prefix{\@gls@nonextpages}%
\@gls@savenonumberlist{true}%
\else
\def\@glo@prefix{\@gls@nextpages}%
\@gls@savenonumberlist{false}%
\fi
}
\newcommand*{\@gls@noidx@nonumberlist}[1]{%
\cslet\@glo@\@glo@label @prenumberlist}\@glo@prefix
}%
\newcommand*\@gls@noidx@prenumberlist}[1]{%
\csuse\@glo@\@gls@detoklabel{#1}@prenumberlist}}
\newcommand*{\@gls@savenonumberlist}[1]{}
\newcommand*{\@gls@initnonumberlist}{}%
\newcommand*{\@gls@storenonumberlist}[1]{}
\newcommand*{\@gls@enablesavenonumberlist}{%
\renewcommand*{\@gls@initnonumberlist}{%
\undef\@glo@nonumberlist
}%
\renewcommand*{\@gls@savenonumberlist}[1]{%
\def\@glo@nonumberlist{##1}%
}%
\renewcommand*{\@gls@storenonumberlist}[1]{%
\ifdef\@glo@nonumberlist
{f%
\cslet\@glo@\@gls@detoklabel{##1}@nonumberlist}\@glo@nonumberlist}%
}%
{}%
}%

```

```

\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \iflssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'}.

```

```

    Maybe makeglossaries + rerun required}%
\else
\PackageError{glossaries}%
{Package option 'savenumberlist=true' required}%
{%
    You must use the 'savenumberlist' package option
    to reference location lists.%
}%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}
}
}
}

```

```

}%
{%
  \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
  {%
    \csuse{gls@assign@#3@field}{#2}{#4}%
  }%
}%
}%
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \@@gls@expand@field{#2}{#3}{#1}%
  }%
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\@gls@expandfields{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*\@gls@noexpandfields{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*\@gls@defdocnewglossaryentry{%
  \let\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’)}%
    }%
  }%
}
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs){%
  \@gls@enablesavenonumberlist
  \edef\@gls@restreat{\noexpand\catcode‘\noexpand\@=\number\catcode‘\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{}{%
  \@gls@restreat
  \undef\@gls@restreat
  \gls@defdocnewglossaryentry
}

```

```

\newcommand*\glsdefs@newdocentry}[2]{%
  \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
      (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
      \gls@glsdefs@addlabel{#1}%
      \gls@defglossaryentry{#1}{#2}%
    }%
}
\newcommand*\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}@\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%
        \glspercentchar
      }%
    }%
  }%
  {}%
}%
\gls@writedefhook
\immediate\write\@gls@deffile
{%
  \glspercentchar^^J%
  \expandafter\@gobble\string\}\glspercentchar^^J%
}%
}
\newcommand*\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%

```

```

{user1}{useri},%
{user2}{userii},%
{user3}{useriii},%
{user4}{useriv},%
{user5}{userv},%
{user6}{uservi},%
{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
      {%
        \edef#1{\expandafter\@secondoftwo\@gls@map}%
        \@endfortrue
      }%
    }%
  }%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  }%
  }%
  \@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  }%
}

```

```

    \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newcommand*\glsaddkey{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*\@sglsaddkey[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  }%
  {}%
  \@glsaddkey{#1}%
}
\newcommand*\@glsaddkey[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
    \newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
    \ifcsdef{@gls@user@#1@}%
    {%
      \PackageError{glossaries}%
      {Can’t define ‘\string#5’ as helper command
        ‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
      }%
    }%
    {%
      \expandafter\newcommand\expandafter*\expandafter
      {\csname @gls@user@#1\endcsname}[2] [] {%
        \new@ifnextchar [%
          {\csuse{@gls@user@#1@}{##1}{##2}}%
          {\csuse{@gls@user@#1@}{##1}{##2} [] }%
        \csdef{@gls@user@#1@}##1##2[##3]{%
          \@gls@field@link{##1}{##2}{#3{##2}##3}%
        }%
        \newrobustcmd*{#5}{%
          \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
        }%
        \ifcsdef{@Gls@user@#1@}%
        {%
          \PackageError{glossaries}%
          {Can’t define ‘\string#6’ as helper command

```

```

        '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
    }%
}%
{
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
\csdef{@Gls@user@#1@}##1##2[##3] {%
  \gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
  '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  {}%
}%
{
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
\csdef{@GLS@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1@\endcsname}%
}%
\glsmfuaddmap{#3}{#4}%
\glsmfuaddmap{#5}{#6}%
\glsmfublocker{#7}%
}%
{
  \PackageError{glossaries}{Key '#1' already exists}{}%
}%
}
\newrobustcmd{\glsfieldxdef}[3] {%
\glsdoifexists{#1}%
{
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {
    \protected@csxdef{glo@\@glo@label @#2}{#3}%
  }%
}
}

```

```

    {%
      \PackageError{glossaries}{Field '#2' doesn't exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfielddedf}[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{\glsfieldddf}[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}%
  }
}
\@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\@glsnoname
  \let\@glo@desc\@glsnodesc
  \let\@glo@descplural\@gls@default@value
  \let\@glo@type\@gls@default@value
  \let\@glo@symbol\@gls@default@value
  \let\@glo@symbolplural\@gls@default@value
  \let\@glo@text\@gls@default@value
  \let\@glo@plural\@gls@default@value
  \let\@glo@first\@gls@default@value
  \let\@glo@firstplural\@gls@default@value
  \let\@glo@sort\@gls@default@value
  \let\@glo@counter\@gls@default@value
  \def\@glo@see{}%
  \def\@glo@parent{}%
  \def\@glo@prefix{}%
  \@gls@initnonumberlist
  \def\@glo@useri{}%

```



```

\def@glo@userii{%
\def@glo@useriii{%
\def@glo@useriv{%
\def@glo@userv{%
\def@glo@uservi{%
\def@glo@short{%
\def@glo@shortpl{%
\def@glo@long{%
\def@glo@longpl{%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{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
  \ifdequal@glo@label@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
\fi

```

```

}%
{%
\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{@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\@glsnname
  \@glsnname
  \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}}%
}%
}%
\fi
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnname
  \@glsnname
  \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{,%
\for@gl@sentries[\@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}%
}
\ifglxsindy
\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\@cgl@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cglformat{##2}{##3}%
\glsunset{##2}%
\else
\@gls@{##1}{##2}[##3]%
\fi
}%
\def\@cgl@pl@##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@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsetused{\@glsentry}%
        {\immediate\write\@auxout
            {\string\@gls@entry@count{\@glsentry}{\glsetentrycurrcount{\@glsentry}}}}%
        {}%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cglset}{\@gls@hyp@opt\@cglset}
\newcommand*{\@cglset}[2][{}]{%

```

```

    \new@ifnextchar[{\@cgl@s@{#1}{#2}}{\@cgl@s@{#1}{#2} []}%
}
\def\@cgl@s@#1#2[#3]{%
  \GlossariesWarning{\string\cgl@s\space is defaulting to
    \string\gl@s\space since you haven't enabled entry counting}%
  \@cgl@s@{#1}{#2}[#3]%
}
\newcommand*{\cgl@sformat}[2]{%
  \ifglshaslong{#1}{\gl@sentrylong{#1}}{\gl@sentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gl@s@hyp@opt\@cGls}
\glsmfuaddmap{\cgl@s}{\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*{\cgl@spl}{\@gl@s@hyp@opt\@cgl@spl}
\newcommand*{\@cgl@spl}[2] []{%
  \new@ifnextchar[{\@cgl@spl@{#1}{#2}}{\@cgl@spl@{#1}{#2} []}%
}
\def\@cgl@spl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl@spl\space is defaulting to
    \string\gl@spl\space since you haven't enabled entry counting}%
  \@gl@spl@{#1}{#2}[#3]%
}
\newcommand*{\cgl@splformat}[2]{%
  \ifglshaslong{#1}{\gl@sentrylongpl{#1}}{\gl@sentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gl@s@hyp@opt\@cGlspl}
\glsmfuaddmap{\cgl@spl}{\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] [\@gl@s@default]{%
  \let\@gl@s@default\gl@sdefaulttype

```

```

\def\glsdefaulttype{#1}\input{#2}%
\let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glsformat[1]{#1}
\newcommand*\glsentryfmt{\glsentryfmt}
\newcommand*\glsentryfirstfmt{\glsentryfirstfmt}
\ifdefempty\glsentryfirstfmt
{
\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*{\glsnacfmt}{%
  \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\glsentencecase\expandafter{\gls@text}%
}

```

```

\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
  \ifKV@glslink@local
  \let\@gls@do@glsunset\glslocalunset
  \else
  \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\glslinkvar}[3]{#1}
\newcommand*\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree

```

```

\@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][{}]{%
\let\glslinkvar\@thirdofthree
\@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
\@gls@hyp@opt\@gls@link
}
\newcommand*{\@gls@link}[3][{}]{%
\glsdoifexistsordo{#2}%
{%
\let\do@gls@link@checkfirsthyper\relax
\@gls@link[#1]{#2}{#3}%
}%
\glstextformat{#3}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
\@gls@hyp@opt\@Gls@link
}
\newcommand*{\@Gls@link}[3][{}]{%
\@gls@link[#1]{#2}{\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{\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\glsstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper@ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\@do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\glsformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\glsformat{#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@mkidxrangear#3\@nil
    \protected@edef#1{%
        \@glo@prefix setentrycounter[#4]{#2}%
        \expandafter\string\csname\@glo@suffix\endcsname
    }%
    \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1(\relax
    \def\@glo@prefix{(%}
    \if\relax#2\relax
        \def\@glo@suffix{glsnumberformat}%
    \else
        \def\@glo@suffix{#2}%
    \fi
\else
    \if#1)\relax
        \def\@glo@prefix{)%}
    \if\relax#2\relax
        \def\@glo@suffix{glsnumberformat}%
    \else
        \def\@glo@suffix{#2}%
    \fi
\else
    \def\@glo@prefix{)\def\@glo@suffix{#1#2}%

```

```

\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
  \def\@gls@checkedmkidx{}%
  \let\gls@xdystring=#1\relax
  \@onelevel@sanitize\gls@xdystring
  \edef\do@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
    \@backslashchar\@backslashchar\noexpand\null}%
  \do@gls@xdycheckbackslash
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
  \def\@gls@checkedmkidx{}%
  \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
  \ifgls@esclocations
    \for\@gls@tmp:=\gls@protected@pagefmts\do
    {%
      \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
      \@onelevel@sanitize\@gls@sanitized@tmp
      \edef\gls@dostubst{%
        \noexpand\DTLsubstituteall\noexpand\gls@xdystring
        {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
      }%
      \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"\@gls@quotechar}%
\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@checkeschar{\@gls@checkeschar#2\|\|\@null}%
\else
\def\@@gls@checkeschar{\@gls@checkeschar#2\|#3\@null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkescllevel#1\!#2\!#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescllevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\\"\@gls@levelchar
\@gls@quotechar\string\\"\@gls@levelchar}%
\def\@@gls@checkescllevel{\@gls@checkescllevel#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\\"\@gls@levelchar}%
\ifx\@null#3\@null
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!\!\@null}%
\else
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!#3\@null}%
\fi
\fi
\@gls@checkescllevel
}
\def\@gls@checkbar#1|#2|#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@@gls@checkbar{\@gls@checkbar#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

    \@gls@quotechar\@gls@encapchar}%
  \ifx\null#3\null
    \def\@gls@checkbar{\@gls@checkbar#2|\null}%
  \else
    \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
  \fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checklevel{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
      \def\@gls@checklevel{\@gls@checklevel#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@levelchar}%
    \ifx\null#3\null
      \def\@gls@checklevel{\@gls@checklevel#2!\null}%
    \else
      \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
    \fi
  \fi
  \@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else

```

```

        \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
        \fi
        \fi
        \@gls@checkactual
    }
    \bgroup
    \catcode'\|0\relax
    \catcode'\|12\relax
    \gdef\@gls@checkesc#1\#2\#3\null{%
        \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
        \toks@={#1}%
        \ifx\null#2\null
            \ifx\null#3\null
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                \def\@gls@checkesc{\relax}%
            \else
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                    \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
                \def\@gls@checkesc{\@gls@checkesc#3\null}%
            \fi
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \@gls@quotechar\@backslashchar}%
            \ifx\null#3\null
                \def\@gls@checkesc{\@gls@checkesc#2\| \null}%
            \else
                \def\@gls@checkesc{\@gls@checkesc#2\#3\null}%
            \fi
        \fi
        \@gls@checkesc
    }
    \egroup
    \def\@gls@xdycheckquote#1"#2"#3\null{%
        \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
        \toks@={#1}%
        \ifx\null#2\null
            \ifx\null#3\null
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                \def\@gls@xdycheckquote{\relax}%
            \else
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                    \string"\string"}%
                \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
            \fi
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \string"}%
            \ifx\null#3\null
                \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
            \else

```



```

        \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
        \fi
        \fi
    \@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*{\glsdohypertarget}[2]{%
\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{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@#1}{#2}]{\@gls@#1}{#2}[{}]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
  \new@ifnextchar[{\@Gls@#1}{#2}]{\@Gls@#1}{#2}[{}]}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }%
}

```

```

\def\@glo@text{\csname gls@\gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2][{}]{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}]%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@scaps@case\@thirdofthree
\let\gls@customtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\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\gls@scaps@case\@firstofthree
\let\gls@customtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\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@ifplural\@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*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
\gl@mf@blocker{\GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \gl@do@ifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@ifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@do@ifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@ifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gl@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}]{%
  \@gl@disp[#1]{#2}{\gl@sentencecase{#3}}%
}
\gl@mf@uaddmap{\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*{\gls@text}{\@gls@hyp@opt\@gls@text}
\newcommand*{\@gls@text}[2] []{%
  \new@ifnextchar[{\@gls@text@{#1}{#2}}{\@gls@text@{#1}{#2} []}]
\def\@gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\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}{\glsuppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Gls@text}{\@gls@hyp@opt\@Gls@text}
\glsmfuaddmap{\gls@text}{\Gls@text}
\newcommand*{\@Gls@text}[2] []{%
  \new@ifnextchar[{\@Gls@text@{#1}{#2}}{\@Gls@text@{#1}{#2} []}]
\def\@Gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*{\gls@first}{\@gls@hyp@opt\@gls@first}
\newcommand*{\@gls@first}[2] []{%
  \new@ifnextchar[{\@gls@first@{#1}{#2}}{\@gls@first@{#1}{#2} []}]
\def\@gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Gls@first}{\@gls@hyp@opt\@Gls@first}
\glsmfuaddmap{\gls@first}{\Gls@first}
\newcommand*{\@Gls@first}[2] []{%
  \new@ifnextchar[{\@Gls@first@{#1}{#2}}{\@Gls@first@{#1}{#2} []}]
\def\@Gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLS@first}{\@gls@hyp@opt\@GLS@first}
\glsmfublocker{\GLS@first}
\newcommand*{\@GLS@first}[2] []{%
  \new@ifnextchar[{\@GLS@first@{#1}{#2}}{\@GLS@first@{#1}{#2} []}]
\def\@GLS@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\gls@plural}{\@gls@hyp@opt\@gls@plural}

```

```

\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*{\glsymbol}{\@gls@hyp@opt\@glsymbol}

```



```

\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*{\@GLssymbol}[2] [] {%
  \new@ifnextchar [{\@GLssymbol@{#1}{#2}}{\@GLssymbol@{#1}{#2} []}]
\def\@GLssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\glsmfublocker{\GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
  \new@ifnextchar [{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
\def\@GLSsymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
\newcommand*{\@glsymbolplural}[2] [] {%
  \new@ifnextchar [{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2} []}]
\def\@glsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfuaddmap{\glsymbolplural}{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar [{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar [{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar [{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def\@glsuseri@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\glsmfuaddmap{\glsuseri}{\Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar [{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
\newcommand*{\@GLSuseri}[2][\%
    \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}]
\def\@GLSuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][\%
    \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}]
\def\@glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\glsmfuaddmap{\glsuserii}{\Glsuserii}
\newcommand*{\@Glsuserii}[2][\%
    \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}]
\def\@Glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2][\%
    \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}]
\def\@GLSuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2][\%
    \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}]
\def\@glsuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2][\%
    \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}]
\def\@Glsuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2][\%
    \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}]
\def\@GLSuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}

```

```

\newcommand*{\@glsuseriv}[2] []{%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2] []{%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2] []{%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] []{%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2] []{%
  \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] []{%
  \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\glsmfuaddmap{\glsuservi}{\Glsuservi}
\newcommand*{\@Glsuservi}[2] []{%
  \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}]
\def\@Glsuservi@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar [{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
\def \@GLSuservi@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar [{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
}
\def \@acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\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}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar [{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]
}
\def \@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\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}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%

```

```

\new@ifnextchar [{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
\new@ifnextchar [{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsentryshortpl{#2}}#3}%
}%
\@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\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@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@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\glsmfublocker{\ACRlong}
\newcommand*{\ns@ACRlong}[2][{}]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@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{%
      \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]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
\newcommand*{\glsentryname}[1]{\@Gls@entry@field{#1}{name}}
\newcommand*{\Glsentryname}[1]{%
  \glstexorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{name}}}%
}
\glsmfuexcl{\Glsentryname}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newrobustcmd*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \protected@edef\@glo@text{\@Gls@entry@field{#1}{name}}%
    \expandafter\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\glssentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc[1]{\@gls@entry@field{#1}{desc}}
\newcommand*\Glsentrydesc[1]{\@Glsentryfield{#1}{desc}}
\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}%
}
}

```

```

\glsmfueycl{\Glsentryshortpl}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfueycl{\Glsentrylong}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfueycl{\Glsentrylongpl}
\newcommand*{\glsentryfull}[1]{%
  \glsentrylong{#1} (\glsentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\glsentryshort{#1})%
}
\glsmfueycl{\Glsentryfull}
\newcommand*{\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}
\glsmfueycl{\Glsentryfullpl}
\newcommand*{\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
\@ifpackageloaded{hyperref} {%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
\newcommand*{\glsdisplaynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \bgroup
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \let\@org@glsnumberformat\glsnumberformat
    \def\glsnumberformat##1{##1}%
  }%
}

```

```

\protected@edef\the@numberlist{%
  \csname glo@\glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{%
\def\@gls@numlist@nextsep{%
\def\@gls@numlist@lastsep{%
\def\@gls@thislist{%
\def\@gls@donext@def{%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*\glsnumlistsep{, }
\newcommand*\glsnumlistlastsep{ \& }
\newcommand*\gls hyperlink}[2][\glsentrytext{\glo@label}]{%
  \def\glo@label{#2}%
  \@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd}[2][]{%
  \@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
  \ifglsxindy
    \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
  \fi
  \newcommand*\GlsSetXdyNumberGroupOrder}[1]{%
    \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
  }
  \ifglsxindy
    \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
      \renewcommand*\@xdynumbergrouporder}{#1}%
    }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

        \renewcommand*{\@xdynumbergrouporder}{#1}%
        \@onelevel@sanitize\@xdynumbergrouporder
    }
\else
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
        \glsnoxywarning\GlsSetXdyNumberGroupOrder}
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
        \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsxindy
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \renewcommand*{\@glsminrange}{#1}}
\else
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
    \def\writeist{%
        \ifundef{glswrite}{\newwrite\glswrite}{}%
        \@gls@addpredefinedattributes
        \openout\glswrite=\istfilename
        \write\glswrite{;; xindy style file created by the glossaries
            package}%
        \write\glswrite{;; for document '\jobname' on
            \the\year-\the\month-\the\day}%
        \write\glswrite{^^J; required styles^^J}
        \@for\@xdystyle:=\@xdyrequiredstyles\do{%
            \ifx\@xdystyle\@empty
            \else
                \protected@write\glswrite}{(require
                    \string"\@xdystyle.xdy\string")}%
            \fi
        }%
        \write\glswrite{^^J%
            ; list of allowed attributes (number formats)^^J}%
        \write\glswrite{(define-attributes ((\@xdyattributes)))}%
        \write\glswrite{^^J; user defined alphabets^^J}%
        \write\glswrite{\@xdyuseralphabets}%
        \write\glswrite{^^J; location class definitions^^J}%
        \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
            \protected@write\glswrite}{(define-location-class
                \string"\@gls@classI\string"^^J\space\space\space
                (
                    :sep "{ }{"
                    \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
                    :sep "}"
                )
                ^^J\space\space\space

```



```

:min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}\{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{}\{\string\providecommand*%
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
        {%
          \string\setentrycounter
            [\expandafter\@gobble\string\#1]{\@this@ctr}%
          \expandafter\string

```

```

\csname\@this@attr\endcsname
  {\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string" \glpercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string{\string
\glossaryentrynumbers\string{\string\relax \string"}
\write\glswrite{delim_1 \string"\string{\string

```

```

    \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_2 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string"\string\}\string}\string}
\write\glswrite{delim_n \string"\string\delimN \string}
\write\glswrite{delim_r \string"\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
    \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
    \string"\string}\string\relax
    \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string"glssymbols\string}
\write\glswrite{numhead_positive \string"glnumbers\string}
\write\glswrite{page_compositor \string"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}{
\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#####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\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@checkescllevel####1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote####1%
\expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%

```

```

        \noexpand\def\noexpand\@gls@checkescquote{%
          \noexpand@gls@checkescquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \noexpand\@gls@quotechar\noexpand\string
  \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
  \expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
  ####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
  \GlossariesWarning{too late to use \string\noist. The style file
  has already been written}%
}
\newcommand*\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
    \ifglssavewrites
      \expandafter\newtoks\csname glo@#1@filetok\endcsname
    \else
      \expandafter\newwrite\csname glo@#1@file\endcsname
      \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%

```

```

\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@glsopenfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglxindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%
\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
\@gls@codepage
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order

```



```

        -s \istfilename\space
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \gls@xdy@set@lang{#1}%
            \ifdefvoid\gls@codepage
            {\let\@gls@codepage\@empty}%
            {\def\@gls@codepage{-C \gls@codepage\space}}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
                -I xindy
                \@gls@order
                -L \@gls@langmod\space
                -M \gls@istfilebase\space
                \@gls@codepage
                -t \jobname.\csuse{@glotype@#1@log}
                -o \jobname.\csuse{@glotype@#1@in}
                \jobname.\csuse{@glotype@#1@out}}%
            }%
            \@gls@dothiswrite
        }%
        {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
            doesn't exist. Rerun may be required}}%
    }%
    {
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%

```

```

\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-1 }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@gloftype@#1@log}
-o \jobname.\csuse{@gloftype@#1@in}
\jobname.\csuse{@gloftype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gloftype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@types}{\@glo@types}
\newcommand{\@gls@@automake@immediate}{}%
\ifnum\gls@automake@nr=2\relax
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\else
\ifdefempty\gls@automake@makegloss{}%
{%
\IfFileExists{\jobname.aux}%
{\immediate\write18{\gls@automake@makegloss\space \jobname}}{}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
}%
\fi
}
\newcommand*{\makeglossaries}{}%
\ifglxsindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
{%
\@gls@@automake@immediate

```

```

\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}
\ifundef\@gls@extramakeindexopts
{%
}%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{}}
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{equal{\@glo@type}{}}{}}{%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}}%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}

```

```

\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*\makenoidxglossaries{%
  \def\glsindexingsetting{noidx}%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\@gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
          {\csname glo@\@gls@label @type\endcsname}%
          {\@gls@label}%
          {%
            \string\glsseeformat##2}%
          }%
      }%
    }%
  }%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J

```

```

    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
  defined in the preamble^^Jwhen you use
  \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
  \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
      a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%

```

```

    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc@glsnoidxdisplayloc
  \let\@gls@org@glsseeformat@glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={##1}] work}%
}

```

```

}
\newcommand*\gls@noidxglossary{%
  \protected@write\auxout{\gls@wrglossdisablelocationcmds}{%
    \string\gls@reference
    {\csname glo@\gls@label @type\endcsname}%
    {\gls@label}%
    {\string\glsnoidxdisplayloc
     {\glo@counterprefix}%
     {\gls@counter}%
     {\glsnumberformat}%
     {\gls@locref}}%
    }%
  }%
}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*\gls@writefiles{%
  \forallglossaries{\glo@type}{%
    \ifcsundef{glo@\glo@type @filetok}%
    {%
      \def\gls@tmp{}}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
        \else
          \GlossariesWarningNoLine{Glossary '@glo@type' has no
            entries}%
        \fi
      \else
        \glsopenfile{\gls@write}{\glo@type}%
        \immediate\write\gls@write{%
          \expandafter\the
            \csname glo@\glo@type @filetok\endcsname}%
        \immediate\closeout\gls@write
        \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{\gls@writeentry{#1}{\@do@wrglossary{#1}}}%
}
\newcommand*\@gls@writeentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*\@gls@disablepagerefexpansion{%
  \gls@wrglossdisablelocationcmds
  \ifglsesclocations
    \@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{\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}{%
  \ifglxindy
    \expandafter\glo@check@mkidxrangear\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
    \fi
    \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 location ‘#1’. You need to modify the
  definition of \string\theH\@gls@counter otherwise you
  will get the warning: “name{\@gls@counter.#1}’ has been
  referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:key (\csname glo@#1@index\endcsname)
:xref (\string\@gls@xref\string)
:attr \string"see\string"
)
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee}[3][\seename]{%
\@do@seeglossary{#2}{[#1]{#3}}}
\newcommand*\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat}[3][\seename]{%
\emph{#1} \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@noprintglossary{}%
\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}{\endgroup}
\expandafter\endgroup\if@endpe\@doendpe\fi
}
\newcommand{\@print@glossary}{%
    \makeatletter
    \input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
    \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
    {}%
    {\null}%
    \ifglsxindy
        \gls@xdy@set@lang{\@glo@type}%
        \edef\@do@auxoutstuff{%
            \noexpand\AtEndDocument{%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\providecommand\string\@xdylanguage[2]{}%
                }%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\@xdylanguage{\@glo@type}{\@gls@langmod}%
                }%
            }%
        }%
        \do@auxoutstuff
        \edef\@do@auxoutstuff{%
            \noexpand\AtEndDocument{%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\providecommand\string\@gls@codepage[2]{}%
                }%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\@gls@codepage{\@glo@type}{\@gls@codepage}%
                }%
            }%
        }%
        \do@auxoutstuff
    \fi
\renewcommand*{\@warn@nomakeglossaries}{%
    \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
    \glosortentrieswarning
    \def\@glo@sortinglist{}%
    \def\@glo@sortinghandler{#1}%

```

```

\protected@edef\@glo@type{#2}%
\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
    \%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \%glo@addchildren{#2}{\@this@label}%
  }%
  }%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {%
      \%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \%glo@addchildren{#1}{\@this@childlabel}%
    }%
    }%
  }%
  \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    }%
  \expandafter\@glo@sortedinsert
  \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
  \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
  {%
  }%
  }%
}

```



```

        \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
}%
{%
    \glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}
\newcommand*\@glo@sortedinsert}[2]{%
    \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*\@glo@sorthandler@word}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*\@glo@sorthandler@letter}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*\@glo@sorthandler@case}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
}

```

```

\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=word]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\cuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}}%
  }%
}

```

```

\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=nocase]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listead{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
      }%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}}%
    }%
  \glossarysection[\glossarytoctitle]{\glossarytitle}%
  \glossary preamble
  \def\@gls@currentlettergroup{}%
}

```

```

\begin{theglossary}%
\glossaryheader
\glsresetentrylist
\forlistcsloop{\@gls@noidx@do}{\@gls@ref@{\@glo@type}%
\end{theglossary}%
\glossarypostamble
}%
{%
\@gls@noref@warn{\@glo@type}%
}%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\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}%
  {%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}%
}%
\letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
\expandafter\glo@grabfirst\@gls@sort{}{}\@nil
\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%
  \ifdefempty{\@gls@currentlettergroup}{}%
  {%
    \global\let\@glo@thislettergrp\@glo@thislettergrp
    \glsgroupskip
  }%
  \glsgroupheading{\@glo@thislettergrp}%
}%
\global\let\@gls@currentlettergroup\@glo@thislettergrp
\ifdefvoid{\@gls@loclist}
{%
  \glossentry{#1}{}%
}%
{%
  \glossentry{#1}%
  {%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}%
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

```

    }%
}
\newcommand*\glsnoidxdisplayloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{%
    \ifinlistcs{#2}{\@glsref@#1}%
    }{%
    \listcsgadd{\@glsref@#1}{#2}%
    \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
    {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
    }{%
    \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
    }%
  }
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@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{\setglossarystyle{#1}}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%

```

```

{false,nolabel,autolabel,nameref}[nolabel]%
{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*{\@@glossarysecstar}{*}%
    \renewcommand*{\@@glossaryseclabel}{}%
  \or
    \renewcommand*{\@@glossarysecstar}{}%
    \renewcommand*{\@@glossaryseclabel}{}%
  \or
    \renewcommand*{\@@glossarysecstar}{}%
    \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*{\@@glossarysecstar}{*}%
    \renewcommand*{\@@glossaryseclabel}{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{\glsautoprefix\@glo@type}}%
  \fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glsesubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%

```

```

        \glsresetentrylist
    }%
}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \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
\glssstepsentry{#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@\glsdetoklabel{#1}@name}%
\expandafter\glsnamefont\expandafter{\glo@name}%
}%
}
\newcommand*{\Glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{
\glsnamefont{\Glsentryname{#1}}%
}%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
\newcommand*{\glossentrydesc}[1]{%
\glsdoifexistsorwarn{#1}%
{
\glsentrydesc{#1}%
}%
}
\newcommand*{\Glossentrydesc}[1]{%

```

```

\glsdoifexistsorwarn{#1}%
{%
  \Glsentrydesc{#1}%
}%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
\newcommand*{\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newrobustcmd*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
  \def#2{#1}%
  \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
  { \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} }
}
\ExplSyntaxOff
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { \glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { \glsnumbers }
    { #1 }
  }
}
}

```

```

\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][{}]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\@xp@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
}
\def\gls@delimR#1\delimR #2\delimR #3\{%
  \ifx\#2\%
  \gls@delimN{#1}%
  \else
    \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
  \fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\{%
  \ifx\#3\%
  \gls@numberlink{#1}%
  \else
    \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
  \fi
}
\def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil
  \endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
  \toks@=\expandafter{\the\toks@#1}%
  \ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
  \else
  \bgroup
  \gls@wrglossdisableanchorcmds
  \protected@edef\@gls@tmp{\gls@entrycounter\@glo@counterprefix
  \gls@wrglosslocationtarget{\@gls@x}}%
  \@onelevel@sanitize\@gls@tmp
  \edef\@gls@x{\noexpand\egroup
  \noexpand\hyperlink{\@gls@tmp}%
  {\noexpand\gls@wrglosslocationtextfmt{\the\toks@}}%

```

```

    }%
    \@gls@x
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand\oldacronym[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
  }%
}%
\ifcsundef{xspace}%
{%
  \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[4]{#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}{%
    \glsuppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields{\description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*\acronymsort}[2]{#1}
\newcommand*\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{%
  }%
  }%
  \ifdefempty{@glsacronymlists}%
  {%
    \DeclareAcronymList{\acronymtype}%
  }%
}

```



```

    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \for\@gls@type:=\@glsacronymlists\do{%
        \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
}%
}
\newcommand*\newacronymstyle}[3]{%
    \ifcsdef{@glsacr@dispstyle@#1}%
    {%
        \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
    }%
    {%
        \csdef{@glsacr@dispstyle@#1}{#2}%
        \csdef{@glsacr@styledefs@#1}{#3}%
    }%
}
\newcommand*\renewacronymstyle}[3]{%
    \ifcsdef{@glsacr@dispstyle@#1}%
    {%
        \csdef{@glsacr@dispstyle@#1}{#2}%
        \csdef{@glsacr@styledefs@#1}{#3}%
    }%
    {%
        \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
    }%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
    \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
    \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
    \renewcommand*\genacrfullformat}[2]{%
        \glsentrylong{##1}##2\space
        (\protect\firstacronymfont{\glsentryshort{##1}})%
    }%
    \renewcommand*\Genacrfullformat}[2]{%
        \Glsentrylong{##1}##2\space
        (\protect\firstacronymfont{\glsentryshort{##1}})%
    }%
    \renewcommand*\genplacrfullformat}[2]{%
        \glsentrylongpl{##1}##2\space
        (\protect\firstacronymfont{\glsentryshortpl{##1}})%
    }%
    \renewcommand*\Genplacrfullformat}[2]{%
        \Glsentrylongpl{##1}##2\space

```

```

(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\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}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{%

```

```

\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
\ifdefempty\glscustomtext
{%
\ifglshaslong{\glslabel}%
{%
\glsifplural
{%
\glscapscase
{%
\glsentrylongpl{\glslabel}\glsinsert
}%
{%
\Glsentrylongpl{\glslabel}\glsinsert
}%
{%
\glsupercase
{\glsentrylongpl{\glslabel}\glsinsert}%
}%
}
}
}
}
}
}

```

```

}%
{%
  \glscapscase
  {%
    \glentrylong{\glslabel}\glsinsert
  }%
  {%
    \Glsentrylong{\glslabel}\glsinsert
  }%
  {%
    \glsupercase
    {\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}{%
      \glsupercase{\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}{%
      \glsupercase{\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}%
{%
    \ifglsashaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
    \glshyperfirstfalse
    \renewcommand*\genacrfullformat[2]{%
        \protect\firstacronymfont{\glsentryshort{##1}}##2%
        \protect\footnote{\glsentrylong{##1}}%
    }%
    \renewcommand*\Genacrfullformat[2]{%
        \firstacronymfont{\Glsentryshort{##1}}##2%
        \protect\footnote{\glsentrylong{##1}}%
    }%
    \renewcommand*\genplacrfullformat[2]{%
        \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
        \protect\footnote{\glsentrylongpl{##1}}%
    }%
    \renewcommand*\Genplacrfullformat[2]{%
        \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
        \protect\footnote{\glsentrylongpl{##1}}%
    }%
    \renewcommand*\acronymentry[1]{\acronymfont{\glsentryshort{##1}}}%
    \renewcommand*\acronymsort[2]{##1}%
    \renewcommand*\acronymfont[1]{##1}%
    \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
    \renewcommand*\acrfullfmt[3]{%
        \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
            (\glsentrylong{##2})}%
    }%

```

```

\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\Glsentrylong{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\Glsentryshort{##2}}##3\space
  (\Glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\Glsentrylongpl{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\Glsentrylongpl{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\Glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\Glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\Glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\Glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\Glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\Glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\Glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%

```



```

{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
  \newacronymstyle{footnote-sc-desc}%
  {%
    \GlsUseAcrEntryDispStyle{footnote-sc}%
  }%
  {%
    \GlsUseAcrStyleDefs{footnote-sc}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
    }
    \newacronymstyle{footnote-sm-desc}%
    {%
      \GlsUseAcrEntryDispStyle{footnote-sm}%
    }%
    {%
      \GlsUseAcrStyleDefs{footnote-sm}%
      \renewcommand*{\GenericAcronymFields}{}%
      \renewcommand*{\acronymsort}[2]{##2}%
      \renewcommand*{\acronymentry}[1]{%
        \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
      }
    }
  \newcommand*{\DefineAcronymSynonyms}{%
    \let\acs\acrshort
    \let\Acs\Acrshort
    \let\acsp\acrshortpl
    \let\Acsp\Acrshortpl
    \let\acl\acrlong
    \let\aclp\acrlongpl
    \let\Acl\Acrlong
    \let\Aclp\Acrlongpl
    \let\acf\acrfull
    \let\acfp\acrfullpl
    \let\Acf\Acrfull
    \let\Acfp\Acrfullpl
    \let\ac\gls
    \let\Ac\Gls
    \let\acp\glspl
    \let\Acp\Glspl
  }
  \ifglsacrshortcuts
    \DefineAcronymSynonyms
  \fi

```

```

\newcommand*\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\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*\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*\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@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*\showglooclist}[1]{%
\expandafter\show\csname glo@glstdetoklabel{#1}@oclist\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
}
}

```

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{\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{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\ifpackageloaded{doc}%
{%
  \@gls@docloadedtrue
}%
{%
  \ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
}
\if@gls@docloaded
\let\glsorg@theglossary\theglossary
\let\glsorg@endtheglossary\endtheglossary
\let\glsorg@PrintChanges\PrintChanges
\renewcommand{\PrintChanges}{%
  \begingroup
    \let\theglossary\glsorg@theglossary
    \let\endtheglossary\glsorg@endtheglossary
    \glsorg@PrintChanges
  \endgroup
}
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}{\gls@debug@val\gls@debug@nr}%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\gls@debug@nr\relax
    % debug=true

```

```

\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\let\@glsshowtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
  \glsshowtargetinner{#1}%
\else
  \ifinner
    \glsshowtargetinner{#1}%
  \else
    \glsshowtargetouter{#1}%
  \fi
\fi
}

```

```

\newcommand*\glsshowtargetfonttext}[1]{%
  \ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbolsymbol{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{}
\newcommand*\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{}
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoinde@nr
  \renewcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key}%
  }%
}

```

```

\or
  \renewcommand*\@gls@see@noindex}{%
    \GlossariesWarning{'\gls@xr@key' key ignored}%
  }%
\or
  \renewcommand*\@gls@see@noindex}{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec}{section}}%
  {\newcommand*\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec{#1}}
\newcommand*\@@glossarysecstar}{*}%
\newcommand*\@@glossaryseclabel}{%
\newcommand*\glsautoprefix{}%
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
    \label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*\glossaryentrynumbers[1]{#1\gls@save@numberlist{#1}}

```



```

\@gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\glo@seeautonumberlist}{%
    \def\glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\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}[=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*\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{%
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{%
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
}%
}

```

```

\glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
\ifglentrycounter
\ifundef\c@glossaryentry
{%
\ifx\@gls@counterwithin\@empty
\newcounter{glossaryentry}%
\else
\newcounter{glossaryentry}[\@gls@counterwithin]%
\fi
\def\theH@glossaryentry{\currentglossary.\theglossaryentry}%
}%
}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\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
\ifglentrycounter
\newcounter{glossarysubentry}[glossaryentry]%
\else
\newcounter{glossarysubentry}%
\fi
\def\theH@glossarysubentry{\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]{%
  \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*{\@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}{%
  \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
}
\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}{%
\ifvoid\@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

```

```

        \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]{-}
\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\@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
}
\gls savewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
    \ifbool{glscompatible-3.07}%
    {%
        \@gls@removedoption{compatible-3.07}%
    }%
    {}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
    \ifbool{glscompatible-2.07}%
    {%
        \@gls@removedoption{compatible-2.07}%
    }%
    {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
    \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
    \gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}%
    [\@gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
{%
    \ifcase\gls@kernelglossredefs@nr\relax
        \def\glossary{\@gls@override@glossary}%
        \def\makeglossary{%
            \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
            \makeglossaries

```

```

}%
\or
  \let\glossary\gls@original@glossary
  \let\makeglossary\gls@original@makeglossary
\or
  \def\makeglossary{\makeglossaries}%
  \renewcommand*{\@gls@override@glossary}[1] [main]{%
    \gls@glossary{##1}%
  }%
\fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{%
\newcommand*{\@gls@symbolsdef}{%
  \newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
  \newcommand*\@printsymbols}[1] [] {\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
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifgls@subentrycounter
    \let\@gls@d@subentrycounterdef\relax
  \else
    \let\@gls@d@subentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
  \@gls@do@acronymsdef
  \@gls@do@numbersdef
  \@gls@do@symbolssdef
  \@gls@do@indexdef
  \@gls@doentrycounterdef
  \@gls@d@subentrycounterdef
}
\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*\glssettocitle[1]{%
\def\glossarytocitle{\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{\glsstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang[1]{%
\@ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\ProvidesGlossariesLang[1]{%
\ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions[1]{}
\ifglstranslate
\RequirePackage{tracklang}[2019/11/30]
\ifdefvoid\@gls@requested@languages{}%
{%
\@for\@gls@thislang:=\@gls@requested@languages\do{%

```

```

\TrackIfKnownLanguage{\@gls@thislang}{}%
{\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
}%
}%
\@gls@usetranslator
\@ifpackageloaded{translator}
{%
\ifboolexpr
{
test {\ifdefstring{\trans@languages}{English}}
and not
test {\ifdefstring{bbl@loaded}{english}}
}
{%
\let\glsifusetranslator\@secondoftwo
}%
{%
\usedictionary{glossaries-dictionary}%
\renewcommand*{\addglossarytocaptions}[1]{%
\ifcsundef{captions#1}{}%
{%
\expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
\expandafter\toks@\expandafter{\@gls@tmp
\renewcommand*{\glossaryname}{\translate{Glossary}}}%
}%
\expandafter\edef\csname captions#1\endcsname{\the\toks@}%
}%
}%
}%
}%
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
{%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
}%
}%
}%
}%
\glsifusetranslator
{%
\renewcommand*{\glssettoctitle}[1]{%
\ifcsdef{gls@tr@set@#1@toctitle}%

```



```

    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glssymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
}{}%
\fi
\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}
\ifglsexindy
  \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{\glslocationstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\@glossarypreamble{%
  \csuse{@glossarypreamble@currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*\@glossarypostamble{}
\newcommand*\@glossarysection[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\@glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\@glossarymark{#1}}
}

```

```

}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
  \else
    \@mkboth{#1}{#1}%
  \fi
}
\newcommand*{\setglossarysection}[1]{%
  \setkeys{glossaries.sty}{section=#1}}
\newcommand*{@glossarysection}[2]{%
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname[#1]{#2}%
  }%
  {%
    \csname\@glossarysec\endcsname*{#2}%
    \gls@toc{#1}{\@glossarysec}%
  }%
  \@glossaryseclabel
}
\newcommand*{@pglossarysection}[2]{%
  \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*\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

```

```

\ifglxindy
  \def\@xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
  \def\@do@ifinlist##1,#1,##2\end@ifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@ifinlist
}
\ifglxindy
  \newcommand*\@xdycounters{\@glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \@for\@gls@ctr:=#1\do{%
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
      \@do@addcounter
    }
  }
  \@onlypremakeg\GlsAddXdyCounters
\else
  \newcommand*\GlsAddXdyCounters[1]{%
    \glsnoxindywarning\GlsAddXdyAttribute
  }
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
  can't be used after \string\GlsAddXdyAttribute}{Move all
  occurrences of \string\GlsAddXdyCounters\space before the first
  instance of \string\GlsAddXdyAttribute}%
}
\ifglxindy
  \newcommand*\@glsaddxdyattribute[2]{%
    \edef\@xdyattributes{\@xdyattributes ^^J
      \string"#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{hyperrrm}
  \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{}
\ifglsxindy
\newcommand*\{GlsAddXdyAlphabet}[2]{%
\edef\@xdyuseralphabets{%
\@xdyuseralphabets ^^J
(define-alphabet "#1" (#2))}}
\else
\newcommand*\{GlsAddXdyAlphabet}[2]{%
\glsnoxindywarning\GlsAddXdyAlphabet}
\fi
\ifglsxindy
\newcommand*\{Gls@xdy@locationlist}{%
roman-page-numbers,%
Roman-page-numbers,%
arabic-page-numbers,%
alpha-page-numbers,%
Alpha-page-numbers,%
Appendix-page-numbers,%
arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
\string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%

```

```

\string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
\string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
\string"ALPHA\string"
:sep \string"\@glsAlphacompositor\string"
\string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
\string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][[]]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\glsopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsopenbrace"
#1
:sep "\glsclosebrace\glsopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnoxindywarning\GlsAddXdyLocation}
\fi

```



```

\ifglxindy
  \def\@xdylocationclassorder{^^J\space\space\space
    \string"roman-page-numbers\string"^^J\space\space\space
    \string"arabic-page-numbers\string"^^J\space\space\space
    \string"arabic-section-numbers\string"^^J\space\space\space
    \string"alpha-page-numbers\string"^^J\space\space\space
    \string"Roman-page-numbers\string"^^J\space\space\space
    \string"Alpha-page-numbers\string"^^J\space\space\space
    \string"Appendix-page-numbers\string"
    \@xdyuserlocationnames^^J\space\space\space
    \string"see\string"
  }
\fi
\ifglxindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoxywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string")}%
    }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi

```

```

\def\@xdylanguage#1#2{
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
  \ifglossaryexists{#1}{%
    \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
  }{%
    \PackageError{glossaries}{Can't set language type for
    glossary type '#1' --- no such glossary}{%
    You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
  \glsnoxindywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{
\ifglxindy
  \newcommand*\{GlsSetXdyCodePage}[1]{%
  \renewcommand*\{gls@codepage}{#1}%
  }
  \AtBeginDocument{%
  \ifx\gls@codepage@empty
    \@ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
  }
\else
  \newcommand*\{GlsSetXdyCodePage}[1]{%
  \glsnoxindywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
\ifglxindy@glsnumbers
  \def\@xdylettergroups{(define-letter-group
  \string"glxnumbers\string"^^J\space\space\space
  :prefixes (\string"0\string" \string"1\string"
  \string"2\string" \string"3\string" \string"4\string"
  \string"5\string" \string"6\string" \string"7\string"
  \string"8\string" \string"9\string")^^J\space\space\space
  \@xdynumbergrouporder)}
  \else
  \def\@xdylettergroups{}
  \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
  (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%

```

```

\@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forglsentries}[3][\glsdefaulttype]{%
\protected@edef\@glo@list{\csname glo@list@#1\endcsname}%
\@for#2:=\@glo@list\do
{%
\ifdefempty{#2}{#3}%
}%
}
\newcommand*\forallglsentries}[3][\@glo@types]{%
\expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
{%
\forglsentries[\@@this@glo@]{#2}{#3}%
}%
}
\newcommand{\ifglossaryexists}{%
\@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
\ifcsundef{glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
\ifcsundef{glo@list@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
\ifcsundef{glo@glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*\ifglsused}[3]{%
\ifbool{glo@glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
\ifglsentryexists{#1}{#2}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}{You need to define a glossary entry before you
can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
\ifglsentryexists{#1}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
\ifglsentryexists{#1}{#2}{%
\GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}%
}%
}
\newcommand{\glsdoifexistsordo}[3]{%
\ifglsentryexists{#1}{#2}{%

```

```

\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}{You need to define a glossary entry before you
can use it.}%
#3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
\ifglossaryexists*{#1}%
{%
\PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
#3%
}%
{#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
\glsdoifexists{#1}%
{%
\def\do@glshaschildren{#3}%
\protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
\expandafter\forglseries\expandafter
[\csname glo@\@gls@thislabel @type\endcsname]
{\glo@label}%
{%
\letcs\glo@parent{glo@\glo@label @parent}%
\ifdefequal\@gls@thislabel\glo@parent
{%
\def\do@glshaschildren{#2}%
\@endfortrue
}%
{}}%
}%
\do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
\glsdoifexists{#1}%
{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
}%
}
\newcommand*{\ifglshasdesc}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@desc}%
{#3}%
{#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
\ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
{#2}%
{#3}%
}
}

```

```

\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@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{\ifglsfieldvoid}[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@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*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
  \protected@edef\@glo@tmp{#1}%
  \ifx\@glo@tmp\@gls@defaulttype
    \GlossariesWarning{\string\defglsentryfmt\space attempting to set
      a display style for \string\glsdefaulttype\space but there
      doesn't seem to be a default glossary. If you have used
      'nomain', you need to define an alternative glossary}%
  \else
    \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
  \fi
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtext}[1]{%
  \ifboolexpr{test {\ifdefstring{#1}{tex}}

```

```

        or test {\ifdefstring{#1}{TEX}}
    {%
      \def#1{nottex}%
      \PackageError{glossaries}%
        {Forbidden '.tex' extension replaced with '.nottex'}%
        {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
        Don't use '.tex' as an extension for a temporary file.}%
    }%
  }%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaulttype
    {%
      \gdef\glsdefaulttype{#2}%
    }{}%
    \toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
    \expandafter\gdef\csname glolist@#2\endcsname{,}%
    \expandafter\edef\csname @glo@type@#2@log\endcsname{#1}%
    \expandafter\edef\csname @glo@type@#2@in\endcsname{#3}%
    \expandafter\edef\csname @glo@type@#2@out\endcsname{#4}%
    \expandafter\@gls@forbidtextext\csname @glo@type@#2@log\endcsname
    \expandafter\@gls@forbidtextext\csname @glo@type@#2@in\endcsname
    \expandafter\@gls@forbidtextext\csname @glo@type@#2@out\endcsname
    \expandafter\def\csname @glo@type@#2@title\endcsname{#5}%
    \@gls@provide@newglossary
    \protected@write\@auxout{\{\string\@newglossary{#2}{#1}{#3}{#4}}%
    \ifcsundef{gls@#2@entryfmt}%
    {%
      \defglsentryfmt[#2]{\glsentryfmt}%
    }%
    {}%
    \@gls@defsortcount{#2}%
    \@ifnextchar[{\@gls@setcounter{#2}}%
    {\@gls@setcounter{#2}[\glscounter]}%
  }%
  }%
  \gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}

```

```

}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @gls@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*{\@gls@getcounter}[1]{%
\csname @gls@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*{\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{gllist@#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\@gls@name{#1}%
}
\define@key{glossentry}{description}{%

```



```

\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*{\@gls@default@value}{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}%
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
\renewcommand*{\gls@xr@key}{#1}%
\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{\@gls@nonextpages}%
\@gls@savenonumberlist{true}%
\else
\def\@glo@prefix{\@gls@nextpages}%
\@gls@savenonumberlist{false}%
\fi
}
\newcommand*{\@gls@noidx@nonumberlist}[1]{%
\cslet\@glo@\@glo@label @prenumberlist}\@glo@prefix
}%
\newcommand*\@gls@noidx@prenumberlist}[1]{%
\csuse{glo@\@gls@detoklabel{#1}@prenumberlist}}
\newcommand*{\@gls@savenonumberlist}[1]{%
}
\newcommand*{\@gls@initnonumberlist}{%
}
\newcommand*{\@gls@storenonumberlist}[1]{%
}
\renewcommand*{\@gls@initnonumberlist}{%
\undef\@glo@nonumberlist
}%
\renewcommand*{\@gls@savenonumberlist}[1]{%
\def\@glo@nonumberlist{##1}%
}%
\renewcommand*{\@gls@storenonumberlist}[1]{%
\ifdef\@glo@nonumberlist
{%
\cslet\@glo@\@gls@detoklabel{##1}@nonumberlist}\@glo@nonumberlist}%
}%
{}%
}%

```

```

\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \iflssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'}.

```

```

    Maybe makeglossaries + rerun required}%
\else
\PackageError{glossaries}%
{Package option 'savenumberlist=true' required}%
{%
    You must use the 'savenumberlist' package option
    to reference location lists.%
}%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
    \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
        \edef\@gls@value{\expandonce{#1}}%
        \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
        \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
    \edef\@gls@value{\expandonce{#1}}%
    \@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
    \protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
        \edef\@gls@value{\expandonce{#1}}%
        \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
}
}

```

```

    }%
    {%
      \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
      {%
        \@@gls@expand@field{#2}{#3}{#4}%
      }%
      {%
        \csuse{gls@assign@#3@field}{#2}{#4}%
      }%
    }%
  }%
}
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \@@gls@expand@field{#2}{#3}{#1}%
  }%
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
}
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\gls@expandfields{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*\gls@noexpandfields{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*\gls@defdocnewglossaryentry{%
  \let\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{\newglossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  }%
  \ifglsentryexists{#1}%
  {%
    \xifinlist{#1}{\gls@docdef@labellist}%
    {%
      \GlossariesWarning{Entry ‘#1’ has already been defined in the
        document environment (ignoring new definition)}%
    }%
    {%
      \xifinlist{#1}{\gls@glsdefs@labellist}%
      {%
        \gls@defglossaryentry{#1}{#2}%
        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
      }%
      {%
        \GlossariesWarning{Entry ‘#1’ has already been defined in
          the preamble (in which case remove the duplicate definition)
          or you have just updated to glossaries.sty
          v4.47 (in which case delete ‘\jobname.glsdefs’)}%
      }%
    }%
  }%
  }%
  \gls@defglossaryentry{#1}{#2}%
  \gls@docdef@addlabel{#1}%
  \@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs){%
  \@gls@enablesavenonumberlist
  \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{}{}%
  \@gls@restreat
  \undef\@gls@restreat
  \gls@defdocnewglossaryentry
}

```

```

\newcommand*{\glsdefs@newdocentry}[2]{%
  \ifglsentryexists{#1}%
  {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
    (if you have moved the definition there from the document
     environment, delete \jobname.glsdefs)}}%
  {%
    \gls@glsdefs@addlabel{#1}%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs@glo@value{glo@\glsdetoklabel{#1}@\expandafter\@secondoftwo\@gls@map}%
    \ifdef@glo@value
    {%
      \@onelevel@sanitize@glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter@gobble\string\{\glo@value\expandafter@gobble\string\},%
        \glspercentchar
      }%
    }%
  }%
  {}%
}%
\gls@writedefhook
\immediate\write\@gls@deffile
{%
  \glspercentchar^^J%
  \expandafter@gobble\string\}\glspercentchar^^J%
}%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%

```

```

{user1}{useri},%
{user2}{userii},%
{user3}{useriii},%
{user4}{useriv},%
{user5}{userv},%
{user6}{uservi},%
{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
      {%
        \edef#1{\expandafter\@secondoftwo\@gls@map}%
        \@endfortrue
      }%
    }%
  }%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname gls@assign@#1@field\endcsname}[2]{%
        \@gls@expand@field{##1}{#1}{##2}%
      }%
  }%
  }%
  \@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  }%
}

```



```

\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*\glsaddkey{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*\@sglsaddkey[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
{}%
\@glsaddkey{#1}%
}
\newcommand*\@glsaddkey[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @gls@user@#1\endcsname}[2] [] {%
\new@ifnextchar [%
{\csuse{@gls@user@#1@}{##1}{##2}}%
{\csuse{@gls@user@#1@}{##1}{##2} []}%
\csdef{@gls@user@#1@}##1##2[##3]{%
\@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
\expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#6’ as helper command

```

```

        '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
    }%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2] []{%
    \new@ifnextchar[%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
  '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  }%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2] []{%
    \new@ifnextchar[%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1@\endcsname}%
}%
\glsmfuaddmap{#3}{#4}%
\glsmfuaddmap{#5}{#6}%
\glsmfublocker{#7}%
}%
{%
  \PackageError{glossaries}{Key '#1' already exists}{}%
}%
}
\newrobustcmd{\glsfieldxdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csxdef{glo@\@glo@label @#2}{#3}%
  }%
}

```

```

    {%
      \PackageError{glossaries}{Field '#2' doesn't exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfielddedf}[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]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@gls@name
  \let\@glo@desc\@gls@desc
  \let\@glo@descplural\@gls@default@value
  \let\@glo@type\@gls@default@value
  \let\@glo@symbol\@gls@default@value
  \let\@glo@symbolplural\@gls@default@value
  \let\@glo@text\@gls@default@value
  \let\@glo@plural\@gls@default@value
  \let\@glo@first\@gls@default@value
  \let\@glo@firstplural\@gls@default@value
  \let\@glo@sort\@gls@default@value
  \let\@glo@counter\@gls@default@value
  \def\@glo@see{}%
  \def\@glo@parent{}%
  \def\@glo@prefix{}%
  \@gls@initnonumberlist
  \def\@glo@useri{}%

```

```

\def@glo@userii{%
\def@glo@useriii{%
\def@glo@useriv{%
\def@glo@userv{%
\def@glo@uservi{%
\def@glo@short{%
\def@glo@shortpl{%
\def@glo@long{%
\def@glo@longpl{%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glolist@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
}%
\ifignoredglossary@glo@type
{%
  \ifx@glo@desc@glsnodesc
    \let@glo@desc@empty
  \fi
}%
}%
\protected@edef@glolist@{\csname glolist@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glolist@\@glo@type\endcsname{%
  \@glolist@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx@glo@parent@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal@glo@label@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
\fi

```

```

}%
{%
\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{@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\@glsnname
  \glsnname
  \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{,%
\for@gl@sentries[\@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}%
}
\ifglxsindy
\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\@cgl@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cglformat{##2}{##3}%
\glsunset{##2}%
\else
\@gls@{##1}{##2}[##3]%
\fi
}%
\def\@cgl@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglspformat{##2}{##3}%
        \glset{##2}%
    \else
        \cglsp@{##1}{##2}[##3]%
    \fi
}%
\def\cGls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglset{##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@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsetentries{\@glsentry}{%
        \ifglsetused{\@glsentry}%
        {\immediate\write\@auxout
            {\string\@gls@entry@count{\@glsentry}{\glsetentrycurrcount{\@glsentry}}}}%
        {}%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cglset}{\@gls@hyp@opt\@cglset}
\newcommand*{\@cglset}[2][{}]{%

```

```

    \new@ifnextchar[{\@cgl@s@{#1}{#2}}{\@cgl@s@{#1}{#2} []}%
}
\def\@cgl@s@#1#2[#3]{%
  \GlossariesWarning{\string\cgl@s\space is defaulting to
    \string\gl@s\space since you haven't enabled entry counting}%
  \@gl@s@{#1}{#2} [#3]%
}
\newcommand*{\cgl@sformat}[2]{%
  \ifglshaslong{#1}{\gl@sentrylong{#1}}{\gl@sentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gl@s@hyp@opt\@cGls}
\glsmfuaddmap{\cgl@s}{\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*{\cgl@spl}{\@gl@s@hyp@opt\@cgl@spl}
\newcommand*{\@cgl@spl}[2] []{%
  \new@ifnextchar[{\@cgl@spl@{#1}{#2}}{\@cgl@spl@{#1}{#2} []}%
}
\def\@cgl@spl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl@spl\space is defaulting to
    \string\gl@spl\space since you haven't enabled entry counting}%
  \@gl@spl@{#1}{#2} [#3]%
}
\newcommand*{\cgl@splformat}[2]{%
  \ifglshaslong{#1}{\gl@sentrylongpl{#1}}{\gl@sentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gl@s@hyp@opt\@cGlspl}
\glsmfuaddmap{\cgl@spl}{\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] [\@gl@s@default]{%
  \let\@gl@s@default\gl@sdefaulttype

```



```

\def\glsdefaulttype{#1}\input{#2}%
\let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glsformat[1]{#1}
\newcommand*\glsentryfmt{\glsentryfmt}
\newcommand*\glsentrytext{\glsentrytext}
\ifdefempty\glsentrytext
{%
\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*{\glsnacfmt}{%
  \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\glsentencecase\expandafter{\gls@text}%
}

```

```

\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
  \ifKV@glslink@local
  \let\@gls@do@glsunset\glslocalunset
  \else
  \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\glslinkvar}[3]{#1}
\newcommand*\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree

```

```

\@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][{}]{%
\let\glslinkvar\@thirdofthree
\@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
\@gls@hyp@opt\@gls@link
}
\newcommand*{\@gls@link}[3][{}]{%
\glsdoifexistsordo{#2}%
{%
\let\do@gls@link@checkfirsthyper\relax
\@gls@link[#1]{#2}{#3}%
}%
\glstextformat{#3}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
\@gls@hyp@opt\@Gls@link
}
\newcommand*{\@Gls@link}[3][{}]{%
\@gls@link[#1]{#2}{\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{\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\glsstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper@ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\@do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\glsformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\glsformat{#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@mkidxrangearchar#3\@nil
    \protected@edef#1{%
        \@glo@prefix setentrycounter[#4]{#2}%
        \expandafter\string\csname\@glo@suffix\endcsname
    }%
    \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangearchar#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@dostsubst{%
        \noexpand\DTLsubstituteall\noexpand\gls@xdystring
        {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
      }%
      \gls@dostsubst
    }%
  \fi
  \let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \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"\@gls@quotechar}%
\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@checkeschar{\@gls@checkeschar#2\|\|\@null}%
\else
\def\@@gls@checkeschar{\@gls@checkeschar#2\|#3\@null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkescllevel#1\!#2\!#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescllevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\\"\@gls@levelchar
\@gls@quotechar\string\\"\@gls@levelchar}%
\def\@@gls@checkescllevel{\@gls@checkescllevel#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\\"\@gls@levelchar}%
\ifx\@null#3\@null
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!\!\@null}%
\else
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!#3\@null}%
\fi
\fi
\@gls@checkescllevel
}
\def\@gls@checkbar#1|#2|#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@@gls@checkbar{\@gls@checkbar#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

    \@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
  \def\@gls@checkbar{\@gls@checkbar#2|\null}%
\else
  \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checklevel{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
      \def\@gls@checklevel{\@gls@checklevel#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@levelchar}%
    \ifx\null#3\null
      \def\@gls@checklevel{\@gls@checklevel#2!\null}%
    \else
      \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
    \fi
  \fi
  \@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else

```

```

        \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
        \fi
        \fi
        \@gls@checkactual
    }
    \bgroup
    \catcode'\|0\relax
    \catcode'\|12\relax
    \gdef\@gls@checkesc#1\#2\#3\null{%
        \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
        \toks@={#1}%
        \ifx\null#2\null
            \ifx\null#3\null
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                \def\@gls@checkesc{\relax}%
            \else
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                    \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
                \def\@gls@checkesc{\@gls@checkesc#3\null}%
            \fi
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \@gls@quotechar\@backslashchar}%
            \ifx\null#3\null
                \def\@gls@checkesc{\@gls@checkesc#2\| \null}%
            \else
                \def\@gls@checkesc{\@gls@checkesc#2\#3\null}%
            \fi
        \fi
        \@gls@checkesc
    }
    \egroup
    \def\@gls@xdycheckquote#1"#2"#3\null{%
        \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
        \toks@={#1}%
        \ifx\null#2\null
            \ifx\null#3\null
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                \def\@gls@xdycheckquote{\relax}%
            \else
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                    \string"\string"}%
                \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
            \fi
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \string"}%
            \ifx\null#3\null
                \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
            \else

```

```

        \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
        \fi
        \fi
    \@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*{\glsdohypertarget}[2]{%
\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{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@#1}{#2}]{\@gls@#1}{#2}[{}]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\gls@type @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
  \new@ifnextchar[{\@Gls@#1}{#2}]{\@Gls@#1}{#2}[{}]}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }%
}

```



```

\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2][{}]{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}]%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@scaps\@thirdofthree
\let\gls@customtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glsp1}{\@gls@hyp@opt\@glsp1}
\newcommand*{\@glsp1}[2][{}]{%
\new@ifnextchar[{\@glsp1@{#1}{#2}}{\@glsp1@{#1}{#2}[]}]%
}
\def\@glsp1@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\gls@scaps\@firstofthree
\let\gls@customtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\glsmfuaddmap{\glsp1}{\Glspl}
\newcommand*{\@Glspl}[2][{}]{%
\new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}]%
}
\def\@Glspl@#1#2[#3]{%
\glsdoifexists{#2}%

```

```

{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\gl@ifplural\@firstoftwo
  \let\gl@scapscase\@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*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
\gl@mf@blocker{\GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \gl@do@ifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@ifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@do@ifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@ifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gl@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}]{%
  \@gl@disp[#1]{#2}{\gl@sentencecase{#3}}%
}
\gl@mf@u@ddmap{\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*{\glsymbol}{\@gls@hyp@opt\@glsymbol}

```

```

\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
\def\@glssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\glsmfuaddmap{\glssymbol}{\Glsymbol}
\newcommand*{\@GLssymbol}[2] [] {%
  \new@ifnextchar [{\@GLssymbol@{#1}{#2}}{\@GLssymbol@{#1}{#2} []}]
\def\@GLssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbol{#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}{\glssuppercase{\glstentrysymbol{#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}{\glstentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfuaddmap{\glssymbolplural}{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar [{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#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}{\glssuppercase{\glstentrysymbolplural{#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}{\glstentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\glsmfuaddmap{\glsuseri}{\Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar [{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
\newcommand*{\@GLSuseri}[2][{}]{%
    \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}}
\def\@GLSuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][{}]{%
    \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
\def\@glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\glsmfuaddmap{\glsuserii}{\Glsuserii}
\newcommand*{\@Glsuserii}[2][{}]{%
    \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
\def\@Glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2][{}]{%
    \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}}
\def\@GLSuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2][{}]{%
    \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}}
\def\@glsuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2][{}]{%
    \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
\def\@Glsuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2][{}]{%
    \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}}
\def\@GLSuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}

```

```

\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar [{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar [{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar [{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%
  \new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2] [] {%
  \new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
\def\@GLSuserv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] [] {%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
\def\@glsuservi@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\glsmfuaddmap{\glsuservi}{\Glsuservi}
\newcommand*{\@Glsuservi}[2] [] {%
  \new@ifnextchar [{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}]
\def\@Glsuservi@#1#2[#3] {%

```



```

    \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
\def\@GLSuservi@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
}
\def\@acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\gls label{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%

```

```

\new@ifnextchar [{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
\new@ifnextchar [{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\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\glsapscase\@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@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{%
      \glsuppercase{\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{%
      \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]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
\newcommand*{\Gls@entryname}[1]{\@Gls@entry@field{#1}{name}}
\newcommand*{\Gls@entryname}[1]{%
  \glstexorpdfstring{\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\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\glssentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc[1]{\@Gls@entry@field{#1}{desc}}
\newcommand*\Glsentrydesc[1]{\@Glsentryfield{#1}{desc}}
\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*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfuexcl{\Glsentrylong}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuexcl{\Glsentrylongpl}
\newcommand*{\glsentryfull}[1]{%
  \glsentrylong{#1} (\glsentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\glsentryshort{#1})%
}
\glsmfuexcl{\Glsentryfull}
\newcommand*{\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}
\glsmfuexcl{\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
\ifglsxindy
  \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsxindy
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
  }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

\renewcommand*\@xdynumbergrouporder}{#1}%
\@onelevel@sanitize\@xdynumbergrouporder
}
\else
\newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
\glsnoxywarning\GlsSetXdyNumberGroupOrder}
\newcommand*\s@GlsSetXdyNumberGroupOrder}{%
\@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*\@glsminrange}{2}
\ifglsxindy
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\renewcommand*\@glsminrange}{#1}}
\else
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef\glswrite{\newwrite\glswrite}{}%
\@gls@addpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite}{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space

```

```

:min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}\{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}\{"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{}\{\string\providecommand*%
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
        {%
          \string\setentrycounter
            [\expandafter\@gobble\string\#1]{\@this@ctr}%
          \expandafter\string

```

```

\csname\@this@attr\endcsname
  {\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string" \glpercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspersentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspersentchar\space for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string{\string
\glossaryentrynumbers\string{\string\relax \string"}
\write\glswrite{delim_1 \string"\string{\string

```



```

    \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_2 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string"\string\}\string}\string}
\write\glswrite{delim_n \string"\string\delimN \string}
\write\glswrite{delim_r \string"\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
    \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
    \string"\string}\string\relax
    \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
    \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
    \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\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#####2#1###3\noexpand\null{

```

```

\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%

```

```

\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkeschar####1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescllevel####1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote####1%
\expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%

```

```

        \noexpand\def\noexpand\@gls@checkescquote{%
          \noexpand\@gls@checkescquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \noexpand\@gls@quotechar\noexpand\string
  \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
  \expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
  ####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{}
\newcommand{\noist}{%
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
  \GlossariesWarning{too late to use \string\noist. The style file
  has already been written}%
}
\newcommand*\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
    \ifglssavewrites
      \expandafter\newtoks\csname glo@#1@filetok\endcsname
    \else
      \expandafter\newwrite\csname glo@#1@file\endcsname
      \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%

```

```

\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@glsopenfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglsxindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%
\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\@gls@run@unrestricted@shell{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@run@output@file{\gls@istfilebase}
\@gls@codepage
-t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
-o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
\@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
}%
}%
\GlossariesWarning{Can’t make glossary ‘#1’, it doesn’t exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-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@onlypremakeg
\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@wrglossdisablelocationcmds}{%
    \string\@gls@reference

```

```

        {\csname glo@\gls@label @type\endcsname}%
        {\@gls@label}%
        {\string\glsnoidxdisplayloc
         {\@glo@counterprefix}%
         {\@gls@counter}%
         {\@glsnumberformat}%
         {\@glslocref}}%
        }%
    }%
}
\AtEndDocument{%
    \glswritefiles
}
\newcommand*{\@glswritefiles}{%
    \forallglossaries{\@glo@type}{%
        \ifcsundef{glo@\@glo@type @filetok}%
        {%
            \def\gls@tmp{}}%
        }%
        {%
            \edef\gls@tmp{\expandafter\the
                \csname glo@\@glo@type @filetok\endcsname}%
        }%
        \ifx\gls@tmp\@empty
            \ifx\@glo@type\glsdefaulttype
                \GlossariesWarningNoLine{Glossary '@@glo@type' has no
                    entries.^^JRemember to use package option 'nomain' if
you
                    don't want to^^Juse the main glossary}%
            \else
                \GlossariesWarningNoLine{Glossary '@@glo@type' has no
                    entries}%
            \fi
        \else
            \@glsopenfile{\glswrite}{\@glo@type}%
            \immediate\write\glswrite{%
                \expandafter\the
                \csname glo@\@glo@type @filetok\endcsname}%
            \immediate\closeout\glswrite
        \fi
    }%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}

```

```

\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
      \expandafter{\@gls@tmp~J}%
  \else
    \ifcsdef{glo@#1@file}%
      {%
        \expandafter\protected@write\csname glo@#1@file\endcsname{%
          \gls@disablepagerefexpansion}{#2}%
        }%
      {%
        \ifignoredglossary{#1}{}%
          {%
            \GlossariesWarning{No file defined for glossary ‘#1’}%
          }%
        }%
      \fi
    \endgroup\@esphack
  }
\newcommand*{\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\gls@writeentry{#1}{\@do@wrglossary{#1}}}%
}
\newcommand*{\gls@writeentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
  \gls@wrglossdisablelocationcmds
  \ifglsclocations
    \for\@gls@this:=\gls@protected@pagefmts\do
      {%
        \expandafter\let\@gls@this\relax
      }%
  \fi
}
\newcommand{\gls@wrglossdisablelocationcmds}{}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}

```

```

\newcommand*\gls@romanpage{\romannumeral\c@page}
\newcommand*\gls@Romanpage{\@Roman\c@page}
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifglsclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
          \expandonce{\csname gls#1page\endcsname}%
          \expandonce{\csname org@gls#1\endcsname}%
        }%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}
\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand{\gls@wrglossdisableanchorcmds}{\let\gls@texorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\gls@wrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\gls@wrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
\newcommand{\gls@wrglosslocationtarget}[1]{#1}
\newcommand{\gls@wrglosslocationtextfmt}[1]{#1}
\newcommand*\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@gls@locref\expandafter{\theglsentrycounter}%
  \expandafter\def\expandafter\@gls@Hlocref\expandafter{\theHglentrycounter}%
  \ifx\@gls@Hlocref\@gls@locref
    \def\@gls@counterprefix{}%
  \else
    \@xp@gls@getcounterprefix{\@gls@locref}{\@gls@Hlocref}%
  \fi
  \protected@edef\@gls@label{\gls@detoklabel{#1}}%

```

```

\do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\gls@locref{\the\glsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theH\glsentrycounter\the\glsentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\gls@Hlocref{\theH\glsentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\@xp@gls@getcounterprefix{\@gls@locref}{\@gls@Hlocref}%
\fi
\protected@edef\gls@label{\gls@detoklabel{#1}}%
\do@wrglossary
}
\newcommand*{\do@wrglossary}{%
\ifglsxindy
\expandafter\@glo@check@mkidxrangechar\@gls@numberformat\@nil
\def\@glo@range{%

```

```

\expandafter\if\@glo@prefix(\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix)\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string" %
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string@glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bggroup
\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} \glsseeitem{#2}}
\newrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
  }
  \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
  \let\@gls@dolast\glsseelastsep
  \let\@gls@donext\glsseesep

```

```

    }%
  }
  \newcommand*\glsseelastsep{\space\andname\space}
  \newcommand*\glsseesep{, }
  \newrobustcmd*\glsseeitem[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
  \newcommand*\glsseeitemformat[1]{%
    \ifgls hasshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
  }
  \newcommand*\gls@save@numberlist[1]{%
    \ifgls@savenumberlist
      \toks@{#1}%
      \protected@edef\@do@writeaux@info{%
        \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
      }%
      \@onelevel@sanitize\@do@writeaux@info
      \protected@write\@auxout{}\@do@writeaux@info%
    \fi
  }
  \newcommand*\warn@noprntglossary{}%
  \ifcsundef{printglossary}{}%
  {%
    \@gls@warnonglossdefined
    \undef\printglossary
  }
  \newcommand*\@printgloss@checkexists@noignored}[2]{%
    \ifglossaryexists{#1}%
    {#2}%
    {%
      \ifignoredglossary{#1}%
      {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
      {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
    }%
  }
  \newcommand*\@printgloss@checkexists@allowignored}[2]{%
    \s@ifglossaryexists{#1}%
    {#2}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \newcommand*\@printglossary}[1][type=\glsdefaulttype]{%
    \let\@printgloss@checkexists\@printgloss@checkexists@noignored
    \@printglossary{#1}{\@printglossary}%
  }
  \newcommand*\@printglossaries{%
    \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
  }
  \newcommand*\@printnoidxglossary}[1][type=\glsdefaulttype]{%
    \let\@printgloss@checkexists\@printgloss@checkexists@noignored
    \@printglossary{#1}{\@printnoidx@glossary}%
  }
}

```

```

\newcommand*\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*\@printgloss@setsort}{%
\newcommand*\@gls@preglossaryhook}{%
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '\@glo@type'. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \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}{\endgroup}
\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
{
  \glsdoifexists { #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:Ne #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:Ne #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:Ne \l__glossaries_entry_letter_title_tl

```

```

        { \exp_args:No \exp_not:o { \DTLsortedletter { ##1 } } }
\tl_set:Nc \l__glossaries_entry_letter_group_tl
  { \DTLsortedletter { ##1 } }
\tl_set:Nc \l__glossaries_entry_label_tl { ##1 }
\group_begin:
  \cs_set_eq:NN \__glossaries_actual:nm \__glossaries_actual_label:nm
  \exp_args:NNNe
\group_end:
  \tl_set:Nn \l__glossaries_entry_label_tl
    { \l__glossaries_entry_label_tl }
\exp_args:NnV \listcseadd { @glsref@ #2 } \l__glossaries_entry_label_tl
\rtl_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:nmm
  {
    \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\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{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:V \@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:n
{
  \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
  }%
  {%
    \glsxtrifemptyglossary{\@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{glsymbols}%
\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@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{\setglossarystyle{#1}}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@glossarysecstar{*}%
    \renewcommand*\@glossaryseclabel{}%
  \or
    \renewcommand*\@glossarysecstar{}%
    \renewcommand*\@glossaryseclabel{}%
  \or
    \renewcommand*\@glossarysecstar{}%
    \renewcommand*\@glossaryseclabel{\label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*\@glossarysecstar{*}%
    \renewcommand*\@glossaryseclabel{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{\glsautoprefix\@glo@type}}%
  \fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glsesubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}

```

```

\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else

```

```

\ifglssubentrycounter
  \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
  \gls{#1}%
\fi
\fi
}
\newcommand*\glsentrycounterlabel{%
\ifglentrycounter
  \theglossaryentry.\space
\fi
}
\newcommand*\glssubentrycounterlabel{%
\ifglssubentrycounter
  \theglossarysubentry)\space
\fi
}
\newcommand*\glsentryitem}[1]{%
\ifglentrycounter
  \glsstepentry{#1}\glsentrycounterlabel
\else
  \glsresetsubentrycounter
\fi
}
\newcommand*\glssubentryitem}[1]{%
\ifglssubentrycounter
  \glsstepsubentry{#1}\glssubentrycounterlabel
\fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheglossdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\newcommand*\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%

```

```

        \glsnamefont{\Glsentryname{#1}}%
    }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
\newcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glsentrydesc{#1}%
    }%
}
\newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \Glsentrydesc{#1}%
    }%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
\newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glsentrysymbol{#1}%
    }%
}
\newcommand*{\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \Glsentrysymbol{#1}%
    }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newrobustcmd*{\glsgetgrouptitle}[1]{%
    \@gls@getgrouptitle{#1}{\@gls@grptitle}%
    \@gls@grptitle
}
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
    \def#2{#1}%
    \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
    { \ifcsundef{#1groupname}{-}{\letcs#2{#1groupname}} }
}
\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][ ]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\renewglossarystyle}[2]{%

```



```

\ifcsundef{@glsstyle@#1}%
{%
  \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
}%
{%
  \csdef{@glsstyle@#1}{#2}%
}%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\xp@glsnumber{#1}}
}
\newrobustcmd{\xp@glsnumber}[1]{%
  \glsnumber#1\nohyperpage{ }\@nil
}
\def\@glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \gls@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glsnumber#3\@nil
  \fi
}
\def\gls@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
  \gls@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
  \begingroup

```

```

\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else
\bgrou
\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}%
}
\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={\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}{%
    \glsuppercase{\genacrfullformat{##2}{##3}}}%
}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genplacrfullformat{##2}{##3}}}%
}%

```

```

\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields[description={\the\glslongtok}]
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*\acronymsort}[2]{#1}
\newcommand*\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
}%
}%
}
\newcommand*\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%

```

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%

```

```

\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglsahaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%

```



```

\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
}

```

```

\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{dua}%

```

```

{%
\ifdefempty\glscustomtext
{%
\ifglshaslong{\glslabel}%
{%
\glsifplural
{%
\glscapscase
{%
\glsentrylongpl{\glslabel}\glsinsert
}%
{%
\Glsentrylongpl{\glslabel}\glsinsert
}%
{%
\glsupercase
{\glsentrylongpl{\glslabel}\glsinsert}%
}%
}%
}%
{%
\glscapscase
{%
\glsentrylong{\glslabel}\glsinsert
}%
{%
\Glsentrylong{\glslabel}\glsinsert
}%
{%
\glsupercase
{\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}{%
\glsupercase{\glsentrylong{##2}##3\space

```

```

        (\acronymfont{\glsentryshort{##2}})}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsupercase{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\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{\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*{\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*\showglouseri}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useri\endcsname
}
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*\showglouseriii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*\showglouseriv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*\showglouserv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*\showglouservi}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*\showgloname}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*\showglodesc}[1]{%

```



```

\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*\showglodescplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*\showglosort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*\showglosymbol}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*\showglosymbolplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*\showgloshort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*\showglolong}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*\showgloindex}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*\showgloflag}[1]{%
\expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*\showgloloclist}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
}
\newcommand*\showglofield}[2]{%
\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 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*\glsgroupeheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*\glossaryheader{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*\glsgroupeheading}[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}{\glsncols}
    \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}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenoname}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%

```

```

        \begin{multicols}{\glsmcols}
        \setlength{\parindent}{0pt}%
        \setlength{\parskip}{Opt plus 0.3pt}%
    }%
    {\end{multicols}}%
}
\newglossarystyle{mcoltreenamegroup}{%
    \setglossarystyle{mcoltreename}%
    \renewcommand{\glsgroupheading}[1]{\par
        \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenamehypergroup}{%
    \setglossarystyle{mcoltreename}%
    \renewcommand*{\glossaryheader}{%
        \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    \renewcommand*{\glsgroupheading}[1]{%
        \par\noindent
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
        \indexspace}%
}
\newglossarystyle{mcoltreenamepannav}{%
    \setglossarystyle{treename}%
    \renewenvironment{theglossary}%
    {%
        \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
        \setlength{\parindent}{0pt}%
        \setlength{\parskip}{Opt plus 0.3pt}%
    }%
    {\end{multicols}}%
    \renewcommand*{\glsgroupheading}[1]{%
        \par\noindent
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
        \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
    \setglossarystyle{almtree}%
    \renewenvironment{theglossary}%
    {%
        \begin{multicols}{\glsmcols}
        \def\@gls@prevlevel{-1}%
        \mbox{}\par
    }%
    {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
    \setglossarystyle{mcolalmtree}%
    \renewcommand{\glsgroupheading}[1]{\par
        \def\@gls@prevlevel{-1}%
        \hangindentOpt\relax
        \parindentOpt\relax

```

```

    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}{%
    {%
      \begin{multicols}{\glscols}[\noindent\glstreenavigationfmt{\glsnavigation}]
      \def\@gls@prevlevel{-1}%
      \mbox{}\par
    }%
    {\par\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
}

```

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{\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}{\ifglsgroups\else\indexspace\fi}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}

```

```

\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space\glossentrysymbol{##1}}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstarget{##2}{\strut}%
    \glossentrydesc{##2}\glspostdescription\space##3\par
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt
    {\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newrobustcmd*{\glsfindwidestoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[##1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%

```



```

\ifglshasparent{\@glo@label}%
{}%
{%
  \glsmeasurewidth{\dimen@}%
  {\glstreenamefmt{\glsentryname{\@glo@label}}}%
  \ifdim\dimen@>\gls@tmplen
    \gls@tmplen=\dimen@
    \letcs{\@glswidestname}{glo@\glsdetoklabel{\@glo@label}@name}%
  \fi
}%
}%
}
\newcommand*\glssetwidest}[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
    #2}%
}
\newcommand*\@glswidestname{}
\newcommand*\glstreenamebox}[2]{%
  \makebox[#1][l]{#2}%
}
\newglossarystyle{almtree}{%
  \renewenvironment{theglossary}%
    {\def\@gls@prevlevel{-1}%
     \mbox{}\par}%
    {\par}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand*\glossentry}[2]{%
    \ifnum\@gls@prevlevel=0\relax
    \else
      \glsmeasurewidth{\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}{%
        \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
          \glsmeasurewidth{\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[Opt][r]{\glstreenamebox{\gls@tmplen}{%
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
  \setglossarystyle{almtree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
    \par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
  \setglossarystyle{almtree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}}

```

Change History

1.01 – 2007-05-17	
General: Added range facility in format key	132
\writeist: Added spaces after \delimN and \delimR in ist file	187
1.04 – 2007-08-03	
General: Added \glstextformat	120
1.05 – 2007-08-10	
\glossarysection: added	
\@mkboth to	
\glossarysection	56
\gls@defglossaryentry:	
Changed the default value of the sort key to just the value of the name key	102
1.07 – 2007-09-13	
\@gls@link: fixed bug caused by \theglsentrycounter setting the page number too soon	130
\glsadd: fixed bug caused by \theglsentrycounter setting the page number too soon	184
1.08 – 2007-10-13	
General: Added babel support	50
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle	304
listgroup: changed listgroup style to use \glsgetgrouptitle	303
1.1 – 2008-02-22	
General: msgen now loaded (\new@ifnextchar needed)	6
\@glossarysection: numbered sections and auto label added	58
\@gls@tmpb: changed \toksdef to \newtoks	134
\@gls@toc: numberline added	59
\@p@glossarysection: numbered sections and auto label added	58
numberedsection:	
numberedsection package option added	11
numberline: numberline option added	10
\setglossarysection: new	58
translate: translate option added	32
1.12 – 2008-03-08	
General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget)	142
\@GLSpl: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	148
\@GLspl0: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	147
\@glspl0: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	146
descriptionplural: new	82
\gls@defglossaryentry:	
Changed default first plural to be first key with s appended (was text key with s appended)	101
descriptionplural support added	101
symbolplural support added	101
\Glsentrydescplural: New	176
\glsentrydescplural: New	176
\Glsentrysymbolplural: New	177
\glsentrysymbolplural: New	177
symbolplural: new	83
1.13 – 2008-05-10	
General: fixed bug that ignored 3rd parameter	149–158
\ACRfullpl: new	270
\Acrfullpl: new	269
\acrfullpl: new	269
\acrpluralsuffix: New	266
\gls@defglossaryentry:	
Changed default first value	101

Changed default firstplural value	101	\@GLspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	147
Removed restriction on only using \newglossaryentry in the preamble	106	\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	144
\newacronym: Removed restriction on only using \newacronym in the preamble	266	\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used	148
1.14 – 2008-06-17		\@glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	146
General: added nonumberlist key to \printglossary	252	\@gls@defglossaryentry: Changed def to let	101
added numberedsection key to \printglossary	250	1.17 – 2008-12-26	
\@gls@hypergroup: new	298	General: added xindy support	35
\firstacronymfont: new	270	\@do@esc@wrglossary: new	214
\glsautoprefix: new	10	\@do@seeglossary: new	218
\glsnavhyperlink: changed \edef to \protected@edef	295	\@glo@storeentry: new	108
\glsnavhypertarget: added write to aux file	297	\@gls@glossary: changed definition to use \index instead of \@index	210
\glsnavigation: changed to only use labels for groups that are present	298	\@gls@defaultplural: new	87
1.15 – 2008-08-15		\@gls@defaultsort: new	87
\@gls@link: added \glslabel	130	\@gls@hypernumber: new	262
\@gls@defglossaryentry: check for \@glo@first in description	105	\@glsnoname: new	86
check for \@glo@text in symbol	106	\@glsnonextpages: new	252
\@gls@hypergroup@rerun: new	298	\@gls@defglossaryentry: added nonumberlist key	102
\@glsnavhypertarget: added check if rerun required	297	added parent key	102
\@glssettoctitle: new	49	added see key	102
\printglossary: changed the way the TOC title is set	221	Stored main part of entry format when entry is defined	106
1.16 – 2008-08-27		\@gls@suffixF: new	54
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	145	\@gls@suffixFF: new	55
\@GLSpl: Test glossary type is \acronymtype in addition to checking if footnote option has been used	148	\@gls@wrglossary: modified to allow for xindy support	211
\@Gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	144	\@gls@hyperlink: new	183
		\@gls@hypernumber: modified to allow material to be attached to location	262

<code>\glsnavhyperlink</code> : replaced	1.46 – 2021-09-20
<code>\hyperlink</code> to <code>\@glslink</code>	295
<code>\glsnavhypertarget</code> : replaced	General: changed <code>\edef</code> to
<code>\hypertarget</code> with	<code>\protected@edef</code> 461
<code>\@glsstarget</code>	<code>\@printglossary</code> : changed <code>\xdef</code>
<code>\glssee</code> : new	to <code>\protected@xdef</code> 224
<code>\glsseeformat</code> : new	<code>\gls@defglossaryentry</code> : changed
<code>\glsSetSuffixF</code> : new	<code>\xdef</code> to
<code>\glsSetSuffixFF</code> : new	<code>\protected@xdef</code> 103, 106
<code>\ifglsxindy</code> : new	<code>\glsaddprotectedpagefmt</code> :
<code>\istfilename</code> : added xindy	changed <code>\eappto</code> to
support	<code>\protected@eappto</code> 212
<code>\newglossarystyle</code> : made	2.01 – 2009 May 30
<code>\newglossarystyle long</code>	<code>\@gls@link</code> : moved
<code>nonumberlist</code> : new	<code>\@do@wrglossary</code> before term
<code>\nopostdesc</code> : new	is displayed to prevent
<code>order</code> : order package option	unwanted whatsit 130
added	<code>\forallglossaries</code> : replaced
<code>parent</code> : new	<code>\ifthenelse</code> with <code>\ifx</code> 69
<code>\printglossary</code> : added check to	<code>\forglentries</code> : replaced
determine if <code>\printglossary</code>	<code>\ifthenelse</code> with <code>\ifx</code> 69
is already defined	<code>\glsdefmain</code> : new 20
added print language to aux	<code>\glsdescwidth</code> : changed
file	<code>\linewidth</code> to <code>\hsize</code> . 306, 330
<code>see</code> : new	<code>\glslistdottedwidth</code> : changed
<code>\writeist</code> : added xindy support 187	<code>\linewidth</code> to <code>\hsize</code> 305
1.18 – 2009-01-14	<code>\glspagelistwidth</code> : changed
<code>\@gls@loadlist</code> : new	<code>\linewidth</code> to <code>\hsize</code> . 306, 331
<code>\@gls@loadlong</code> : new	<code>nomain</code> : added <code>nomain</code> package
<code>\@gls@loadsuper</code> : new	option 21
<code>\@gls@loadtree</code> : new	<code>\writeist</code> : removed <code>item_02</code> -
<code>\gls@defglossaryentry</code> :	no such <code>makeindex</code> key 191
Changed default value of sort	2.02 – 2007-07-13
to <code>\@glsdefaultsort</code>	<code>\@printglossary</code> : suppressed
moved sort sanitization to	warning globally rather than
<code>\newglossaryentry</code>	locally 224
<code>\glsstarget</code> : new	2.02 – 2009-07-13
<code>nolist</code> : new	<code>\glossarysection</code> : changed
<code>nolong</code> : new	<code>\@mkboth</code> to <code>\glossarymark</code> 56
<code>nostyles</code> : new	<code>\gls glossarymark</code> : New 57
<code>nosuper</code> : new	2.03 – 2009-09-23
<code>notree</code> : new	<code>\@GLS@</code> : Added check for
<code>\oldacronym</code> : new	<code>hyperfirst</code> 145
<code>sort</code> : moved sanitization to	<code>\@GLSp1</code> : Added check for
<code>\newglossaryentry</code>	<code>hyperfirst</code> 148
1.19 – 2009-03-02	<code>\@GLS@</code> : Added check for
<code>\glsclearpage</code> : new	<code>hyperfirst</code> 144
<code>\glsdisp</code> : new	<code>\@GLSp1@</code> : Added check for
	<code>hyperfirst</code> 147
	<code>\@gls@</code> : Added check for
	<code>hyperfirst</code> 144

<code>\@gls@link</code> : new	128	<code>\gls@defglossaryentry</code> : added	
<code>\@gls@link</code> : added <code>\leavevmode</code>	130	user1-6 keys	102
Moved entry existence check to		<code>\glsadd</code> : fixed bug that ignored	
avoid duplicate code	130	counter	184
<code>\@glsdisp</code> : Added check for		<code>\Glsentryuseri</code> : new	179
<code>hyperfirst</code>	148	<code>\glsentryuseri</code> : new	179
<code>\@glspl@</code> : Added check for		<code>\Glsentryuserii</code> : new	179
<code>hyperfirst</code>	146	<code>\glsentryuserii</code> : new	179
<code>\gls glossarymark</code> : Added check		<code>\Glsentryuseriii</code> : new	179
to see if it's already defined	57	<code>\glsentryuseriii</code> : new	179
<code>hyperfirst</code> : new	34	<code>\Glsentryuseriv</code> : new	180
2.04 – 2009-11-10		<code>\glsentryuseriv</code> : new	179
<code>\@GLS@</code> : Changed test to check if		<code>\Glsentryuserv</code> : new	180
glossary type has been		<code>\glsentryuserv</code> : new	180
identified as a list of		<code>\Glsentryuservi</code> : new	180
acronyms	145	<code>\glsentryuservi</code> : new	180
<code>\@GLSpl@</code> : Changed test to check if		<code>\ns@newglossary</code> : added check to	
glossary type has been		determine if	
identified as a list of		<code>\gls@(<i>type</i>)@display</code> and	
acronyms	148	<code>\gls@(<i>type</i>)@displayfirst</code>	
<code>\@Gls@</code> : Changed test to check if		have been defined.	79
glossary type has been		<code>\SetAcronymLists</code> : new	23
identified as a list of		<code>\SetDefaultAcronymDisplayStyle</code> :	
acronyms	144	new	285
<code>\@Glspl@</code> : Changed test to check		<code>\SetDefaultAcronymStyle</code> : new	285
if glossary type has been		<code>\SetDescriptionAcronymDisplayStyle</code> :	
identified as a list of		new	286
acronyms	147	<code>\SetDescriptionDUAAcronymDisplayStyle</code> :	
<code>\@glossaryentryfield</code> : new	107	new	286
<code>\@glossarysubentryfield</code> : new	107	<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\@gls@</code> : Changed test to check if		new	286
glossary type has been		<code>\SetDUADisplayStyle</code> : new	287
identified as a list of		<code>\SetFootnoteAcronymDisplayStyle</code> :	
acronyms	144	new	287
<code>\@glsacronymlists</code> : new	22	<code>\SetSmallAcronymDisplayStyle</code> :	
<code>\@glsdisp</code> : Changed test to check		new	287
if glossary type has been		2.05 – 2010-02-06	
identified as a list of		<code>\@glsdisp</code> : Added closing brace.	
acronyms	148	Patch provided by Sergiu	
<code>\@glspl@</code> : Changed test to check		Dotenco	148
if glossary type has been		Removed spurious brace. Patch	
identified as a list of		provided by Sergiu Dotenco	148
acronyms	146	<code>\writeist</code> : Added <code>\string</code>	
<code>\@newglossaryentryposthook</code> :		before opening and closing	
new	107	braces. Patch provided by	
<code>\@newglossaryentryprehook</code> :		Sergiu Dotenco	192
new	107	2.06 – 2010-06-14	
<code>acronymlists</code> : new	23	<code>\altnewglossary</code> : new	79
<code>\DeclareAcronymList</code> : new	22	<code>\CustomAcronymFields</code> : new	288
<code>\DefineAcronymSynonyms</code> : new	283	<code>\CustomNewAcronymDef</code> : new	288

\SetCustomDisplayStyle: new .	288	added \glsnextpages	223
\SetCustomStyle: new	288	make toctitle default to title .	223
2.07 – 2010-07-10		\@xdyattributelist: new	60
General: glsadd format key stored		\acrfootnote: new	285
in \@glsnumberformat (was		\ACRfull: added starred version	269
mistakenly stored in		\Acrfull: added starred version	269
\@glo@format)	183	\acrfull: added starred version	268
3.0 – 2010-07-12		\ACRfullpl: added starred	
\@makeglossary: Added check for		version	270
savewrites	197	\Acrfullpl: added starred	
\@gls@wrglossary: modified to		version	269
take into account savewrites	211	\acrfullpl: added starred	
3.0 – 2010/03/31		version	269
\@set@glo@numformat: added 4th		\acrlinkfootnote: new	286
argument	132	\acrnofootnote: new	286
3.0 – 2011-04-02		compatible-2.07:	
General: added prefix to		compatible-2.07 option added	41
hyperlink	263	counter: replaced \@ifundefined	
etoolbox now loaded	6	with \ifcsundef	83
replaced \@ifundefined with		counterwithin: new	15
\ifcsundef	48, 51, 126, 250	entrycounter: new	15
\@do@esc@wrglossary: added		glossaryentry: new	15
check for hyper location		\glossarysection: replaced	
prefix	216	\@ifundefined with	
modified to use new format .	214	\ifcsundef	56
\@glossarysec: replaced		\glossarystyle: replaced	
\@ifundefined with		\@ifundefined with	
\ifcsundef	10	\ifcsundef	261
\@do@seeglossary: Sanitize and		glossarysubentry: new	15
escape cross-referencing		\gls@codepage: replaced	
information	218	\@ifundefined with	
\@gls@counterwithin: new	15	\ifcsundef	36
\@gls@ifinlist: new	60	\gls@defglossaryentry: added	
\@gls@link: added		\@gls@defsort	106
\@gls@saveentrycounter	130	added short and long keys	102
added \@gls@setsort	130	replaced \@ifundefined with	
\@gls@saveentrycounter: new	131	\ifcsundef	102
\@gls@setupsort@def: new	17	\gls@doclearpage: replaced	
\@gls@setupsort@standard: new	17	\@ifundefined with	
\@gls@setupsort@use: new	18	\ifcsundef	58
\@gls@xdy@locationlist: new	63	\glsadd: added	
\@glslink: replaced		\@gls@saveentrycounter	184
\@ifundefined with		\GlsAddXdyCounters: new	60
\ifcsundef	142	\glsentrycounterlabel: new	254
\@glsnextpages: new	252	\glsentryitem: new	254
\@print@glossary: replaced		\Glsentrylong: new	181
\@ifundefined with		\glsentrylong: new	181
\ifcsundef	225	\Glsentrylongpl: new	181
\@printglossary: added		\glsentrylongpl: new	181
\currentglossary	223	\Glsentryshort: new	180

<code>\glstryshort</code> : new	180	replaced <code>\@ifundefined</code> with	
<code>\Glsentryshortpl</code> : new	181	<code>\ifcsundef</code>	79
<code>\glstryshortpl</code> : new	180	<code>\oldacronym</code> : replaced	
<code>\glsgetgrouptitle</code> : replaced		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	265
<code>\ifcsundef</code>	259	<code>\printglossary</code> : replaced	
<code>\glsglossarymark</code> : replaced		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	221
<code>\ifcsundef</code>	57	<code>savewrites</code> : new	40
<code>\glshyperlink</code> : changed default		<code>see</code> : added	
from <code>\glstryname</code> to		<code>\@glo@seeautonumberlist</code>	83
<code>\glstrytext</code>	183	<code>seeautonumberlist</code> : new	13
<code>\glshyphnumber</code> : replaced		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\@ifundefined</code> with		expanded options link	
<code>\ifcsundef</code>	262	options	286
<code>\glsnumberformat</code> : replaced		<code>\setentrycounter</code> : added	
<code>\@ifundefined</code> with		optional argument	260
<code>\ifcsundef</code>	55	<code>short</code> : new	86
<code>\glsrefentry</code> : new	254	<code>shortplural</code> : new	86
<code>\glsresetsubentrycounter</code> :		<code>\showacronymlists</code> : new	293
new	253	<code>\showglocounter</code> : new	290
<code>\glsseeitem</code> : hyperlink uses		<code>\showgloDESC</code> : new	291
<code>\glsseeitemformat</code> instead of		<code>\showgloDESCplural</code> : new	292
<code>\glstryname</code>	220	<code>\showglofirst</code> : new	289
<code>\glsseeitemformat</code> : new	220	<code>\showglofirstpl</code> : new	290
<code>\glssortnumberfmt</code> : new	17	<code>\showgloflag</code> : new	293
<code>\glsstepentry</code> : new	253	<code>\showgloindex</code> : new	293
<code>\glsstepsubentry</code> : new	253	<code>\showglolevel</code> : new	289
<code>\glssubentrycounterlabel</code> :		<code>\showglongame</code> : new	291
new	254	<code>\showgloparent</code> : new	289
<code>\glssubentryitem</code> : new	254	<code>\showgloplural</code> : new	289
<code>\ifglossaryexists</code> : replaced		<code>\showglosort</code> : new	292
<code>\@ifundefined</code> with		<code>\showglossaries</code> : new	293
<code>\ifcsundef</code>	70	<code>\showglossarycounter</code> : new	294
<code>\ifglstryexists</code> : replaced		<code>\showglossaryentries</code> : new	294
<code>\@ifundefined</code> with		<code>\showglossaryin</code> : new	294
<code>\ifcsundef</code>	71	<code>\showglossaryout</code> : new	294
<code>\istfile</code> : deprecated	209	<code>\showglossarytitle</code> : new	294
<code>long</code> : new	86	<code>\showglosymbol</code> : new	292
<code>longplural</code> : new	86	<code>\showglosymbolplural</code> : new	292
<code>\newglossaryentry</code> : replaced		<code>\showglotext</code> : new	289
<code>\DeclareRobustCommand</code> with		<code>\showglotype</code> : new	290
<code>\newrobustcmd</code>	89	<code>\showglouserI</code> : new	290
<code>\newglossarystyle</code> : replaced		<code>\showglouserII</code> : new	290
<code>\@ifundefined</code> with		<code>\showglouserIII</code> : new	291
<code>\ifcsundef</code>	261	<code>\showglouserIV</code> : new	291
<code>nonumberlist</code> : now boolean	84	<code>\showglouserV</code> : new	291
<code>\ns@newglossary</code> : added		<code>\showglouserVI</code> : new	291
<code>\@gls@defsortcount</code>	79	<code>sort</code> : new	16
		<code>subentrycounter</code> : new	16

theglossary: replaced		\glsname: made robust	153
\@ifundefined with		\GLSpl: made robust	147
\ifcsundef	255	\Glspl: made robust	146
\writeist: added xindy-only		\glspl: made robust	145
macro definitions to glossary		\GLSplplural: made robust	152
open tag	189	\GLSsymbol: made robust	157
modified to support new		\Glsymbol: made robust	156
format	187	\glsymbol: made robust	156
3.01 – 2011-04-12		\GLSsymbolplural: made robust	158
General: made robust	145	\Glsymbolplural: made robust	157
\@glswritefiles: added check		\glsymbolplural: made robust	157
for empty glossaries	209	\GLstext: made robust	150
\ACRfull: made robust	269	\glstext: made robust	149
\Acrfull: made robust	268	\GLSuseri: made robust	159
\acrfull: made robust	268	\Glsuseri: made robust	158
\acrfullformat: removed		\glsuseri: made robust	158
\acronymfont as it should		\GLSuserii: made robust	160
already be set in the second		\Glsuserii: made robust	159
argument.	267	\glsuserii: made robust	159
\ACRfullpl: made robust	270	\GLSuseriii: made robust	161
\Acrfullpl: made robust	269	\Glsuseriii: made robust	160
\acrfullpl: made robust	269	\glsuseriii: made robust	160
\ACRlong: made robust	169	\GLSuseriv: made robust	162
\Acrlong: made robust	169	\Glsuseriv: made robust	161
\acrlong: made robust	168	\glsuseriv: made robust	161
\ACRlongpl: made robust	171	\GLSuseriv: made robust	163
\Acrlongpl: made robust	171	\Glsuseriv: made robust	162
\acrlongpl: made robust	170	\glsuseriv: made robust	162
\ACRshort: made robust	165	\GLSuservi: made robust	164
\Acrshort: made robust	165	\Glsuservi: made robust	163
\acrshort: made robust	164	\glsuservi: made robust	163
\ACRshortpl: made robust	167	3.02 – 2012-05-19	
\Acrshortpl: made robust	167	\glsnumlistlastsep: new	183
\acrshortpl: made robust	166	\glsnumlistsep: new	183
\GLs: made robust	144	3.02 – 2012-05-21	
\glsadd: made robust	184	\@do@wrglossary: changed	
\glsaddall: made robust	184	\@glslocref to	
\GLSdesc: made robust	155	\theglsentrycounter	217
\Glsdesc: made robust	154	\do@wrglossary: changed	
\glsdesc: made robust	154	\do@wr@glossary to test for	
\GLSdescplural: made robust	156	indexonlyfirst option; put old	
\Glsdescplural: made robust	155	\do@wr@glossary code into	
\glsdescplural: made robust	155	\@do@wrglossary	211
\glsfirst: made robust	150	\@gls@missingnumberlist: new	87
\GLSfirstplural: made robust	153	\@glswritefiles: added check	
\Glsfirstplural: made robust	152	for existence of token in case	
\glsfirstplural: made robust	152	\makeglossaries has been	
\glslink: made robust	128	omitted	209
\GLSname: made robust	154	\@printglossary: add a way to	
\Glsname: made robust	153	fetch current entry label	224

<code>\gls@defglossaryentry</code> : added numberlist element	105	<code>\glspostinline</code> : replaced “.” with <code>\glspostdescription</code>	301
<code>\gls@save@numberlist</code> : new . . .	220	index: added check for <code>glsnogroupskip</code>	408
<code>\gls@wrglossary</code> : added check for glossary file defined . . .	211	list: added check for <code>glsnogroupskip</code>	303
<code>\glsdisplaynumberlist</code> : new . .	182	long: added check for <code>glsnogroupskip</code>	307
<code>\glsentrycounter</code> : set default value	131	long3col: added check for <code>glsnogroupskip</code>	308
<code>\Glsentryfull</code> : fixed bug (replaced <code>\glsentryshortpl</code> with <code>\glsentryshort</code>)	181	long4col: added check for <code>glsnogroupskip</code>	310
<code>\glsentryfullpl</code> : fixed bug (replaced <code>\glsentryshort</code> with <code>\glsentryshortpl</code>) . .	181	longragged: added check for <code>glsnogroupskip</code>	318
<code>\glsentrynumberlist</code> : new . . .	182	longragged3col: added check for <code>glsnogroupskip</code>	319
<code>\glsmoveentry</code> : new	107	nogroupskip: new	14
<code>\glsresetentrycounter</code> : new . .	253	nopostdot: new	14
<code>\ifglshaschildren</code> : new	73	super: added check for <code>glsnogroupskip</code>	331
<code>\ifglshasparent</code> : new	73	super3col: added check for <code>glsnogroupskip</code>	333
indexonlyfirst: new	34	super4col: added check for <code>glsnogroupskip</code>	335
<code>\makeglossaries</code> : added list parser	202	superragged: added check for <code>glsnogroupskip</code>	339
<code>\renewglossarystyle</code> : new . . .	261	superragged3col: added check for <code>glsnogroupskip</code>	341
savenumberlist: new	12	tree: added check for <code>glsnogroupskip</code>	409
<code>\showglossaryentries</code> : fixed misspelt command	294	treenoname: added check for <code>glsnogroupskip</code>	411
<code>\SmallNewAcronymDef</code> : fixed broken short and long plural	287		
ucmark: new	14		
3.03 – 2012/09/21			
General: allow title to set toctitle	250		
<code>\@gls@sanitizesort</code> : new	26	3.04 – 2012-11-11	
<code>\@gls@setupsort@standard</code> : used <code>\@gls@sanitizesort</code>	17	<code>altlist</code> : replaced <code>\newline</code> with paragraph break	304
<code>\@printglossary</code> : allow title to override default toctitle . . .	223	3.04 – 2012-11-18	
<code>altlongragged4col</code> : added check for <code>glsnogroupskip</code>	321	General: Added check for doc package	6
<code>altsuperragged4col</code> : added check for <code>glsnogroupskip</code> . .	342	added datatool-base as a required package	52
<code>almtree</code> : added check for <code>glsnogroupskip</code>	414	added local key	127
<code>\glsinlinedescformat</code> : new . .	301	<code>\@do@wrglossary</code> : changed <code>\theglsentrycounter</code> back to <code>\@glslocref</code>	217
<code>\glsinlineemptydescformat</code> : new	301	<code>\@do@esc@wrglossary</code> : modified to compensate for possible incorrect page number	215
<code>\glsinlinenameformat</code> : new . .	301	<code>\@gls@esc@sdq</code> : unsanitize <code>\gls@numberpage</code> , <code>\gls@alphpage</code> ,	
<code>\glsinlinepostchild</code> : new . . .	301		
<code>\glsinlinesubdescformat</code> : new	301		
<code>\glsinlinesubnameformat</code> : new	301		

\gls@Alphpage and			
\gls@romanpage	133		
\@print@glossary: Moved aux			
write to end of document to			
prevent unwanted whatsit			
occurring here.	225		
\gls@Alphpage: new	212		
\gls@alphpage: new	212		
\gls@disablepagerefexpansion:			
new	212		
\gls@numberpage: new	212		
\gls@protected@pagefmts: new	211		
\gls@romanpage: new	212		
\glsdefmain: added check for doc			
package	20		
\glsorg@endtheglossary: new	6		
\glsorg@theglossary: new	6		
\PrintChanges: new	6		
3.05 – 2013-04-21			
\@do@esc@wrglossary: add			
Roman case. Fixed bugs in			
the else statements	216		
\@gls@link: added check for			
“nohypertypes”	130		
\gls@protected@pagefmts:			
added Roman to list	211		
\gls@Romanpage: new	212		
\glsgetgrouplabel: fixed bug			
(typo in \equal)	260		
\glsmcols: new	325		
mcolalttree: replaced ‘2’ with			
\glsmcols	329		
mcolindex: replaced ‘2’ with			
\glsmcols	325		
mcolindexspannav: replaced ‘2’			
with \glsmcols	326		
mcoltree: replaced ‘2’ with			
\glsmcols	326		
mcoltreename: replaced ‘2’			
with \glsmcols	328		
mcoltreepspannav: replaced ‘2’			
with \glsmcols	327		
\nopostdesc: made robust	53		
3.05 – 2013/04/21			
\@gls@nohyperlist: new	23		
\GlsDeclareNoHyperList: new	23		
nohypertypes: new	24		
3.06 – 2013/06/17			
\@xdy@main@language: Changed			
back to using \language	35		
\findrootlanguage: Obsoleted	67		
3.07 – 2013-07-05			
\@gls@link: fixed bug that failed			
to find entry in list	130		
\glossarypreamble: modified to			
work with			
\setglossarypreamble	56		
\gls@docclearpage: added check			
for openright	58		
\glspostdescription: Added			
spacefactor code	14		
\GlsSetXdyCodePage: Added			
check for fontspec	68		
\SetDescriptionAcronymDisplayStyle:			
now using			
\glsdoparenifnotempty	286		
\setglossarypreamble: new	56		
3.08a – 2013-08-30			
altlist: updated altlist style to			
use \glossentry and			
\subglossentry	304		
inline: updated inline style to			
use \glossentry and			
\subglossentry	299		
list: updated list style to use			
\glossentry and			
\subglossentry	303		
listdotted: updated listdotted			
style to use \glossentry and			
\subglossentry	305		
3.08a – 2013-09-28			
General: added nogroupskip key			
to \printglossary	251		
removed definition of			
\@glossaryentryfield	450		
removed definition of			
\@glossarysubentryfield	450		
\@glo@storeentry: no longer			
need to check for special			
characters in any of the fields			
other than sort	108		
updated for \glossentry	108		
\@glossaryentryfield: switched			
to \glossentry	107		
\@glossarysubentryfield:			
switched to \subglossentry	107		
\@gls@sanitizedesc: deprecated	25		
altlongragged4col: updated to			
use \glossentry and			
\subglossentry	321		

alttree: updated to use		superragged: updated to use	
\glossentry and		\glossentry and	
\subglossentry	413	\subglossentry	338
\compatibleglossentry: new ..	255	tree: updated to use	
\compatiblesubglossentry:		\glossentry and	
new	258	\subglossentry	409
\glossaryentryfield:		3.09a – 2013-10-09	
deprecated	258	\@gls@assign@symbolplural@field:	
\glossarystyle: deprecated ..	261	new	26
\glossentry: new	255	\@gls@default@value: new ...	82
\Glossentrydesc: new	256	\Glsentrydesc: made robust ..	175
\glossentrydesc: new	256	\Glsentrydescplural: made	
\Glossentryname: new	256	robust	176
\glossentryname: new	255	\Glsentryfirst: made robust ..	177
\Glossentrysymbol: new	257	\Glsentryfirstplural: made	
\glossentrysymbol: new	257	robust	178
\gls@assign@desc@field: new ..	26	\Glsentryfull: made robust ..	181
\gls@assign@descplural@field:		\Glsentryfullpl: made robust	182
new	26	\Glsentrylong: made robust ..	181
\gls@assign@field: new	89	\Glsentrylongpl: made robust	181
\gls@ifnotmeasuring: new ...	110	\Glsentryname: made robust ..	174
\glsaddallunused: new	185	\Glsentryplural: made robust	176
\glsexpandfields: new	89	\Glsentryshort: made robust ..	180
\glsnoexpandfields: new	89	\Glsentryshortpl: made robust	181
\glssee: made robust	219	\Glsentrysymbol: made robust	177
\glsseeformat: made robust ..	219	\Glsentrysymbolplural: made	
\glsseeitem: made robust	220	robust	177
\glsseelist: made robust	219	\Glsentrytext: made robust ..	176
\ifglsdescsuppressed: new ..	73	\Glsentryuseri: made robust ..	179
\ifglsdesc: new	73	\Glsentryuserii: made robust	179
\ifglsymbol: new	74	\Glsentryuseriii: made robust	179
index: added paragraph break at		\Glsentryuseriv: made robust	180
end of environment	407	\Glsentryuserv: made robust ..	180
updated to use \glossentry		\glstextup: new	266
and \subglossentry	407	\ifglsymbol: changed test	
long: updated to use		to check for	
\glossentry and		\@gls@default@symbol	74
\subglossentry	306	3.10a – 2013-09-28	
longragged: updated to use		\gls@assign@type@field: new ..	26
\glossentry and		3.10a – 2013-10-13	
\subglossentry	318	\@gls@keymap: new	92
longragged3col: updated to use		\@gls@provide@newglossary:	
\glossentry and		new	77
\subglossentry	319	\@gls@writedef: new	92
\setglossarystyle: new	261	\@glsdefaultplural: Obsolete ..	87
\setglossentrycompatibility:		\@glsnodesc: new	86
new	258	\@print@glossary: Added	
\subglossentry: new	258	providecommand code to aux	
		file	225

<code>\gls@defglossaryentry:</code>	
Changed to using	
<code>\@gls@default@value</code> .	101, 102
new	101
<code>\gls@writedefhook:</code> new	100
<code>\makeglossaries:</code> Added	
providecommand code to aux	
file	201
<code>\new@glossaryentry:</code> new	90
<code>\ns@newglossary:</code> added	
<code>\@gls@provide@newglossary</code>	79
3.11a – 2013-10-15	
General: added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	164–172
changed to just use	
<code>\Glsentrydescplural</code>	156
changed to just use	
<code>\glsentrydescplural</code>	155, 156
changed to just use	
<code>\Glsentrydesc</code>	155
changed to just use	
<code>\glsentrydesc</code>	154, 155
changed to just use	
<code>\Glsentryfirstplural</code>	153
changed to just use	
<code>\glsentryfirstplural</code>	152, 153
changed to just use	
<code>\Glsentryfirst</code>	151
changed to just use	
<code>\glsentryfirst</code>	150, 151
changed to just use	
<code>\Glsentryname</code>	154
changed to just use	
<code>\glsentryname</code>	153, 154
changed to just use	
<code>\Glsentryplural</code>	152
changed to just use	
<code>\glsentryplural</code>	151, 152
changed to just use	
<code>\Glsentrysymbolplural</code>	158
changed to just use	
<code>\glsentrysymbolplural</code>	157, 158
changed to just use	
<code>\Glsentrysymbol</code>	157
changed to just use	
<code>\glsentrysymbol</code>	156, 157
Changed to just use	
<code>\Glsentrytext</code>	150
changed to just use	
<code>\glsentrytext</code>	149
changed to just use	
<code>\Glsentryuseriii</code>	161
changed to just use	
<code>\glsentryuseriii</code>	160, 161
changed to just use	
<code>\Glsentryuserii</code>	160
changed to just use	
<code>\glsentryuserii</code>	159, 160
changed to just use	
<code>\Glsentryuseriv</code>	162
changed to just use	
<code>\glsentryuseriv</code>	161, 162
changed to just use	
<code>\Glsentryuseri</code>	159
changed to just use	
<code>\glsentryuseri</code>	158, 159
changed to just use	
<code>\Glsentryuservi</code>	164
changed to just use	
<code>\glsentryuservi</code>	163, 164
changed to just use	
<code>\Glsentryuserv</code>	163
changed to just use	
<code>\glsentryuserv</code>	162, 163
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>	445
<code>\@ACRshort:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	444
<code>\@Acrlong:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	445
<code>\@Acrshort:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	443
<code>\@GLS@:</code> add <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glscustomtext</code> and	
<code>\glsinsert</code>	145
change to using <code>\glsentryfmt</code>	
style commands	145

removed \MakeUppercase (now moved to \glentryfmt) ..	145	\@glspl@: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert	146
\@GLSpl: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert	147	change to using \glentryfmt style commands	146
change to using \glentryfmt style commands	148	acronymlists: replaced	
removed \MakeUppercase as now dealt with in \glentryfmt	148	\@addtoacronymlists with \DeclareAcronymList	23
\@Gls@: add \glsifplural, \glscapscase, \glscustomtext and \glsinsert	144	compatible-3.07: new	40
change to using \glentryfmt style commands	144	\defglsdisplay: obsolete	125
removed \makefirstuc (now dealt with in \glentryfmt)	144	\defglsdisplayfirst: obsolete	125
\@Glspl@: add \glsifplural, \glscapscase, \glscustomtext and \glsinsert	147	\defglentryfmt: new	77
change to using \glentryfmt style commands	147	\forglentries: replaced with \ifdefempty	69
removed \makefirstuc (now dealt with in \glentryfmt)	147	\gls@assign@desc: new	100
\@acrlong: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext	444	\gls@defglossaryentry: Fixed default counter if none supplied	105
\@acrshort: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext	443	\gls@doentryfmt: new	77
\@gls@: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert	143	\glsdisplay: obsolete	125
change to using \glentryfmt style commands	144	\glsdisplayfirst: obsolete ..	125
\@gls@noexpand@fields: Fixed bug expand replaced with noexpand	88	\glsentryfmt: new	120
\@glsdisp: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert	148	\glsgetgrouptitle: Added check in case non-Latin alphabet in use	259
change to using \glentryfmt style commands	148	\gls glossarymark: replaced \MakeUppercase with \mfirstucMakeUppercase ..	57
		\glsnavigation: switched to using \@gls@getgrouptitle	298
		\ifglshasdesc: replaced \ifdefempty with \ifcempty	73
		\ifglshaslong: new	74
		\ifglshasshort: new	75
		\ifglshassymbol: replaced \ifdefempty with \ifcempty	74
		\ifglsused: replaced \ifthenelse with \ifbool ..	71
		\longnewglossaryentry: new ..	100
		\ns@newglossary: replaced \glsdisplay and \glsdisplayfirst with \glentryfmt	79
		numbers: new	43
		\SetCustomDisplayStyle: updated to use \defglentryfmt	288

<code>\SetDefaultAcronymDisplayStyle:</code>	<code>\gls@assign@desc@field:</code>
changed to use	changed to use
<code>\defglsentryfmt</code> 285	<code>\glssetnoexpandfield</code> 26
<code>\SetDescriptionAcronymDisplayStyle:</code>	<code>\gls@assign@descplural@field:</code>
updated to use	changed to use
<code>\defglsentryfmt</code> 286	<code>\glssetnoexpandfield</code> 26
<code>\SetDescriptionDUAAcronymDisplayStyle:</code>	<code>\gls@assign@name@field:</code>
updated to use	changed to use
<code>\defglsentryfmt</code> 286	<code>\glssetnoexpandfield</code> 26
<code>\SetDescriptionFootnoteAcronymDisplayStyle:</code>	<code>\gls@assign@type@field:</code>
updated to use	changed to use
<code>\defglsentryfmt</code> 286	<code>\glssetexpandfield</code> 26
<code>\SetDUADisplayStyle:</code> updated	<code>\gls@checkseeallowed:</code> new .. 84
to use <code>\defglsentryfmt</code> .. 287	<code>\glsaddallunused:</code> set default to
<code>\SetFootnoteAcronymDisplayStyle:</code>	<code>\@glo@types</code> 185
updated to use	<code>\Glsentryfull:</code> changed to use
<code>\defglsentryfmt</code> 287	<code>\acrfullformat</code> 181
<code>\SetSmallAcronymDisplayStyle:</code>	<code>\glsentryfull:</code> changed to use
updated to use	<code>\acrfullformat</code> 181
<code>\defglsentryfmt</code> 287	<code>\Glsentryfullpl:</code> changed to use
<code>\setupglossaries:</code> new 47	<code>\acrfullformat</code> 182
<code>\showglolong:</code> new 292	<code>\glsentryfullpl:</code> changed to use
<code>\showgloshort:</code> new 292	<code>\acrfullformat</code> 181
<code>symbols:</code> new 42	<code>\gls glossarymark:</code> renamed
3.12a – 2013-10-16	<code>\glossarymark</code> to
<code>\gls@defglossaryentry:</code> added	<code>\gls glossarymark</code> to avoid
<code>\glslabel</code> 101	conflict with memoir 57
<code>\glsaddkey:</code> new 94	<code>\glsprestandardsort:</code> new ... 16
3.13a – 2013-11-05	<code>\glssetexpandfield:</code> new 25
<code>\@gls@assign@symbol@field:</code>	<code>\glssetnoexpandfield:</code> new .. 25
changed to use	long: switched to
<code>\glssetnoexpandfield</code> 26	<code>\tabularnewline</code> 306, 307
<code>\@gls@assign@symbolplural@field:</code>	long3col: switched to
changed to use	<code>\tabularnewline</code> 308
<code>\glssetnoexpandfield</code> 26	long3colheader: switched to
<code>\@gls@link:</code> removed <code>\relax</code> .. 130	<code>\tabularnewline</code> 309
<code>\@gls@notranslatorhook:</code> new . 31	long3colheaderborder: switched
<code>\@gls@setupsort@standard:</code>	to <code>\tabularnewline</code> 309
moved <code>\@gls@santizesort</code> to	long4col: switched to
<code>\glsprestandardsort</code> 17	<code>\tabularnewline</code> 309
altsuper4colheader: switched to	long4colheader: switched to
<code>\tabularnewline</code> 336	<code>\tabularnewline</code> 310
altsuper4colheaderborder:	longheader: switched to
switched to	<code>\tabularnewline</code> 307
<code>\tabularnewline</code> 337	longheaderborder: switched to
<code>\glossarysection:</code> changed	<code>\tabularnewline</code> 307
<code>\glossarymark</code> to	see: added
<code>\gls glossarymark</code> 57	<code>\gls@checkseeallowed</code> 83
	super: switched to
	<code>\tabularnewline</code> 331

super3col: switched to		\@acrfull: now using	
\tabularnewline	333	\acrfullfmt	268
super3colheader: switched to		\@gls@indexdef: new	43
\tabularnewline	334	\@gls@numbersdef: new	43
super4col: switched to		\@gls@symbolsdef: new	42
\tabularnewline	335	\ACRfullfmt: new	268
super4colheader: switched to		\Acrfullfmt: new	268
\tabularnewline	335	\acrfullfmt: new	267
super4colheaderborder:		\ACRfullplfmt: new	268
switched to		\Acrfullplfmt: new	268
\tabularnewline	336	\acrfullplfmt: new	268
superheader: switched to		\acronymentry: new	272
\tabularnewline	332	dua: new	278
superheaderborder: switched to		dua-desc: new	280
\tabularnewline	332	footnote: new	281
ucmark: added check for memoir	14	footnote-desc: new	282
3.14a – 2013-11-12		footnote-sc: new	282
General: new	417	footnote-sc-desc: new	283
\@glswritefiles: renamed		footnote-sm: new	282
\glswritefiles to		footnote-sm-desc: new	283
\@glswritefiles and used		\Genacrfullformat: new	125
“savewrites” option to set		\genacrfullformat: new	124
\glswritefiles	209	\GenericAcronymFields: new	272
acronyms: new	21	\Genplacrfullformat: new	125
compatible-2.07: added check		\genplacrfullformat: new	125
for 2.07 before setting 3.07		\Glsentryfull: bug fix: added	
compatibility	41	missing \acronymfont	181
\@gls@defglossaryentry: added		\Glsentryfull: bug fix: added	
check for existence of default		missing \acronymfont	181
glossary	102	\Glsentryfullpl: bug fix: added	
set the default for firstplural to		missing \acronymfont	182
be the value of plural	104	\Glsentryfullpl: bug fix: added	
\longprovideglossaryentry:		missing \acronymfont	181
new	101	\glsacrfmt: new	122
notranslate: new	32	\GlsUseAcrEntryDisplayStyle:	
\provideglossaryentry: new	89	new	273
xindygloss: new	36	\GlsUseAcrStyleDefs: new	273
4.0 – 2013-11-14		index: new	43
\@gls@defglossaryentry: added		long-sc-short: new	275
check for first key	104	long-sc-short-desc: new	277
super: fixed typo in		long-short: new	273
\subglossentry		long-short-desc: new	276
(\glossentrydesc)	331	long-sm-short: new	276
4.01 – 2013-11-16		long-sm-short-desc: new	277
General: fixed non-value options		\newacronymstyle: new	273
so that they can be passed to		numberedsection: added nameref	
document class	12	option	11
4.02 – 2013-12-05		sanitize: fixed bug that caused	
General: Removed		an error here	31
\acronymfont	168–172	sc-short-long: new	276

sc-short-long-desc: new	278	moved \@gls@link@opts and	
\setacronymstyle: new	272	\@gls@link@label to	
\SetGenericNewAcronym: new	271	\@gls@link	130
short-long: new	275	\@gls@writedef: added	
short-long-desc: new	278	\glsdetoklabel	92
sm-short-long: new	276	\@glsdisp: removed \glslabel	
sm-short-long-desc: new	278	(defined in \@gls@link) . .	148
xindynoglsnumbers: new	36	\@glspl@: removed \glslabel	
4.02 – 2013-13-05		(defined in \@gls@link) . .	146
\makeglossaries: made		\@printglossary: added	
preamble only	202	\glsdetoklabel	224
4.03 – 2014-01-17		\compatibleglossentry: added	
General: changed default to		\glsdetoklabel	425
\@empty instead of \relax . .	40	\compatiblesubglossentry:	
4.03 – 2014-01-20		added \glsdetoklabel	425
General: removed \glslabel		dua: fixed bug in \acrfullfmt . .	280
(defined in \@gls@link) . .	164	fixed bug in \Acrfullplfmt . .	280
\@do@esc@wrglossary: added		fixed bug in \acrfullplfmt . .	280
\glsdetoklabel	217	redefined to use accessibility	
\@do@noesc@wrglossary: added		information	455
\glsdetoklabel	214	dua-desc: commented spurious	
\@ACRlong: removed \glslabel		EOL	281
(defined in \@gls@link) . .	445	redefined to use accessibility	
\@ACRshort: removed \glslabel		information	457
(defined in \@gls@link) . .	444	footnote: redefined to use	
\@Acrlong: removed \glslabel		accessibility information . .	458
(defined in \@gls@link) . .	445	footnote-desc: redefined to use	
\@Acrshort: removed \glslabel		accessibility information . .	460
(defined in \@gls@link) . .	443	footnote-sc: redefined to use	
\@GLS@: removed \glslabel		accessibility information . .	459
(defined in \@gls@link) . .	145	footnote-sc-desc: redefined to	
\@GLSpl: removed \glslabel		use accessibility information	460
(defined in \@gls@link) . .	147	footnote-sm: redefined to use	
\@Gls@: removed \glslabel		accessibility information . .	460
(defined in \@gls@link) . .	144	footnote-sm-desc: redefined to	
\@Gls@entry@field: new	173	use accessibility information	460
\@Glspl@: removed \glslabel		\Genacrfullformat: redefined to	
(defined in \@gls@link) . .	147	use accessibility information	442
\@acrlong: removed \glslabel		\genacrfullformat: redefined to	
(defined in \@gls@link) . .	444	use accessibility information	442
\@acrshort: removed \glslabel		\Genplacrfullformat: redefined	
(defined in \@gls@link) . .	443	to use accessibility	
\@gls@: removed \glslabel		information	443
(defined in \@gls@link) . .	143	\genplacrfullformat: redefined	
\@gls@access@display: new . .	434	to use accessibility	
\@gls@entry@field: new	172	information	443
\@gls@fetchfield: new	93	\glossentryname: added	
\@gls@field@link: new	149	\glsdetoklabel	255
\@gls@link: added		\gls@defglossaryentry: added	
\glsdetoklabel	130	\glsdetoklabel	101

replaced #1 with		\glslocalreset: added	
\@glo@label	102	\glsdetoklabel	112
replaced \ifthenelse with		\glslocalunset: added	
\ifdefequal	103	\glsdetoklabel	112
\glsadd: added \glsdetoklabel	184	\glsmoveentry: added	
\glsaddkey: switched to using		\glsdetoklabel	107
\@gls@field@link	95	replaced \ifthenelse with	
\glsdetoklabel: new	71	\ifdefequal	107
\glsdisplaynumberlist: added		\glsrefentry: added	
\glsdetoklabel	182	\glsdetoklabel	254
\glsdoifexistsorwarn: new . .	72	\glsreset: added	
\glsentryaccess: switched to		\glsdetoklabel	111
using \@gls@entry@field .	430	\glsseelist: added	
\glsentrydescaccess: switched		\expandafter commands .	220
to using \@gls@entry@field	431	\glsstepentry: added	
\glsentrydescpluralaccess:		\glsdetoklabel	253
switched to using		\glsstepsubentry: added	
\@gls@entry@field	431	\glsdetoklabel	253
\glsentryfirstaccess: switched		\glsunset: added	
to using \@gls@entry@field	430	\glsdetoklabel	112
\glsentryfirstplural: added		\ifglsdescsuppressed: added	
\glsdetoklabel	177	\glsdetoklabel	73
\glsentrylongaccess: switched		fixed typo	73
to using \@gls@entry@field	431	\ifglsentryexists: added	
\glsentrylongpluralaccess:		\glsdetoklabel	71
switched to using		\ifglschaschildren: added	
\@gls@entry@field	432	\glsdetoklabel	73
\glsentrypluralaccess:		\ifglschasdesc: added	
switched to using		\glsdetoklabel	73
\@gls@entry@field	431	\ifglschasfield: new	75
\glsentryshortaccess: switched		\ifglschaslong: added	
to using \@gls@entry@field	431	\glsdetoklabel	74
\glsentryshortpluralaccess:		\ifglschasparent: added	
switched to using		\glsdetoklabel	73
\@gls@entry@field	431	\ifglschasshort: added	
\glsentrysymbolaccess:		\glsdetoklabel	75
switched to using		\ifglschassymbol: added	
\@gls@entry@field	431	\glsdetoklabel	74
\glsentrysymbolpluralaccess:		replaced \ifcsemtyp with	
switched to using		\ifdefempty and replaced	
\@gls@entry@field	431	\ifx with \ifdefequal	74
\glsentrytextaccess: switched		\ifglsused: added	
to using \@gls@entry@field	430	\glsdetoklabel	71
\glsgenacfmt: redefined to use		long-sc-short-desc: redefined	
accessibility information . .	440	to use accessibility	
\glsgenentryfmt: redefined to		information	453
use accessibility information	437	long-short: redefined to use	
\glshyperlink: added		accessibility information . .	452
\glsdetoklabel	183	long-short-desc: redefined to	
		use accessibility information	453

long-sm-short-desc: redefined to use accessibility information	454	\showgloplural: added	
\renewacronymstyle: new	273	\glsdetoklabel	289
sc-short-long-desc: redefined to use accessibility information	454	\showglopluralaccess: added	
short-long: commented spurious EOL	275	\glsdetoklabel	462
redefined to use accessibility information	452	\showgloshort: added	
short-long-desc: redefined to use accessibility information	454	\glsdetoklabel	292
\showglocounter: added		\showgloshortaccess: added	
\glsdetoklabel	290	\glsdetoklabel	463
\showglodesc: added		\showgloshortpluralaccess: added	
\glsdetoklabel	291	\glsdetoklabel	463
\showglodescaccess: added		\showglosort: added	
\glsdetoklabel	462	\glsdetoklabel	292
\showglodescplural: added		\showglosymbol: added	
\glsdetoklabel	292	\glsdetoklabel	292
\showglodescpluralaccess: added		\showglosymbolaccess: added	
\glsdetoklabel	462	\glsdetoklabel	462
\showglofirst: added		\showglosymbolplural: added	
\glsdetoklabel	289	\glsdetoklabel	292
\showglofirstaccess: added		\showglosymbolpluralaccess: added	
\glsdetoklabel	462	\glsdetoklabel	462
\showglofirstpl: added		\showglotext: added	
\glsdetoklabel	290	\glsdetoklabel	289
\showglofirstpluralaccess: added		\showglotextaccess: added	
\glsdetoklabel	462	\glsdetoklabel	462
\showgloflag: added		\showglotype: added	
\glsdetoklabel	293	\glsdetoklabel	290
\showgloindex: added		\showglouseri: added	
\glsdetoklabel	293	\glsdetoklabel	290
\showglolevel: added		\showglouserii: added	
\glsdetoklabel	289	\glsdetoklabel	290
\showglolong: added		\showglouseriii: added	
\glsdetoklabel	292	\glsdetoklabel	291
\showglolongaccess: added		\showglouseriv: added	
\glsdetoklabel	463	\glsdetoklabel	291
\showglolongpluralaccess: added		\showglouserv: added	
\glsdetoklabel	463	\glsdetoklabel	291
\showgloname: added		\showglouservi: added	
\glsdetoklabel	291	\glsdetoklabel	291
\showglonameaccess: added		sm-short-long-desc: redefined to use accessibility information	455
\glsdetoklabel	462	4.04 – 2014-03-04	
\showgloparent: added		\@gls@getcounterprefix: added	
\glsdetoklabel	289	warning if no prefix can be formed	218
		4.04 – 2014-03-06	
		General: added sort key to printgloss group	252
		\@@gls@noidx@nosanitizesort: new	27

<code>\@gls@noidx@sanitizesort:</code>		<code>\gls@defglossaryentry:</code>	
new	26	replaced <code>\ifx</code> with	
<code>\@gls@nosanitizesort:</code> new	26	<code>\ifdefvoid</code>	106
<code>\@gls@sanitizesort:</code> new	26	<code>\glsnoidxdisplayloc:</code> new	247
<code>\@glo@addchildren:</code> new	234	<code>\glsnoidxdisplayloclisthandler:</code>	
<code>\@glo@do@sortentries:</code> new	235	new	247
<code>\@glo@grabfirst:</code> new	245	<code>\glsnoidxloclist:</code> new	246
<code>\@glo@sortedinsert:</code> new	235	<code>\glsnoidxloclisthandler:</code> new	247
<code>\@glo@sortentries:</code> new	234	<code>\glsnoidxstripaccents:</code> new	28
<code>\@glo@sorthandler@case:</code> new	236	<code>\makeglossaries:</code> Moved	
<code>\@glo@sorthandler@letter:</code>		definition of <code>\glswrite</code> to	
new	236	<code>\makeglossaries</code>	201
<code>\@glo@sorthandler@nocase:</code>		<code>\makenoidxglossaries:</code> new	203
new	236	<code>\printglossary:</code> changed to use	
<code>\@glo@sorthandler@word:</code> new	236	new <code>\@printglossary</code>	222
<code>\@glo@sortmacro@case:</code> new	238	<code>\printnoidxglossaries:</code> new	222
<code>\@glo@sortmacro@def:</code> new	238	<code>\printnoidxglossary:</code> new	222
<code>\@glo@sortmacro@def@do:</code> new	239	<code>\showgloclist:</code> new	293
<code>\@glo@sortmacro@letter:</code> new	237	<code>\warn@noprintglossary:</code>	
<code>\@glo@sortmacro@nocase:</code> new	238	Activate warning in	
<code>\@glo@sortmacro@standard:</code>		<code>\makeglossaries</code>	221
new	238	<code>\writeist:</code> checked for definition	
<code>\@glo@sortmacro@use:</code> new	239	of <code>\glswrite</code>	187, 191
<code>\@glo@sortmacro@word:</code> new	237	4.06 – 2014-03-12	
<code>\@gls@noidx@do:</code> new	245	General: added	
<code>\@gls@noidx@getgrouptitle:</code>		<code>\glsifhyper</code>	165–172
new	259	<code>\@GLS@:</code> added <code>\glsifhyper</code>	145
<code>\@gls@noref@warn:</code> new	207	<code>\@GLSpl:</code> added <code>\glsifhyper</code>	148
<code>\@gls@reference:</code> new	247	<code>\@Gls@:</code> added <code>\glsifhyper</code>	144
<code>\@gls@warnonglossdefined:</code> new	25	<code>\@Glspl@:</code> added <code>\glsifhyper</code>	147
<code>\@gls@warnontheglossdefined:</code>		<code>\@gls@:</code> added <code>\glsifhyper</code>	144
new	25	<code>\@gls@numbersdef:</code> added hook	
<code>\@no@makeglossaries:</code> new	206	to set toc title	43
<code>\@print@glossary:</code> new	224	<code>\@gls@symbolsdef:</code> added hook	
<code>\@print@noidx@glossary:</code> new	239	to set toc title	43
<code>\@printgloss@setsort:</code> new	222	<code>\@glsdisp:</code> added <code>\glsifhyper</code>	148
<code>\@printglossary:</code> new	222	<code>\@glspl@:</code> added <code>\glsifhyper</code>	146
<code>almtree:</code> moved hangindent and		acronym: added hook to set toc	
parindent assignments outside		title	21
level test	413	acronyms: added hook to set toc	
<code>\compatibleglossentry:</code> changed		title	22
<code>\newcommand</code> to <code>\def</code> as is		<code>\glsdefmain:</code> added hook to set	
may or may not be defined	425	toc title	20
<code>\compatiblesubglossentry:</code>		4.07 – 2014-04-04	
changed <code>\newcommand</code> to <code>\def</code>		<code>\@glossarysection:</code> added	
as is may or may not be		optional argument when using	
defined	425	unstarred version	58
<code>\glo@grabfirst:</code> new	244	<code>\@gls@noidx@do:</code> added <code>\global</code>	
		in case it's used in a	
		tabular-like style	245

<code>\Acrfullplfmt</code> : fixed no case	
change bug	268
<code>\glsletentryfield</code> : new	172
4.08 – 2014-07-30	
General: added <code>entrycounter</code>	
option to <code>printgloss</code> family	251
added <code>nopostdot</code> option to	
<code>printgloss</code> family	251
added <code>subentrycounter</code> option	
to <code>printgloss</code> family	251
explicitly initialise <code>hyper</code> key	126
moved <code>\glsifhyper</code>	165–172
removed <code>\@sACRlongpl</code>	171
removed <code>\@sAcrlongpl</code>	171
removed <code>\@sacrlongpl</code>	170
removed <code>\@sACRlong</code>	169
removed <code>\@sAcrlong</code>	169
removed <code>\@sacrlong</code>	168
removed <code>\@sACRshortpl</code>	167
removed <code>\@sAcrshortpl</code>	167
removed <code>\@sacrshortpl</code>	166
removed <code>\@sACRshort</code>	165
removed <code>\@sAcrshort</code>	165
removed <code>\@sacrshort</code>	164
removed <code>\@gls@link</code>	128
removed <code>\@sGLSdescplural</code>	156
removed <code>\@sGlsdescplural</code>	156
removed <code>\@sglsdescplural</code>	155
removed <code>\@sGLSdesc</code>	155
removed <code>\@sGlsdesc</code>	155
removed <code>\@sglsdesc</code>	154
removed <code>\@sglsdisp</code>	148
removed <code>\@sGLSfirstplural</code>	153
removed <code>\@sGlsfirstplural</code>	153
removed <code>\@sglsfirstplural</code>	152
removed <code>\@sGLSfirst</code>	151
removed <code>\@sGlsfirst</code>	151
removed <code>\@sglsfirst</code>	150
removed <code>\@sGLSname</code>	154
removed <code>\@sGlsname</code>	154
removed <code>\@sglsname</code>	153
removed <code>\@sGLSplural</code>	152
removed <code>\@sGlsplural</code>	152
removed <code>\@sglsplural</code>	151
removed <code>\@sGLSpl</code>	147
removed <code>\@sGlspl</code>	146
removed <code>\@sglspl</code>	146
removed <code>\@sGLSsymbolplural</code>	158
removed <code>\@sGlsymbolplural</code>	158
removed <code>\@sglsymbolplural</code>	157
removed <code>\@sGLSsymbol</code>	157
removed <code>\@sGlsymbol</code>	157
removed <code>\@sglsymbol</code>	156
removed <code>\@sGLStext</code>	150
removed <code>\@sGlstext</code>	150
removed <code>\@sglstext</code>	149
removed <code>\@sGLSuseriii</code>	161
removed <code>\@sGlsuseriii</code>	161
removed <code>\@sglsuseriii</code>	160
removed <code>\@sGLSuserii</code>	160
removed <code>\@sGlsuserii</code>	160
removed <code>\@sglsuserii</code>	159
removed <code>\@sGLSuseriv</code>	162
removed <code>\@sGlsuseriv</code>	162
removed <code>\@sglsuseriv</code>	161
removed <code>\@sGLSuseri</code>	159
removed <code>\@sGlsuseri</code>	159
removed <code>\@sglsuseri</code>	158
removed <code>\@sGLSuservi</code>	164
removed <code>\@sGlsuservi</code>	164
removed <code>\@sglsuservi</code>	163
removed <code>\@sGLSuserv</code>	163
removed <code>\@sGlsuserv</code>	163
removed <code>\@sglsuserv</code>	162
removed <code>\@sGLS</code>	145
removed <code>\@sGls</code>	144
removed <code>\@sgls</code>	143
removed <code>\@thirdofthree</code>	
(defined in kernel)	143
removed <code>sPGLS</code>	423
removed <code>sPglS</code>	421
removed <code>spglS</code>	420
removed <code>sPGLSpl</code>	423
removed <code>sPglSpl</code>	422
removed <code>spglSpl</code>	420
<code>\@ACRlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	445
<code>\@ACRshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	444
<code>\@Acrlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	444
<code>\@Acrshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	443
<code>\@GLS@</code> : moved <code>\glsifhyper</code>	145
moved check for first use to	
<code>\@gls@link</code>	145

\@GLSp1: moved \glsifhyper ..	148	\ACRfullpl: removed	
moved check for first use to		\s@ACRfullpl	270
\@gls@link	148	switched to using	
\@Gls0: moved \glsifhyper ...	144	\@gls@hyp@opt	270
moved check for first use to		\Acrfullpl: removed	
\@gls@link	144	\s@Acrfullpl	269
\@Glspl0: moved \glsifhyper .	147	switched to using	
moved check for first use to		\@gls@hyp@opt	269
\@gls@link	147	\acrfullpl: removed	
\@acrlong: added		\s@acrfullpl	269
\do@gls@link@checkfirsthyper		switched to using	
.....	444	\@gls@hyp@opt	269
\@acrshort: added		\ACRlong: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	169
.....	443	\Acrlong: switched to using	
\@closegls: new	197	\@gls@hyp@opt	169
\@gls0: moved \glsifhyper ...	144	\acrlong: switched to using	
moved check for first use to		\@gls@hyp@opt	168
\@gls@link	144	\ACRlongpl: switched to using	
\@gls@automake: new	197	\@gls@hyp@opt	171
\@gls@doautomake: new	40	\Acrlongpl: switched to using	
\@gls@field@link: added		\@gls@hyp@opt	171
assignment of		\acrlongpl: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	170
.....	149	\ACRshort: switched to using	
\@gls@forbidtextext: new	77	\@gls@hyp@opt	165
\@gls@hyp@opt: new	127	\Acrshort: switched to using	
\@gls@link: removed		\@gls@hyp@opt	165
redundancy	130	\acrshort: switched to using	
renamed \gls@type to		\@gls@hyp@opt	164
\glstype	130	\ACRshortpl: switched to using	
\@gls@link@checkfirsthyper:		\@gls@hyp@opt	167
new	129	\Acrshortpl: switched to using	
\@glsdisp: moved \glsifhyper	148	\@gls@hyp@opt	167
moved check for first use to		\acrshortpl: switched to using	
\@gls@link	148	\@gls@hyp@opt	166
\@glspl0: moved \glsifhyper .	146	altlongragged4col: fixed bug	
moved check for first use to		that displayed description	
\@gls@link	146	instead of symbol	321
\@ignored@glossaries: new ..	81	automake: new	39
\ACRfull: removed \s@ACRfull	269	\forallacronyms: new	69
switched to using		\GLS: switched to using	
\@gls@hyp@opt	269	\@gls@hyp@opt	145
\Acrfull: removed \s@Acrfull	269	\Gls: switched to using	
switched to using		\@gls@hyp@opt	144
\@gls@hyp@opt	268	\gls: switched to using	
\acrfull: removed \s@acrfull	268	\@gls@hyp@opt	143
switched to using		\gls@defglossaryentry: added	
\@gls@hyp@opt	268	check for ignored glossary .	103
		\gls@istfilebase: new	53

<code>\glsaddkey: removed</code>		<code>\glsname: switched to using</code>	
<code>\@sGLS@user@{key}</code>	96	<code>\@gls@hyp@opt</code>	153
removed <code>\@sGls@user@{key}</code>	96	<code>\GLSpl: switched to using</code>	
removed <code>\@sgls@user@{key}</code>	95	<code>\@gls@hyp@opt</code>	147
switched to using		<code>\Glspl: switched to using</code>	
<code>\@gls@hyp@opt</code>	95, 96	<code>\@gls@hyp@opt</code>	146
<code>\GLSdesc: switched to using</code>		<code>\glspl: switched to using</code>	
<code>\@gls@hyp@opt</code>	155	<code>\@gls@hyp@opt</code>	145
<code>\Glsdesc: switched to using</code>		<code>\GLSplural: switched to using</code>	
<code>\@gls@hyp@opt</code>	154	<code>\@gls@hyp@opt</code>	152
<code>\glsdesc: switched to using</code>		<code>\Glsplural: switched to using</code>	
<code>\@gls@hyp@opt</code>	154	<code>\@gls@hyp@opt</code>	151
<code>\GLSdescplural: switched to</code>		<code>\glsplural: switched to using</code>	
using <code>\@gls@hyp@opt</code>	156	<code>\@gls@hyp@opt</code>	151
<code>\Glsdescplural: switched to</code>		<code>\glspace: new</code>	267
using <code>\@gls@hyp@opt</code>	155	<code>\GLSsymbol: switched to using</code>	
<code>\glsdescplural: switched to</code>		<code>\@gls@hyp@opt</code>	157
using <code>\@gls@hyp@opt</code>	155	<code>\Glsymbol: switched to using</code>	
<code>\glsdisablehyper: added</code>		<code>\@gls@hyp@opt</code>	156
<code>\KV@glslink@hyperfalse</code> to		<code>\glsymbol: switched to using</code>	
definition	142	<code>\@gls@hyp@opt</code>	156
<code>\glsdisp: switched to using</code>		<code>\GLSsymbolplural: switched to</code>	
<code>\@gls@hyp@opt</code>	148	using <code>\@gls@hyp@opt</code>	158
<code>\glsdohyperlink: new</code>	142	<code>\Glsymbolplural: switched to</code>	
<code>\glsdohypertarget: new</code>	140	using <code>\@gls@hyp@opt</code>	157
<code>\glsenablehyper: added</code>		<code>\glsymbolplural: switched to</code>	
<code>\KV@glslink@hypertrue</code> to		using <code>\@gls@hyp@opt</code>	157
definition	143	<code>\GLStext: switched to using</code>	
<code>\GLSfirst: switched to using</code>		<code>\@gls@hyp@opt</code>	149
<code>\@gls@hyp@opt</code>	151	<code>\Glstext: switched to using</code>	
<code>\Glsfirst: switched to using</code>		<code>\@gls@hyp@opt</code>	150
<code>\@gls@hyp@opt</code>	150	<code>\glstext: switched to using</code>	
<code>\glsfirst: switched to using</code>		<code>\@gls@hyp@opt</code>	149
<code>\@gls@hyp@opt</code>	150	<code>\glstreenamfmt: new</code>	406
<code>\GLSfirstplural: switched to</code>		<code>\GLSuseri: switched to using</code>	
using <code>\@gls@hyp@opt</code>	153	<code>\@gls@hyp@opt</code>	159
<code>\Glsfirstplural: switched to</code>		<code>\Glsuseri: switched to using</code>	
using <code>\@gls@hyp@opt</code>	152	<code>\@gls@hyp@opt</code>	158
<code>\glsfirstplural: switched to</code>		<code>\glsuseri: switched to using</code>	
using <code>\@gls@hyp@opt</code>	152	<code>\@gls@hyp@opt</code>	158
<code>\glsifhyper: deprecated</code>	127	<code>\GLSuserii: switched to using</code>	
<code>\glslink: switched to using</code>		<code>\@gls@hyp@opt</code>	160
<code>\@gls@hyp@opt</code>	128	<code>\Glsuserii: switched to using</code>	
<code>\glslinkcheckfirsthyperhook:</code>		<code>\@gls@hyp@opt</code>	159
new	129	<code>\glsuserii: switched to using</code>	
<code>\glslinkvar: new</code>	127	<code>\@gls@hyp@opt</code>	159
<code>\GLSname: switched to using</code>		<code>\GLSuseriii: switched to using</code>	
<code>\@gls@hyp@opt</code>	154	<code>\@gls@hyp@opt</code>	161
<code>\Glsname: switched to using</code>		<code>\Glsuseriii: switched to using</code>	
<code>\@gls@hyp@opt</code>	153	<code>\@gls@hyp@opt</code>	160

<code>\glsuseriii</code> : switched to using <code>\@gls@hyp@opt</code>	160	<code>\@gls@glossary</code> : Renamed <code>\@glossary</code> to <code>\@gls@glossary</code>	210
<code>\GLSuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	162	<code>alttree</code> : moved space after symbol	413, 414
<code>\Glsuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	161	<code>\glspercentchar</code> : new	186
<code>\glsuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	161	<code>\glstildechar</code> : new	186
<code>\GLSuserv</code> : switched to using <code>\@gls@hyp@opt</code>	163	4.11 – 2014-09-01 <code>\@do@esc@wrglossary</code> : added hook	216
<code>\Glsuserv</code> : switched to using <code>\@gls@hyp@opt</code>	162	<code>\gls@wrglossary</code> : renamed from <code>\@wrglossary</code> to <code>\gls@wrglossary</code>	211
<code>\glsuserv</code> : switched to using <code>\@gls@hyp@opt</code>	162	<code>\glsaddprotectedpagefmt</code> : new	212
<code>\GLSuservi</code> : switched to using <code>\@gls@hyp@opt</code>	164	<code>\glsbackslash</code> : new	185
<code>\Glsuservi</code> : switched to using <code>\@gls@hyp@opt</code>	163	<code>sanitize</code> : none option	31
<code>\glsuservi</code> : switched to using <code>\@gls@hyp@opt</code>	163	4.12 – 2014-11-22 <code>\@gls@addpredefinedattributes</code> : Added <code>glsignore</code> attribute . .	63
<code>\ifignoredglossary</code> : new	81	<code>\@gls@adjustmode</code> : new	184
<code>\newglossary</code> : added starred version	78	<code>\@gls@notranslatorhook</code> : removed	31
<code>\newignoredglossary</code> : new . . .	80	<code>\@gls@toc</code> : added <code>\protect</code> to <code>\numberline</code>	59
<code>\ns@newglossary</code> : added <code>\@glotype@{<i>name</i>}@log</code>	79	<code>\@gls@usetranslator</code> : new . . .	31
new	78	<code>\glsacrpluralsuffix</code> : new . . .	50
<code>\p@gls@hyp@opt</code> : new	128	<code>\glsadd</code> : added check for vertical mode	184
<code>\PGLS</code> : changed to use <code>\@gls@hyp@opt</code>	423	<code>\glsaddallunused</code> : replaced <code>@gobble</code> with <code>glsignore</code> . . .	185
<code>\PglS</code> : changed to use <code>\@gls@hyp@opt</code>	421	<code>\glsifusedtranslatordict</code> : new	32
<code>\pglS</code> : changed to use <code>\@gls@hyp@opt</code>	420	<code>\glsignore</code> : new	185
<code>\PGLSpl</code> : changed to use <code>\@gls@hyp@opt</code>	423	<code>\glsupacrpluralsuffix</code> : new .	50
<code>\PglSpl</code> : changed to use <code>\@gls@hyp@opt</code>	422	<code>\ProvidesGlossariesLang</code> : new	50
<code>\pglSpl</code> : changed to use <code>\@gls@hyp@opt</code>	420	<code>\RequireGlossariesLang</code> : new .	50
<code>\s@gls@hyp@opt</code> : new	127	4.13 – 2015-02-03 <code>\indexspace</code> : new . . .	302, 325, 406
<code>\s@newglossary</code> : new	78	4.14 – 2015-02-28 <code>\@@glslocalreset</code> : new	113
4.09 – 2014-08-12 <code>\glsaddkey</code> : fixed bug in user commands	95	<code>\@@glslocalunset</code> : new	112
4.10 – 2014-08-27 <code>\@Gls@acrentryname</code> : new	174	<code>\@@glsreset</code> : new	113
<code>\@Gls@entryname</code> : new	174	<code>\@@glsunset</code> : new	113
		<code>\@@newglossaryentry@defcounters</code> : new	114
		<code>\@cGls</code> : new	118
		<code>\@cGls@</code> : new	118
		<code>\@cGlspl@</code> : new	119
		<code>\@cgls</code> : new	117
		<code>\@cgls@</code> : new	117
		<code>\@cglspl</code> : new	118, 119

\cglspl@: new	118	\@GLSpl: added	
\@gls@entry@count: new	117	\glspostlinkhook	148
\@gls@increment@currcount:		\@Gls@: added	
new	116	\glspostlinkhook	145
\@gls@local@increment@currcount:		\@GLspl@: added	
new	116	\glspostlinkhook	147
\@gls@write@entrycounts: new	117	\@acrlong: added	
\@glslocalreset: new	113	\glspostlinkhook	444
\@glslocalunset: new	112	\@acrshort: added	
\@glsreset: new	113	\glspostlinkhook	443
\@glsunset: new	112	\@gls@: added	
\@newglossaryentry@defcounters:		\glspostlinkhook	144
new	107	\@gls@@link: added	
\cGls: new	118	\glspostlinkhook	128
\cgl: new	117	\@gls@field@link: added	
\cGlsformat: new	118	\glspostlinkhook	149
\cglformat: new	117	\@gls@link: moved definition of	
\cGLspl: new	119	\glsifhyperon outside of this	
\cgl: new	118	macro	130
\cGLsplformat: new	119	\@glsdisp: added	
\cglformat: new	119	\glspostlinkhook	148
\gls@defdocnewglossaryentry:		\@GLspl@: added	
new	89	\glspostlinkhook	146
\glsenableentrycount: new	114	\glsacspace: new	275
\glslocalreset: switched to		\glsadd: changed	
\@glslocalreset	112	\@do@wrglossary to	
\glslocalunset: switched to		\@do@wrglossary	184
\@glslocalunset	112	\glsfielddef: new	98
\glsreset: switched to		\glsfieldedef: new	97
\@glsreset	111	\glsfieldfetch: new	98
\glsunset: switched to		\glsfieldgdef: new	98
\@glsunset	112	\glsfieldxdef: new	97
4.15 – 2015-03-16		\glsifhyperon: moved definition	
General: bug fix replaced		of \glsifhyperon	129
\@glo@type with \glstype	171	\glslinkpostsetkeys: new	129
4.16 – 2015-06-18		\glspostlinkhook: new	129
\glsaddstoragekey: new	94	\glswriteentry: new	211
4.16 – 2015-07-08		\ifglsfieldcseq: new	100
General: added		\ifglsfielddefeq: new	99
\glspostlinkhook	165–172	\ifglsfieldeq: new	99
\@ACRlong: added		long-sp-short: new	274
\glspostlinkhook	445	long-sp-short-desc: new	277
\@ACRshort: added		\showglofield: new	293
\glspostlinkhook	444		
\@Acrlong: added		4.18 – 2015-09-09	
\glspostlinkhook	445	General: split mfirstuc into	
\@Acrshort: added		separate bundle	4
\glspostlinkhook	444	4.19 – 2015-10-31	
\@GLS@: added		\glstreenamebox: new	412
\glspostlinkhook	145		

4.19 – 2015-11-22				
\@gls@link@nocheckfirsthyper:			\glsentrytitlecase: new 178	
new	149		\glsfindwidesttoplevelname:	
\@gls@preglossaryhook: new	222		new	411
\@printglossary: added			\glslistgroupheaderfmt: new	302
\@gls@preglossaryhook	224		\glslistnavigationitem: new	302
\do@glsdisablehyperinlist:			\glstreegroupheaderfmt: new	406
new	129		\glstreenavigationfmt: new	406
\doifglossarynoexistsordo:			\ifglswrallowprimitivemods:	
new	72		new	214
\gls@gobbleopt: new	78		list: fixed missing space before	
\glsdoifexistsordo: new	72		description	303
4.20 – 2015-11-30			long: fixed typo in	
\@gls@link: added			\glossentrydesc	306
\@gls@setdefault@glslink@opts			mcolalttreesspannav: new	330
.	130		mcolindexspannav: new	326
added \glsdonohyperlink			mcoltreenonamespannav: new	328
when hyperlink is suppressed	130		mcoltreesspannav: new	327
\@gls@setdefault@glslink@opts:			super4col: fixed bug in	
new	129		\glossentry	335
\gls@checkseeallowed@preambleonly:			4.23 – 2016-04-30	
new	84		altlongragged4col: check for	
\glsdonohyperlink: new	142		nogroupskip changed	321
4.21 – 2016-01-24			altsuperragged4col: check for	
General: changed checkfirsthyper			nogroupskip changed	342
assignment	164–172		\glscurrentfieldvalue: new	77
\@printglossary: warn if no			\ifglshasfield: added	
style has been set	223		\glscurrentfieldvalue	76
altlong4col-booktabs: new	314		long: check for nogroupskip	
altlongragged4col-booktabs:			changed	307
new	315		long-booktabs: check for	
\glsLTpenaltycheck: new	316		nogroupskip changed	313
\glspatchLToutput: new	316		long3col: check for nogroupskip	
\glspenaltygroupskip: new	316		changed	308
long-booktabs: new	312		long3col-booktabs: check for	
long3col-booktabs: new	313		nogroupskip changed	313
long4col-booktabs: new	313		long4col: check for nogroupskip	
longragged-booktabs: new	314		changed	310
longragged3col-booktabs: new	315		long4col-booktabs: check for	
\setglossarystyle: set default			nogroupskip changed	314
style if not already set	261		longragged: check for	
4.22 – 2016-04-19			nogroupskip changed	318
\@do@esc@wrglossary: added			longragged3col: check for	
check for \@arabic	215		nogroupskip changed	319
added test to allow temporary			super: check for nogroupskip	
primitive modifications and			changed	331
added arabic case	216		super3col: check for nogroupskip	
\gls@arabicpage: new	212		changed	333
\gls@protected@pagefmts:			super4col: check for nogroupskip	
added arabic to list	211		changed	335

superragged: check for nogroupskip changed	339	outside of theglossary environment	244
superragged3col: check for nogroupskip changed	341	\glsnavhyperlinkname: new	296
4.24 – 2016-05-27		4.30 – 2017-06-11	
\@gls@extramakeindexopts: new	196	\@glo@autosee: new	106
\@gls@glossary: added check for debug mode	210	\@glo@autoseehook: new	107
\@gls@see@noindex: new	9	\@glo@check@sortallowed: new	16
debug: new	7	\@gls@noidx@do: letter group assignment made global	246
\glsnomakeindexwarning: new	60	\@gls@setupsort@def: added check for register	18
\GlsSetQuote: new	193	\@gls@setupsort@none: new	19
\GlsSetWriteIstHook: new	193	\@xdycrossrefhook: new	65
seenindex: new	9	\@xdylocationclassorder: bug fix: changed \edef to \def	66
4.25 – 2016-06-09		\glosortentrieswarning: new	24
General: changed		\gls@set@xr@key: new	83
\DeclareRobustCommand to \newrobustcmd and changed		\gls@xr@key: new	84
\ifundefined to \ifcsundef	437	\GlsAddXdyLocation: bug fix: changed #1 to #2	65
\@gls@enablesavenonumberlist: new	85	\glsnoidxstripaccents: added \a	28
\@gls@initnonumberlist: new	85	added \TH, \dh and \DH	29
\@gls@savenonumberlist: new	85	4.31 – 2017-08-10	
4.26 – 2016-10-12		nolist: added check for “list” style	13
\@glossary@default@style: added check for classicthesis	11	4.31 – 2017-09-10	
\glstreechildpredesc: new	407	style: changed \renewcommand to \def	11
\glstreeitem: new	406	4.32 – 2017-08-24	
\glstreepredesc: new	407	\@glsnavhypertarget: new	297
\glstreesubitem: new	406	\@glsshowtarget: new	8
\glstreesubsubitem: new	406	\glsshowtarget: new	8
mcolindex: replaced \@idxitem with \glstreeitem	325	4.33 – 2017-09-20	
mcolindexspannav: replaced \@idxitem with \glstreeitem	326	\@do@esc@wrglossary: added \gls@the and \gls@number	216
4.28 – 2017-01-07		renamed from \@do@esc@wrglossary	214
\glspatchtabularx: new	111	\@do@noesc@wrglossary: new	213
4.29 – 2017-01-19		\@do@wrglossary: changed to check for esclocations	213
General: added check for \@glsxtr@doaccsupp	424	\@gls@missinglang@warn: new	24
\@gls@noidx@do: current letter group assignment made global	246	esclocations: new	13
\@gls@old@print@noidx@glossary: moved definition of \@gls@currentlettergroup		\GlsSetXdyFirstLetterAfterDigits: added starred version	186
		\GlsSetXdyNumberGroupOrder: new	186

4.34 – 2017-11-03		
<code>\glsshowtarget</code> : modified to check for math mode and inner	8	
<code>mcolalttreesspannav</code> : removed spurious space	330	
4.35 – 2017-11-14		
<code>\glsadd</code> : added <code>\@gls@setsort</code> (in case of <code>sort=use</code>)	184	
4.36 – 2018-03-07		
<code>\@gls@glossary</code> : removed <code>\index</code>	210	
4.37 – 2018-04-07		
<code>\gls@begindocdefs</code> : new	91	
4.38 – 2018-05-10		
<code>\@gls@define@glossaryentrycounter</code> : added check for existence of glossaryentry counter	15	
new	15	
<code>\@gls@define@glossarysubentrycounter</code> : new	15	
prepended <code>\currentglossary.</code> to <code>\theHglossarysubentry</code> and removed spurious eol space	16	
<code>\glsaccsupp</code> : added braces around actual text argument	434	
<code>\glseentrycounterlabel</code> : bug fix: move conditional inside command	254	
<code>\GlsEntryCounterLabelPrefix</code> : new	251	
<code>\glseentryitem</code> : bug fix: move conditional inside command	254	
<code>\glsrefentry</code> : bug fix: move conditional inside command	254	
<code>\glsresetentrycounter</code> : bug fix: move conditional inside command	253	
<code>\glsresetsubentrycounter</code> : bug fix: move conditional inside command	253	
<code>\glsstepentry</code> : bug fix: move conditional inside command	253	
<code>\glsstepsubentry</code> : bug fix: move conditional inside command	253	
<code>\glssubentrycounterlabel</code> : bug fix: move conditional inside command	254	
		<code>\glssubentryitem</code> : bug fix: move conditional inside command
		254
		<code>\showglnonameaccess</code> : bug fix: corrected field (was showing text access field)
		462
4.40 – 2018-06-01		
<code>\istfile</code> : changed <code>\def</code> to <code>\providecommand</code>	209	
<code>\makenoidxglossaries</code> : false . .	203	
4.41 – 2018-07-23		
General: changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>	250	
<code>\@gls@override@glossary</code> : new	41	
debug: changed <code>\val</code> and <code>\nr</code> to <code>\gls@debug@val</code> and <code>\gls@debug@nr</code>	7	
<code>\glossary</code> : added warning	42	
<code>\gls@original@glossary</code> : new . .	41	
<code>\gls@original@makeglossary</code> : new	41	
<code>kernelglossredefs</code> : new	42	
<code>\makeglossaries</code> : removed redefinition of <code>\makeglossary</code>	201	
<code>\makeglossary</code> : added warning	41	
<code>nonumberlist</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@nonumberlist@val</code> and <code>\gls@nonumberlist@nr</code>	84	
<code>numberedsection</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>	11	
<code>seenoindex</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@seenoindex@val</code> and <code>\gls@seenoindex@nr</code>	9	
<code>translate</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@translate@val</code> and <code>\gls@translate@nr</code>	32	
4.42 – 2019-01-06		
<code>\@gls@@automake@immediate</code> : new	200	
<code>\@gls@automake@immediate</code> : new	198	
<code>altlong4col-booktabs</code> : removed superfluous <code>\glspatchLToutput</code>	314	

\@glsplural@: added accessibility support	446	\Glsentryprefixfirstplural: added \glsdetoklabel	418
\@glsshowaccsupp: new	9	\glsentryprefixfirstplural: added \glsdetoklabel	418
\@glssymbol@: added accessibility support	448	\Glsentryprefixplural: added \glsdetoklabel	419
\@glssymbolplural@: added accessibility support	448	\glsentryprefixplural: added \glsdetoklabel	418
\@glstext@: added accessibility support	445	\glsentrytitlecase: added existence check	178
\@glsuseri@: added accessibility support	448	\glsentryuseriaccess: new	432
\@glsuserii@: added accessibility support	449	\glsentryuseriiaccess: new	432
\@glsuseriii@: added accessibility support	449	\glsentryuseriiiaccess: new	432
\@glsuseriv@: added accessibility support	449	\glsentryuserivaccess: new	432
\@glsuserv@: added accessibility support	450	\glsentryuserviaccess: new	432
\@glsuservi@: added accessibility support	450	\glsfieldaccsupp: new	433
\@org@Gls@acrentryname: added check for \glsshortaccessdisplay	175	\glsfirstaccessdisplay: added check for existence	435
acronyms: changed \renewcommand to \def	22	\glsfirstpluralaccessdisplay: added check for existence	435
debug: showaccsupp	7	\glslongaccessdisplay: added check for existence	436
disablemakegloss: new	36	\glslongpluralaccessdisplay: added check for existence	436
\gls@accessibility: new	425	\glsnameaccessdisplay: added check for existence	434
\gls@accsupp@engine: new	425	\glspluralaccessdisplay: added check for existence	435
\glsaccessibility: new	425	\glsprefixsep: new	420
\glsdefaultshortaccess: new	461	\glssee: switched to \newrobustcmd	219
\glsdescriptionaccessdisplay: added check for existence	435	\glsseeformat: switched to \newrobustcmd	219
\glsdescriptionpluralaccessdisplay: added check for existence	435	\glsseeitem: switched to \newrobustcmd	220
\glsentrydescpluralaccess: corrected field reference	431	\glsseelist: switched to \newrobustcmd	219
\glsentryfirstpluralaccess: switched to using \@gls@entry@field	431	\glsshortaccessdisplay: added check for existence	436
\glsentryparent: new	178	\glsshortaccsupp: new	433
\Glsentryprefix: added \glsdetoklabel	419	\glsshortplaccsupp: new	433
\glsentryprefix: added \glsdetoklabel	418	\glsshortpluralaccessdisplay: added check for existence	436
\Glsentryprefixfirst: added \glsdetoklabel	418	\glsshowaccsupp: new	9
\glsentryprefixfirst: added \glsdetoklabel	418	\glsshowtargetfont: new	8
		\glsshowtargetouter: new	8
		\glsshowtargetsymbol: new	8
		\glsymbolaccessdisplay: added check for existence	435

<code>\glssymbolpluralaccessdisplay:</code>		<code>\ifglossaryexists:</code> added	
added check for existence	435	starred form	70
<code>\glstextaccessdisplay:</code> added		<code>\s@ifglossaryexists:</code> new	70
check for existence	434	<code>\setglossarypreamble:</code> switched	
<code>\glsuseriaccessdisplay:</code> new	436	to starred form of	
<code>\glsuseriiaccessdisplay:</code> new	436	<code>\ifglossaryexists</code>	56
<code>\glsuseriiiaccessdisplay:</code>		4.47 – 2021-09-20	
new	436	General: changed <code>\x</code> to <code>\@gls@x</code>	263
<code>\glsuserivaccessdisplay:</code> new	437	<code>\@do@esc@wrglossary:</code> changed	
<code>\glsuservaccessdisplay:</code> new	437	<code>\edef</code> to <code>\protected@edef</code>	217
<code>\glsuserviiaccessdisplay:</code> new	437	<code>\@do@noesc@wrglossary:</code>	
<code>\ifglshaschildren:</code> made robust	73	changed <code>\edef</code> to	
<code>\ifglshasfield:</code> made robust	75	<code>\protected@edef</code>	214
<code>\ifglshaslong:</code> made robust	74	<code>\@do@write@glslabels:</code> fully	
<code>\ifglshasprefix:</code> added		expand list	37
<code>\glsdetoklabel</code>	419	<code>\@gls@@default@acr@descplural:</code>	
<code>\ifglshasprefixfirst:</code> added		new	285
<code>\glsdetoklabel</code>	419	<code>\@gls@@default@acr@firstplural:</code>	
<code>\ifglshasprefixfirstplural:</code>		new	285
added <code>\glsdetoklabel</code>	419	<code>\@gls@@default@acr@plural:</code>	
<code>\ifglshasprefixplural:</code> added		new	285
<code>\glsdetoklabel</code>	419	<code>\@gls@noexpand@assign@default@plurals:</code>	
<code>\ifglshasshort:</code> made robust	75	new	285
<code>\ifglshassymbol:</code> made robust	74	<code>\glo@do@sortentries:</code> changed	
<code>\makeglossaries:</code> let		<code>\edef</code> to <code>\protected@edef</code>	235
<code>\@makeglossary</code> to <code>\@gobble</code>		<code>\glo@sortentries:</code> changed	
instead of <code>\relax</code>	201	<code>\edef</code> to <code>\protected@edef</code>	234
<code>restoremakegloss:</code> new	37	<code>\glo@storeentry:</code> changed	
<code>user1access:</code> new	427	<code>\edef</code> to <code>\protected@edef</code>	108
<code>user2access:</code> new	427	<code>\@gls@checkesc:</code> new	139
<code>user3access:</code> new	427	<code>\@gls@deprecated@acr@warning:</code>	
<code>user4access:</code> new	427	new	288
<code>user5access:</code> new	428	<code>\@gls@link:</code> changed <code>\edef</code> to	
<code>user6access:</code> new	428	<code>\protected@edef</code>	130
<code>writeglslabels:</code> new	38	<code>\@gls@setupsort@use:</code> changed	
<code>\xglfieldaccsupp:</code> new	433	<code>\edef</code> to <code>\protected@edef</code>	18
4.46 – 2020-03-19		<code>\@gls@writedef:</code> added a	
<code>\@ifglossaryexists:</code> new	70	warning if entry has already	
<code>\@printgloss@checkexists:</code>		been defined	92
new	221	<code>\@makeglossaries@warn@noprintglossary:</code>	
<code>\@printgloss@checkexists@allowignored:</code>		new	202
new	221	<code>\@write@glslabelsname:</code> new	38
<code>\@printgloss@checkexists@noignored:</code>		<code>\do@write@glslabelsname:</code> new	38
new	221	<code>\forglsentries:</code> changed <code>\edef</code>	
<code>\@printglossary:</code> replaced		to <code>\protected@edef</code>	69
<code>\ifglossaryexists</code> with		<code>\gls@defdocnewglossaryentry:</code>	
<code>\@printgloss@checkexists</code>	223	removed see warning	89
<code>\doifglossarynoexistsordo:</code>		<code>\gls@defglossaryentry:</code> changed	
switched to starred form of		<code>\edef</code> to	
<code>\ifglossaryexists</code>	72	<code>\protected@edef</code>	101, 102

<code>\gls@docdef@addlabel</code> : new ..	90	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\gls@docdef@labellist</code> : new ..	90	added deprecated warnings	286
<code>\gls@glsdefs@addlabel</code> : new ..	90	<code>\SetDUASStyle</code> : added deprecated	
<code>\gls@glsdefs@labellist</code> : new ..	90	warnings	287
<code>\gls@save@numberlist</code> : changed		<code>\SetFootnoteAcronymStyle</code> :	
<code>\edef</code> to <code>\protected@edef</code>	221	added deprecated warnings	287
<code>\glsadd</code> : changed <code>\edef</code> to		<code>\SetSmallAcronymStyle</code> : added	
<code>\protected@edef</code>	184	deprecated warnings	287
<code>\glsaddall</code> : changed <code>\edef</code> to		<code>writeglslabelnames</code> : new	38
<code>\protected@edef</code>	184	4.48 – 2021-10-19	
<code>\glsdefs@newdocentry</code> : new ..	91	<code>\glscapitalisewords</code> : new	178
<code>\glsdisplaynumberlist</code> : changed		<code>\glslistexpandedname</code> : new	302
<code>\edef</code> to <code>\protected@edef</code>	182	<code>\glslistinit</code> : new	302
<code>\glsfielddef</code> : changed <code>\edef</code> to		<code>\glsunexpandedfieldvalue</code> : new	99
<code>\protected@edef</code>	98	4.49 – 2021-11-01	
<code>\glsfielddedef</code> : changed <code>\edef</code> to		<code>\glslistinit</code> : added	
<code>\protected@edef</code>	97	<code>\glsnavigation</code>	302
<code>\glsfieldfetch</code> : changed <code>\edef</code>		4.50 – 2022-10-14	
to <code>\protected@edef</code>	98	General: added initialisation	266
<code>\glsfieldgdef</code> : changed <code>\edef</code> to		allow markup to be stripped	
<code>\protected@edef</code>	98	from location	264
<code>\glsfieldxdef</code> : changed <code>\edef</code> to		corrected description mapping	428
<code>\protected@edef</code>	97	don't load <code>glossary-super</code> if <code>xtab</code>	
<code>\glsIfListOfAcronyms</code> : changed		has been loaded	13
<code>\edef</code> to <code>\protected@edef</code>	22	No longer requires <code>textcase</code>	
<code>\glsmoveentry</code> : changed <code>\edef</code> to		with <code>mfirstuc 2.08+</code>	5
<code>\protected@edef</code>	107	removed automatic loading of	
<code>\glsnavhyperlink</code> : changed		<code>glossaries-compatible-307</code>	44
<code>\edef</code> to <code>\protected@edef</code>	295	switched to using	
<code>\GlsSetQuote</code> : added check for		<code>\glsuppercase</code>	
backslash	195 150–164, 166, 168, 170, 172	
<code>\glsstepsubentry</code> : changed		<code>\@do@esc@wrglossary</code> : moved	
<code>\edef</code> to <code>\protected@edef</code>	254	<code>\romannumeral</code> inside allow	
<code>\ifglsfieldcseq</code> : changed <code>\edef</code>		primitive mods	216
to <code>\protected@edef</code>	100	<code>\@gls@@default@acr@descplural</code> :	
<code>\ifglshaschildren</code> : changed		removed	285
<code>\edef</code> to <code>\protected@edef</code>	73	<code>\@gls@@default@acr@firstplural</code> :	
<code>\ifignoredglossary</code> : changed		removed	285
<code>\edef</code> to <code>\protected@edef</code>	81	<code>\@gls@@default@acr@plural</code> :	
<code>\makenoidxglossaries</code> : changed		removed	285
<code>\edef</code> to <code>\protected@edef</code>	203	<code>\@gls@default@entryfmt</code> :	
<code>\new@glossaryentry</code> : allow		removed	437
overwrite if defined in		<code>\@gls@navhypertarget</code> : new	297
<code>.glsdefs</code>	90	<code>\@gls@noexpand@assign@default@plurals</code> :	
<code>\newignoredglossary</code> : changed		removed	285
<code>\edef</code> to <code>\protected@edef</code>	80	<code>\@Gls@acronymname</code> : removed	
<code>\ns@newglossary</code> : changed <code>\edef</code>		<code>mfirstuc</code> internal commands	174
to <code>\protected@edef</code>	79	<code>\@Glsdisp</code> : new	149
<code>\SetDescriptionDUAAcronymStyle</code> :		<code>\@Glsentryfield</code> : new	173
added deprecated warning	286		

\@domakeglossaries: added		\CustomNewAcronymDef:	
check for \nofiles	36	removed	288
\@gls@automake: added check for		\DefaultNewAcronymDef:	
void codepage	198	removed	285, 461
\@gls@automake@immediate:		\defglsdisplay: removed	125
added check for void		\defglsdisplayfirst: removed	125
codepage	199	\DescriptionDUANewAcronymDef:	
\@gls@automake@types: new	200	removed	286
\@gls@defaulttype: new	77	\DescriptionFootnoteNewAcronymDef:	
\@gls@do@automake@err: new	39	removed	286, 461
\@gls@do@glsunset: new	127	\DescriptionNewAcronymDef:	
\@gls@entry@field: switched to		removed	286, 461
\csuse	172	dua: switched to using	
\@gls@escbsdq: check		\glsuppercase	279, 280
esclocations setting	133	dua-desc: removed \acronymfont	
\@gls@getcounterprefix:		from around long form	281
changed \edef to \def	218	\DUANewAcronymDef: removed	287
\@gls@getgrouptitle: switched		esclocations: default now false	13
to L ^A T _E X3 syntax	259	\findrootlanguage: removed	67
\@gls@glslink@hyper@update@hook:		footnote: switched to using	
new	126	\glsuppercase	281
\@gls@hypergroup: updated to		\FootnoteNewAcronymDef:	
use etoolbox	298	removed	287, 461
\@gls@noist@toolate: new	196	\glossary: removed	42
\@gls@removedoption: new	29	\glossaryentryfield: removed	258
\@gls@requested@languages:		\glossarymark: switched to using	
new	33	\glsuppercase	57
\@gls@restore@glslocal: new	127	\glossarystyle: removed	261
\@gls@sanitizedesc: removed	25	\glossarysubentryfield:	
\@gls@save@glslocal: new	126	removed	258
\@gls@setupsort@clear: new	20	\gls@automake@makegloss: new	38
\@gls@defaultplural: Removed	87	\gls@automake@option: new	38
\@org@Gls@acentryname: saved		\gls@codepage: set default	
original definition of		codepage to utf8 if	
\@Gls@acentryname	174	\inputencodingname not	
\@xp@gls@getcounterprefix:		defined and switched to	
new	217	\ifdefvoid	36
\@xp@gls@hypernumber: new	262	\gls@delimN: renamed \@delimN	
automake: added makegloss and		to \gls@delimN	263
lite options	39	\gls@delimR: renamed \@delimR	
automakegloss: new	40	to \gls@delimR	263
automakeglosslite: new	40	\gls@disablepagerefexpansion:	
compatible-2.07: deprecated	41	added check for esclocations	212
compatible-3.07: deprecated	40	\gls@noidx@nonumberlist: new	84
\compatibleglossentry:		\gls@noidx@glossary: added	
removed	255, 425	\gls@wrglossdisablelocationcmds	
\compatiblesubglossentry:			207
removed	258, 425	\gls@xdy@set@lang: new	35
\CustomAcronymFields:		\gls@addkey: allow expansion in	
removed	288	PDF bookmarks	95

switched to using		
\glssupercase	96	
\GlsAddXdyAttribute: remove duplicate counterless attribute definitions	61, 62	
\Glsdisp: new	149	
\glsdisplay: removed	125	
\glsdisplayfirst: removed	125	
\glsdoparenifnotempty: replaced		
\DeclareRobustCommand with		
\newrobustcmd	287	
\glsdoshowtarget: new	8	
\glsencapwrcntent: new	131	
\Glsentrydesc: allow expansion in PDF bookmarks	175	
\Glsentrydescplural: allow expansion in PDF bookmarks	176	
\Glsentryfirst: allow expansion in PDF bookmarks	177	
\Glsentryfirstplural: allow expansion in PDF bookmarks	178	
\glsentryfmt: obsolete commands replaced with		
\glsgenentryfmt	120	
\Glsentrylong: allow expansion in PDF bookmarks	181	
\Glsentrylongpl: allow expansion in PDF bookmarks	181	
\Glsentryname: allow expansion in PDF bookmarks	174	
\Glsentryplural: allow expansion in PDF bookmarks	176	
\Glsentryprefix: allow expansion in PDF bookmarks	419	
\Glsentryprefixfirst: allow expansion in PDF bookmarks	418	
\Glsentryprefixfirstplural: allow expansion in PDF bookmarks	418	
\Glsentryprefixplural: allow expansion in PDF bookmarks	419	
\Glsentryshort: allow expansion in PDF bookmarks	180	
\Glsentryshortpl: allow expansion in PDF bookmarks	181	
\Glsentrysymbol: allow expansion in PDF bookmarks	177	
\Glsentrysymbolplural: allow expansion in PDF bookmarks	177	
\Glsentrytext: allow expansion in PDF bookmarks	176	
\glsentrytitlecase: use sentence case for PDF bookmark	178	
\Glsentryuseri: allow expansion in PDF bookmarks	179	
\Glsentryuserii: allow expansion in PDF bookmarks	179	
\Glsentryuseriii: allow expansion in PDF bookmarks	179	
\Glsentryuseriv: allow expansion in PDF bookmarks	180	
\Glsentryuserv: allow expansion in PDF bookmarks	180	
\Glsentryuservi: allow expansion in PDF bookmarks	180	
\glsfielddef: made robust	98	
\glsfieldedef: made robust	97	
\glsfieldfetch: made robust	98	
\glsfieldgdef: made robust	98	
\glsfieldxdef: made robust	97	
\glsgenacfmt: switched to using		
\glssupercase	123	
\glsgenentryfmt: switched to using		
\glssupercase	121	
\glsgetgrouplabel: converted to use L ^A T _E X3	260	
\glsgetgrouptitle: made robust	259	
\gls glossarymark: switched to		
\glssupercase	57	
\glsifhyper: removed	127	
\glsindexingsetting: new	16	
\glsinlineifhaschildren: new	300	

<code>\Glslink</code> : new	128	<code>\SetCustomDisplayStyle</code> :	
<code>\glslocationstoencap</code> : new	55	removed	288
<code>\glslowercase</code> : new	4	<code>\SetCustomStyle</code> : removed	288
<code>\glsmfuaddmap</code> : new	5	<code>\SetDefaultAcronymStyle</code> :	
<code>\glsmfublocker</code> : new	5	removed	285
<code>\glsmfuexcl</code> : new	5	<code>\SetDescriptionAcronymDisplayStyle</code> :	
<code>\glsnavhyperlink</code> : changed		removed	286
<code>\@glo@type</code> to		<code>\SetDescriptionAcronymStyle</code> :	
<code>\currentglossary</code>	295	removed	286
<code>\glsnavhypertarget</code> : changed		<code>\SetDescriptionDUAAcronymDisplayStyle</code> :	
<code>\@glo@type</code> to		removed	286
<code>\currentglossary</code>	297	<code>\SetDescriptionDUAAcronymStyle</code> :	
<code>\glsnavigation</code> : changed		removed	286
<code>\@glo@type</code> to		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\currentglossary</code>	298	removed	286
<code>\glsnoidxprenumberlist</code> : new	85	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\glsnumberlistloop</code> : corrected		removed	286
parameter number	206	<code>\SetDUADisplayStyle</code> : removed	287
<code>\glsseeitemformat</code> : added test		<code>\SetDUASStyle</code> : removed	287
for short field	220	<code>\SetFootnoteAcronymDisplayStyle</code> :	
<code>\glsentencecase</code> : new	4	removed	287
<code>\glsshowtargetfonttext</code> : new	8	<code>\SetFootnoteAcronymStyle</code> :	
<code>\glsshowtargetinner</code> : new	8	removed	287
<code>\glstexpdfstring</code> : new	173	<code>\SetGenericNewAcronym</code> :	
<code>\glsuppercase</code> : new	5	corrected long plural suffix	271
<code>\GlsWarnAddProtectedPageFmt</code> :		switched to using	
new	213	<code>\glsuppercase</code>	271
<code>\glswrglossdisableanchorcmds</code> :		<code>\setglossentrycompatibility</code> :	
new	213	removed	258
<code>\glswrglossdisablelocationcmds</code> :		<code>\SetSmallAcronymDisplayStyle</code> :	
new	212	removed	287
<code>\glswrglosslocationtarget</code> :		<code>\SetSmallAcronymStyle</code> :	
new	213	removed	287
<code>\glswrglosslocationtextfmt</code> :		<code>\SmallNewAcronymDef</code> :	
new	213	removed	287, 461
<code>\ifglsfieldcseq</code> : made robust	100	<code>sort</code> : added clear	16
<code>\ifglsfielddefeq</code> : made robust	99	4.51 – 2022-11-02	
<code>\ifglsfieldeq</code> : made robust	99	General: switch to	
<code>\ifglsfieldvoid</code> : new	75	<code>\TrackIfKnownLanguage</code>	51
<code>\ifglsresetcurrcount</code> : new	114	<code>\do@wrglossary</code> : added	
<code>\ifglsrallowprimitivemods</code> :		<code>\gls@ifnotmeasuring</code>	211
switch default to false	214	<code>\@gls@@automake@immediate</code> :	
<code>\istfile</code> : removed	209	ensure list is expanded before	
<code>kernelglossredefs</code> : corrected		iterating	200
names of choice variables	42	<code>\@gls@if@measuring</code> : new	109
<code>languages</code> : new	33	<code>alttree</code> : replaced <code>\settowidth</code>	
<code>\makeglossary</code> : removed	41	with <code>\glsmeasurewidth</code>	413, 414
<code>mfirstuc</code> : new	44	<code>\gls@if@measuring</code> : new	109
<code>\SetAcronymStyle</code> : removed	288	<code>\gls@local@stepcounter</code> : new	110
		<code>\gls@org@refstepcounter</code> : new	109

<code>\gls@start@measuring</code> : new ..	110	<code>\Glsentryplural</code> : change map to exclusion	176
<code>\gls@stop@measuring</code> : new ...	111	<code>\Glsentryshort</code> : change map to exclusion	180
<code>\glsdohypertarget</code> : added measuring and group	140	<code>\Glsentryshortpl</code> : change map to exclusion	181
<code>\glsfindwidesttoplevelname</code> : replaced <code>\settowidth</code> with <code>\glsmeasurewidth</code>	411	<code>\Glsentrysymbol</code> : change map to exclusion	177
<code>\glsifmeasuring</code> : new	110	<code>\Glsentrysymbolplural</code> : change map to exclusion	177
<code>\glsmeasuredepth</code> : new	111	<code>\Glsentrytext</code> : change map to exclusion	176
<code>\glsmeasureheight</code> : new	111	<code>\Glsentryuseri</code> : change map to exclusion	179
<code>\glsmeasurewidth</code> : new	111	<code>\Glsentryuserii</code> : change map to exclusion	179
<code>\glspatchtabularx</code> : switch measuring on and off	111	<code>\Glsentryuseriii</code> : change map to exclusion	179
<code>\makeglossaries</code> : ensure list of glossaries is expanded before iterating	202	<code>\Glsentryuseriv</code> : change map to exclusion	180
4.52 – 2022-11-03		<code>\Glsentryuserv</code> : change map to exclusion	180
<code>\gls@local@stepcounter</code> : corrected spelling of <code>\@ne</code> ..	110	<code>\Glsentryuservi</code> : change map to exclusion	180
4.53 – 2023-09-29		<code>\glslabelhypertarget</code> : new ..	141
<code>\@gls@hypergroup</code> : rewritten in <code>ℒ_{TeX}3</code>	298	<code>\glslabelhypertargetdefs</code> : new	141
<code>\glsnavhypergroupdotarget</code> : new	297	<code>\glslabelhypertargetprefix</code> : new	141
<code>\glsnavigationitem</code> : new	298	<code>\glslabelhypertargetvalue</code> : new	141
4.54 – 2024-04-03		<code>\glslowercase</code> : ensure argument is expanded	4
General: added check for <code>\l_datatool_measure_hook_tl</code>	110	<code>\glsuppercase</code> : ensure argument is expanded	5
<code>\gls@printglossary@begin</code> : new	224	4.55 – 2024-11-01	
<code>\gls@printglossary@end</code> : new ..	224	<code>\@gls@run@output@dir</code> : new ..	45
<code>\glsdohyperlinkhook</code> : new ...	142	<code>\@gls@run@output@file</code> : new ..	45
<code>\glsdohypertargethook</code> : new ..	141	<code>\@gls@run@unrestricted@shell</code> : new	44
<code>\Glsentrydesc</code> : change map to exclusion	175	<code>\@gls@runshell</code> : new	44
<code>\Glsentrydescplural</code> : change map to exclusion	176	<code>\gls@set@xr@key</code> : added check for empty value	83
<code>\Glsentryfirst</code> : change map to exclusion	177	locales: new	33
<code>\Glsentryfirstplural</code> : change map to exclusion	178	4.56 – 2025-03-03	
<code>\Glsentryfull</code> : change map to exclusion	181	General: store original sort as well as label in actual part	226
<code>\Glsentrylong</code> : change map to exclusion	181	4.57 – 2025-03-12	
<code>\Glsentrylongpl</code> : change map to exclusion	181	<code>\@@gls@noidx@nosanitizesort</code> : check for datatool preprocess	
<code>\Glsentryname</code> : change map to exclusion	174		

function	27	\gls@defglossaryentry: moved	
\@print@noidx@glossary:		auto see after index field has	
switched to constructing		been set	106
token list	239	\GlsMcolTreeSetup: new	323
\glsnoidxprecontenthook: new	241	\glsnoidxinithook: new	241
4.58 – 2025-03-19		\glsnoidxitemhook: new	241
\@gls@noidx@nosanitizesort:		\GlsNoIdxMissingAction: new	241
defer existence check	27	\GlsTreeSetup: new	344
\@gls@setupsort@use: internal		\GlsTreeStarBox: new	345
count register changed to L3		\GlsTreeStarItemCounterBox:	
int var	18	new	346
switched to L ^A T _E X3	18	\GlsTreeStarNameBox: new	346
mfirstuc: redefine		\GlsTreeStarOuterBox: new	346
\glsmakefirstuc as a long		\GlsTreeStarSubItemCounterBox:	
command	44	new	346
4.59 – 2025-04-11		\GlsTreeStarSymbolBox: new	346
\@glo@assign@sortkey:		\GlsTreeUpdateWidestNameAndSymbol:	
initialise	200	new	345
\@glo@sortmacro@lettercase:		\GlsTreeUpdateWidestNameOrSymbol:	
new	237	new	345
\@glo@sortmacro@wordcase:		mcoltree*: new	323
new	237	style-options: new	11
\@gls@noidx@addtorefs: new	207	tree*: new	344
\GLOSSentrydesc: new	256	4.6 – 2025-04-14	
\GlossEntryDesc: new	257	mcoltree*: corrected tree header	
\GLOSSentryname: new	256	function name	324
\GlossEntryName: new	256	tree*: corrected parameter spec	
\GLOSSentrysymbol: new	257	for tree header function	405
\GlossEntrySymbol: new	257	corrected sub-group title	346

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	
\!	134, 137, 195, 538, 539, 787, 789, 790, 820, 1041, 1043, 1074, 1196, 1198, 1230, 1353, 1355, 1387
\"	28, 134– 137, 139, 475, 536–538, 540, 725, 787–790, 792, 979, 1040– 1043, 1045, 1046, 1134, 1195– 1198, 1200, 1289, 1352–1355, 1357
\#	139, 189, 563, 564, 791, 792, 815, 1045, 1070, 1200, 1225, 1226, 1357, 1382, 1383
\%	186, 192, 561, 565, 666–668, 813, 817, 919–921, 1067, 1072, 1223, 1227, 1380, 1384
\&	50, 183, 486, 560, 736, 812, 992, 1066, 1146, 1222, 1303, 1379
\'	28, 475, 725, 979, 1134, 1289
\.	14, 28, 468, 475, 718, 725, 972, 979, 1127, 1134, 1281, 1289
\=	28, 475, 725, 979, 1134, 1289
\?	134–136, 194, 536–538, 567, 787– 789, 819, 1040, 1042, 1074, 1195, 1197, 1229, 1352, 1354, 1386
\@	91, 508, 759, 1015, 1169, 1326
\@@delimN	598, 850
\@do@wrglossary	203, 214, 217, <u>217</u> , 573, 579, 580, 825, 830, 831, 1080, 1086, 1087, 1236, 1242, 1243, 1393, 1399
\@do@esc@wrglossary	213, <u>214</u> , 578, 579, 830, 831, 1086, 1241, 1242, 1398, 1399
\@do@noesc@wrglossary	213, <u>213</u> , 578, 830, 1086, 1241, 1398
\@do@wrglossary	184, 211, <u>213</u> , 560, 577, 578, 812, 829, 830, 1067, 1084, 1086, 1222, 1240, 1241, 1379, 1397, 1398
\@do@write@glslabels	37, <u>37</u> , 481, 731, 986, 1140, 1296
\@glo@sortentries	233, 237, 238, 1409, 1411, 1412
\@glo@assign@sortkey	204, <u>252</u> , 574, 592, 826, 844, 1081, 1100, 1237, 1255, 1394, 1421
\@glo@list	69, 497, 747, 1003, 1158, 1315
\@glo@sort	28, 475, 725, 979, 1134, 1289
\@glo@type	222, 582, 834, 1090, 1245, 1246, 1402, 1403
\@glossarysec	10, <u>10</u> , 58, 59, 398, 467, 490, 716, 717, 740, 971, 996, 1125, 1150, 1151, 1280, 1307, 1308
\@glossaryseclabel	10, 11, 58, 251, 467, 490, 592, 717, 740, 843, 844, 971, 996, 1099, 1125, 1126, 1150, 1151, 1255, 1280, 1307, 1308, 1420
\@glossarysecstar	10, 11, 58, 251, 467, 490, 592, 717, 740, 843, 844, 971, 996, 1099, 1125, 1126, 1150, 1255, 1280, 1307, 1420
\@gls@default@acr@desclplural	285, 864
\@gls@default@acr@firstplural	285, 864
\@gls@default@acr@plural	285, 864
\@gls@checkactual	138, 540, 791, 1044, 1045, 1199, 1200, 1356, 1357
\@gls@checkbar	137, 539, 790, 1044, 1198, 1199, 1355, 1356
\@gls@checkescactual	136, 537, 538, 788, 789, 1042, 1197, 1354
\@gls@checkescbar	136, 538, 789, 1043, 1197, 1198, 1354, 1355
\@gls@checkesclevel	137, 538, 539, 789, 790, 1043, 1198, 1355
\@gls@checkescquote	135, 195, 196, 537, 568, 569, 788, 820, 821, 1042, 1075, 1196, 1197, 1230, 1231, 1353, 1354, 1387, 1388

<code>\@@gls@checklevel</code>	138, 539, 540, 790, 791, 1044, 1199, 1356	<code>\@@gls@localreset</code>	113, 113, 115, 521, 523, 772, 773, 1029, 1030, 1184, 1185, 1341, 1342
<code>\@@gls@checkquote</code>	135, 193, 194, 536, 537, 567, 787, 788, 818, 819, 1041, 1073, 1196, 1229, 1353, 1386	<code>\@@gls@localunset</code>	112, 112, 115, 521, 522, 772, 773, 1029, 1030, 1184, 1185, 1341, 1342
<code>\@@gls@codepage</code>	198, 199, 1076–1078, 1232, 1233, 1389, 1390	<code>\@@gls@reset</code>	113, 113, 115, 521, 522, 772, 773, 1029, 1030, 1184, 1185, 1341, 1342
<code>\@@gls@default@entryfmt</code> . .	437, 525, 531, 532, 636, 776, 782, 889	<code>\@@gls@showtarget</code>	7, 9, 9, 465, 466, 715, 716, 969, 970, 1124, 1278, 1279
<code>\@@gls@expand@field</code> . .	25, 88, 88, 94, 95, 474, 506, 507, 510, 612–615, 617, 619, 621, 623, 657–661, 724, 757, 760, 761, 864, 866, 867, 869, 871, 872, 874, 876, 911–914, 978, 1013, 1016, 1017, 1133, 1167, 1168, 1171, 1172, 1287, 1324, 1325, 1328, 1329	<code>\@@gls@unset</code>	112, 113, 115, 521, 522, 772, 773, 1029, 1030, 1184, 1185, 1341, 1342
<code>\@@gls@extramakeindexopts</code> .	193, 201, 566, 572, 818, 824, 1073, 1079, 1228, 1235, 1385, 1391, 1392	<code>\@@newglossaryentry@defcounters</code>	114, 114, 522, 773, 1030, 1184, 1185, 1341, 1342
<code>\@@gls@fixbraces</code>	219, 219, 581, 833, 1088, 1089, 1244, 1401	<code>\@@this@glo@</code>	69, 497, 747, 1003, 1158, 1315
<code>\@@gls@navhypertarget</code>	297, 297, 1121	<code>\@ACRfull</code>	269, 600, 852, 1107, 1262, 1263, 1428
<code>\@@gls@noexpand@assign@default@plurals</code>	285, 864, 911	<code>\@ACRfullpl</code>	270, 600, 601, 852, 1107, 1263, 1428
<code>\@@gls@noexpand@field</code>	25, 87, 88, 474, 506, 724, 756, 757, 978, 1012, 1013, 1133, 1167, 1288, 1324	<code>\@ACRlong</code>	169, 268, 445, 554, 600, 644, 805, 852, 897, 1059, 1106, 1215, 1262, 1372, 1427
<code>\@@gls@noidx@no@sanitizesort</code>	28, 475, 725, 979, 1134, 1289	<code>\@ACRlongpl</code>	171, 172, 268, 555, 601, 806, 853, 1060, 1106, 1216, 1262, 1373, 1427
<code>\@@gls@noidx@nosanitizesort</code>	27, 206, 475, 575, 725, 827, 979, 1082, 1134, 1238, 1288, 1395	<code>\@ACRshort</code>	165, 166, 268, 444, 551, 600, 643, 803, 852, 897, 1057, 1106, 1213, 1262, 1370, 1427
<code>\@@gls@noidx@sanitizesort</code>	26, 475, 724, 979, 1133, 1288	<code>\@ACRshortpl</code>	167, 268, 552, 553, 601, 804, 853, 1058, 1106, 1214, 1262, 1371, 1427
<code>\@@gls@nosanitizesort</code>	26, 26, 206, 474, 475, 575, 724, 827, 978, 979, 1082, 1133, 1238, 1288, 1395	<code>\@Acrfull</code>	269, 600, 852, 1107, 1262, 1428
<code>\@@gls@old@noidx@nosanitizesort</code>	27, 27, 1289	<code>\@Acrfullpl</code>	269, 270, 600, 852, 1107, 1263, 1428
<code>\@@gls@sanitizesort</code> .	26, 26, 206, 474, 475, 575, 724, 827, 978, 979, 1082, 1133, 1238, 1288, 1395	<code>\@Acrlong</code>	169, 268, 444, 553, 600, 644, 805, 852, 897, 1059, 1106, 1214, 1215, 1262, 1371, 1372, 1427
<code>\@@gls@xdycheckbackslash</code> . .	140, 541, 792, 793, 1046, 1201, 1358	<code>\@Acrlongpl</code>	171, 268, 554, 600, 806, 852, 1060, 1106, 1216, 1262, 1373, 1427
<code>\@@gls@xdycheckquote</code>	139, 140, 540, 792, 1045, 1046, 1200, 1201, 1357, 1358	<code>\@Acrshort</code>	165, 443, 551, 643, 803, 896, 1057, 1212, 1369
		<code>\@Acrshortpl</code>	167, 552, 804, 1058, 1213, 1370

<code>\@Glsdisp</code>	. 149, <u>149</u> , 1050, 1205, 1362	<code>\@Glsuseriv</code> 161,
<code>\@Glsentryfield</code>	95, <u>173</u> , 175–181,		162, 550, 801, 1055, 1211, 1368
	418, 419, 1017, 1061–1065,	<code>\@Glsuseriv@</code> 162,
	1172, 1217–1221, 1329, 1374–1378		<u>449</u> , 550, 801, 1055, 1211, 1368
<code>\@Glsfirst</code> 150,	<code>\@Glsuserv</code> 162,
	151, 546, 797, 1051, 1206, 1363		163, 550, 802, 1055, 1211, 1368
<code>\@Glsfirst@</code> 151, <u>446</u> , 546,	<code>\@Glsuserv@</code> 163,
	645, 797, 898, 1051, 1206, 1363		<u>450</u> , 550, 802, 1055, 1211, 1368
<code>\@Glsfirstplural</code> 152,	<code>\@Glsuservi</code> 163,
	153, 546, 798, 1051, 1207, 1364		164, 550, 802, 1056, 1211, 1368
<code>\@Glsfirstplural@</code> 153,	<code>\@Glsuservi@</code> 164,
	<u>446</u> , 546, 798, 1051, 1207, 1364		<u>450</u> , 550, 802, 1056, 1211, 1368
<code>\@Glsname</code> 153,	<code>\@Mi</code> 316, 693, 947
	154, 547, 798, 1052, 1207, 1364	<code>\@PGLS</code> 423, <u>423</u> , 683, 936
<code>\@Glsname@</code> 154, <u>447</u> , 547,	<code>\@PGLS@</code> 423, <u>423</u> , 683, 936
	645, 798, 899, 1052, 1207, 1364	<code>\@PGLSpl</code> 423, <u>423</u> , 683, 936
<code>\@Glspl</code>	146, 544, 795, 1049, 1204, 1361	<code>\@PGLSpl@</code> 423, <u>423</u> , 683, 936
<code>\@Glspl@</code>	116, 119, 146, <u>147</u> , 422, 523,	<code>\@PglS</code> 421, <u>421</u> , 682, 935
	525, 544, 682, 683, 774, 775,	<code>\@PglS@</code> 421, <u>421</u> , 682, 935
	795, 936, 1031, 1033, 1049,	<code>\@PglSpl</code> 422, <u>422</u> , 682, 935
	1186, 1187, 1204, 1343, 1344, 1361	<code>\@PglSpl@</code> 422, <u>422</u> , 682, 935
<code>\@Glsplural</code> 151,	<code>\@Roman</code>	212, 216, 578, 579, 830, 831,
	152, 546, 798, 1051, 1207, 1364		1085, 1086, 1241, 1242, 1398, 1399
<code>\@Glsplural@</code> 152, <u>446</u> , 546,	<code>\@acrfull</code> 268, <u>268</u> ,
	645, 798, 898, 1051, 1207, 1364		599, 851, 1106, 1262, 1427, 1428
<code>\@Glsymbol</code> 156,	<code>\@acrfullpl</code>
	157, 548, 799, 1053, 1209, 1366		. 269, 600, 852, 1107, 1263, 1428
<code>\@Glsymbol@</code> 157, <u>448</u> , 548,	<code>\@acrlong</code>	168, 267, <u>444</u> , 553, 599,
	646, 799, 899, 1053, 1209, 1366		644, 804, 805, 851, 897, 1058,
<code>\@Glsymbolplural</code> 157,		1059, 1106, 1214, 1262, 1371, 1427
	158, 548, 800, 1053, 1209, 1366	<code>\@acrlongpl</code>
<code>\@Glsymbolplural@</code>	158, <u>448</u> , 548,		170, 268, 554, 600, 806, 852,
	646, 800, 900, 1053, 1209, 1366		1060, 1106, 1215, 1262, 1372, 1427
<code>\@Glstext</code>	<code>\@acrshort</code>
	. 150, 545, 797, 1050, 1206, 1363		164, 267, 268, <u>443</u> , 551, 599,
<code>\@Glstext@</code> 150, <u>445</u> , 545,		600, 643, 802, 851, 852, 896,
	645, 797, 898, 1050, 1206, 1363		1056, 1106, 1212, 1262, 1369, 1427
<code>\@Glsuser@i</code>	448–450, 647, 648, 900, 901	<code>\@acrshortpl</code>
<code>\@Glsuseri</code> 158,		166, 268, 552, 600, 803, 852,
	159, 549, 800, 1054, 1209, 1366		1057, 1106, 1213, 1262, 1370, 1427
<code>\@Glsuseri@</code> 159,	<code>\@addtoacronymlists</code>
	<u>448</u> , 549, 800, 1054, 1209, 1366		. . . 22, 472, 722, 976, 1131, 1286
<code>\@Glsuserii</code> 159,	<code>\@addtoacronymlists</code> <u>22</u>
	160, 549, 801, 1054, 1210, 1367	<code>\@afterheading</code>
<code>\@Glsuserii@</code> 160,		304, 400, 406, 670, 687, 923, 941
	<u>449</u> , 549, 801, 1054, 1210, 1367	<code>\@alph</code> 212,
<code>\@Glsuseriii</code> 160,		215, 216, 578, 579, 830, 831,
	161, 549, 801, 1055, 1210, 1367		1085–1087, 1241, 1242, 1397, 1399
<code>\@Glsuseriii@</code> 161,		
	<u>449</u> , 549, 801, 1055, 1210, 1367		

<code>\@arabic</code>	212, 215, 216, 578, 579, 830, 831, 1085, 1086, 1241, 1242, 1397, 1399	<code>\@currentlabelname</code> 11, 251, 467, 592, 717, 844, 971, 1099, 1126, 1255, 1280, 1420
<code>\@auxout</code>	77, 79, 117, 141, 201, 203, 204, 207, 221, 225, 297, 501, 524, 572–574, 576, 582, 584, 684, 751, 752, 774, 775, 824–826, 828, 834, 836, 937, 1007, 1008, 1031, 1032, 1079, 1081, 1083, 1089, 1091, 1092, 1121, 1161, 1162, 1186, 1202, 1235, 1236, 1239, 1245, 1247, 1318, 1319, 1343, 1359, 1391–1393, 1395, 1402, 1404	<code>\@curroptions</code> 44, 484, 734, 990, 1144, 1300
<code>\@backslashchar</code>	133, 134, 140, 195, 535, 541, 786, 787, 792, 793, 820, 1040, 1041, 1046, 1074, 1195, 1196, 1201, 1230, 1352, 1353, 1358, 1387	<code>\@declaredoptions</code> 44, 484, 734, 990, 1144, 1300
<code>\@bsphack</code> 211, 577, 829, 1084, 1240, 1397	<code>\@delimN</code>	598, 850
<code>\@cGls</code>	118, <u>118</u> , 524, 775, 1032, 1187, 1344	<code>\@delimR</code>	598, 850
<code>\@cGls@</code>	116, 118, <u>118</u> , 523, 524, 774, 775, 1031, 1032, 1186, 1187, 1343, 1344	<code>\@disable@onlypremakeg</code> <u>49</u> , 201, 485, 572, 736, 824, 991, 1079, 1146, 1235, 1303, 1392
<code>\@cGlspl</code> 119, 524, 775, 1032, 1187, 1344		<code>\@disable@premakecs</code>	49, <u>49</u> , 486, 736, 991, 1146, 1303
<code>\@cGlspl@</code>	116, 119, <u>119</u> , 523–525, 774, 775, 1031, 1032, 1186, 1187, 1343, 1344	<code>\@disabled@glssaddxdycounters</code> <u>61</u> , 62, 492, 742, 743, 998, 1152, 1153, 1309, 1310
<code>\@cclv</code>	316, 693, 694, 947	<code>\@do@addcounter</code>	61, 491, 492, 742, 997, 1152, 1309
<code>\@cgls</code>	117, <u>117</u> , 524, 775, 1032, 1186, 1343	<code>\@do@auxoutstuff</code>	225, 584, 836, 1091, 1092, 1247, 1404
<code>\@cgls@</code>	115, 117, <u>117</u> , 523, 524, 773, 775, 1030, 1032, 1185, 1187, 1342, 1344	<code>\@do@glossentry</code> ..	594, 628, 846, 881
<code>\@cglspl</code>	118, <u>118</u> , <u>119</u> , 524, 775, 1032, <u>1187</u> , 1344	<code>\@do@gls@checkesc</code> 134, 787, 1041, 1196, 1353
<code>\@cglspl@</code>	116, 118, <u>118</u> , 523, 524, 774, 775, 1031, 1032, 1185, 1187, 1342, 1344	<code>\@do@gls@getcounterprefix</code> 217, 218, 578, 579, 830, 831, 1087, 1088, 1243, 1400
<code>\@chapter</code>	48, 485, 735, 991, 1146, 1302, 1303	<code>\@do@gls@islistofacronyms</code> 22, 472, 722, 976, 1131, 1286
<code>\@classoptionslist</code> 44, 484, 734, 990, 1144, 1300	<code>\@do@glssee</code> 107, 518, 769, 1025, 1180, 1337
<code>\@closegls</code> 197, <u>197</u> , 198, 569, 570, 821, 822, 1076, 1077, 1232, 1389		<code>\@do@ifinlist</code> 60, 491, 741, 997, 1152, 1309
<code>\@colht</code>	316, 694, 947	<code>\@do@newglossaryentry</code> .	271, 601, 611–624, 657–661, 853, 863– 878, 910–914, 1108, 1264, 1429
<code>\@colroom</code>	316, 694, 947	<code>\@do@seeglossary</code> .	203, <u>218</u> , 219, 573, 581, 825, 832, 833, 1081, 1088, 1089, 1236, 1244, 1393, 1401
		<code>\@do@subglossentry</code> 595, 628, 847, 881	
		<code>\@do@wrglossary</code>	130, <u>211</u> , 534, 577, 664, 784, 829, 917, 1038, 1084, 1193, 1240, 1350, 1397
		<code>\@do@write@glslabels</code> <u>37</u> , 38, 481, 731, 985, 986, 1140, 1141, 1296	
		<code>\@do@writeaux@info</code> 221, 582, 834, 1089, 1245, 1402
		<code>\@doendpe</code> ..	224, 400, 406, 1247, 1404
		<code>\@domakeglossaries</code> <u>36</u> , 37, 201, 203, 480,

	481, 572, 573, 730, 731, 824, 825, 985, 1079, 1080, 1139, 1140, 1234, 1236, 1295, 1391, 1393		985, 1134, 1140, 1289, 1290, 1295
<code>\@ehc</code> 316, 693, 947	<code>\@firstofthree</code>	127, 143, 146, 148, 164, 166, 168, 170, 443–445, 532, 542, 543, 545, 551–554, 643, 644, 783, 794–796, 802, 804–806, 896–898, 1037, 1047, 1049, 1050, 1056, 1057, 1059, 1060, 1191, 1203–1205, 1212– 1215, 1348, 1360–1362, 1369–1372
<code>\@empty</code> 15, 22, 40, 43, 44, 48, 60, 62, 65, 68, 69, 103, 108, 109, 131, 132, 143–147, 164– 172, 187, 190, 192, 198, 199, 209, 211, 218, 260, 443–445, 469, 471, 472, 482, 484, 485, 491, 492, 494, 496, 497, 516, 519, 534, 542–544, 551–555, 562, 564, 566, 570, 571, 576, 577, 580, 596, 612, 614–621, 623, 625, 643, 644, 664, 666– 668, 718, 720, 722, 732, 734, 735, 741, 742, 745, 747, 766, 767, 770, 785, 794–796, 802– 807, 814, 816, 818, 821–823, 828, 829, 832, 848, 864, 866, 868, 870, 872–874, 876, 878, 896–898, 918, 920, 921, 972, 974–976, 987–991, 997, 998, 1000, 1002, 1003, 1022, 1023, 1026, 1038, 1039, 1047–1049, 1056–1061, 1068, 1071, 1072, 1076–1078, 1083, 1084, 1088, 1103, 1127, 1129, 1131, 1142, 1144–1146, 1152, 1153, 1155, 1157, 1158, 1177, 1181, 1193, 1194, 1203–1205, 1212–1216, 1224, 1226, 1228, 1232–1234, 1239, 1240, 1243, 1259, 1282, 1286, 1298–1300, 1303, 1309, 1310, 1312, 1314, 1315, 1334, 1338, 1350, 1351, 1360–1362, 1369–1373, 1381, 1383, 1385, 1389, 1390, 1396, 1397, 1400, 1424	<code>\@firstoftwo</code> 32, 33, 92, 93, 110, 127, 146, 147, 166–168, 170–172, 478, 479, 508, 509, 532, 543, 544, 552– 555, 727–729, 759, 760, 783, 795, 796, 803, 804, 806, 807, 982, 983, 1015, 1016, 1027, 1037, 1049, 1057, 1058, 1060, 1061, 1137, 1138, 1170, 1171, 1182, 1191, 1204, 1205, 1213– 1216, 1292, 1293, 1327, 1328, 1339, 1348, 1361, 1362, 1370–1373
<code>\@end@fixbraces</code> 219, 581, 833, 1088, 1089, 1244, 1401	<code>\@for</code> 33, 44, 49, 51, 60, 62, 69, 92, 93, 133, 187–189, 200–202, 212, 219, 234, 272, 479, 484, 485, 491, 492, 497, 508, 509, 535, 562, 563, 572, 573, 578, 581, 585, 602, 612, 614, 616, 617, 619, 622, 623, 625, 664, 684, 685, 729, 734, 736, 742, 747, 759, 760, 786, 814, 815, 824, 825, 830, 833, 836, 837, 854, 864, 867, 868, 870, 872, 875, 876, 878, 918, 937, 938, 983, 990–992, 997, 998, 1003, 1015, 1016, 1040, 1068–1070, 1078–1080, 1085, 1089, 1092, 1109, 1121, 1138, 1144, 1146, 1147, 1152, 1153, 1157, 1158, 1170, 1171, 1195, 1224, 1225, 1234, 1235, 1240, 1244, 1248, 1265, 1293, 1300, 1303, 1309, 1310, 1314, 1315, 1327, 1328, 1352, 1381, 1382, 1391, 1392, 1397, 1401, 1409, 1430
<code>\@end@fortrue</code> 33, 73, 93, 479, 499, 509, 684, 729, 749, 760, 937, 983, 1004, 1016, 1121, 1138, 1159, 1171, 1293, 1316, 1328	<code>\@glo@@desc</code> 105, 517, 768, 1024, 1179, 1336
<code>\@esphack</code> 211, 577, 829, 1084, 1240, 1397	<code>\@glo@@symbol</code> 106, 518, 769, 1025, 1179, 1336
<code>\@expandtwoargs</code> 44, 484, 734, 990, 1144, 1300	<code>\@glo@access</code>	426, 428, 429, 434, 628, 630, 631, 633, 634, 882–884, 887
<code>\@firstofone</code>	28, 37, 302, 475, 476, 481, 725, 731, 940, 979, 980,		

<code>\@glo@addchildren</code>	234, 234, 239, 585, 588, 837, 840, 1092, 1095, 1248, 1251, 1409, 1412	<code>\@glo@descpluralaccess</code>	427, 429, 430, 629–631, 882–884
<code>\@glo@assign@sortkey</code> . .	200, 202, 204, 252, 573, 574, 592, 824, 826, 844, 1080, 1081, 1100, 1235, 1237, 1255, 1392, 1394, 1421	<code>\@glo@do@sortentries</code> 234, 235, 585, 836, 837, 1092, 1093, 1248, 1249, 1409, 1410
<code>\@glo@autosee</code>	106, 106, 518, 769, 1025, 1180, 1337	<code>\@glo@entry</code>	184, 185, 561, 812, 1067, 1223, 1380
<code>\@glo@autoseehook</code>	107, 107, 518, 769, 1025, 1180, 1337	<code>\@glo@entryprefix</code> 417, 418, 679, 680, 932, 933
<code>\@glo@check@mkidxrangear</code> 132, 217, 535, 580, 664, 786, 831, 917, 1039, 1087, 1194, 1243, 1351, 1399	<code>\@glo@entryprefixfirst</code> 417, 418, 679, 680, 932, 933
<code>\@glo@check@sortallowed</code> . . .	16, 17–20, 202, 206, 469–471, 573, 575, 719–721, 825, 827, 973– 975, 1080, 1082, 1128–1130, 1235, 1238, 1282–1284, 1392, 1395	<code>\@glo@entryprefixfirstplural</code> 417, 418, 679, 680, 932, 933
<code>\@glo@childlist</code> 234, 585, 837, 1092, 1248, 1409	<code>\@glo@entryprefixplural</code> 417, 418, 679, 680, 932, 933
<code>\@glo@counter</code>	83, 102, 105, 504, 515, 517, 754, 766, 768, 1010, 1022, 1024, 1164, 1176, 1179, 1321, 1333, 1336	<code>\@glo@esclabel</code> 108, 109, 519, 520, 770, 771, 1026, 1027, 1181, 1338	
<code>\@glo@counterprefix</code> 207, 214, 216–	218, 260, 264, 576, 578–580, 596, 598, 828, 830–832, 848, 850, 1083, 1086–1088, 1103, 1105, 1239, 1241–1243, 1259, 1260, 1396, 1398–1400, 1424, 1426	<code>\@glo@etext</code>	525–527, 776–778
<code>\@glo@default@sorttype</code> . .	16, 16, 204, 237, 238, 469, 574, 587, 588, 719, 826, 838–840, 973, 1081, 1094, 1095, 1128, 1237, 1250, 1251, 1282, 1394, 1411, 1412	<code>\@glo@first</code>	82, 101, 104, 105, 503, 515, 517, 518, 621, 660, 753, 766, 768, 874, 914, 1009, 1022, 1024, 1164, 1176, 1178, 1179, 1321, 1333, 1335, 1336
<code>\@glo@defaultcounter</code> 105, 517, 768, 1024, 1179, 1336	<code>\@glo@firstaccess</code> 426, 429, 629–631, 882–884
<code>\@glo@desc</code>	82, 100, 101, 103, 105, 503, 514– 518, 753, 765, 766, 768, 1009, 1021, 1022, 1024, 1164, 1176, 1177, 1179, 1321, 1333, 1334, 1336	<code>\@glo@firstplural</code> 82, 101, 104, 105, 503, 515, 517, 660, 753, 766, 768, 914, 1009, 1022, 1024, 1164, 1176, 1178, 1321, 1333, 1335	
<code>\@glo@descaccess</code>	426, 429, 430, 629–631, 882–884	<code>\@glo@firstpluralaccess</code> 426, 429, 629–631, 882–884
<code>\@glo@descplural</code> . .	82, 100, 101, 503, 514, 515, 753, 765, 1009, 1021, 1022, 1164, 1176, 1321, 1333	<code>\@glo@grabfirst</code>	245, 245, 589, 841, 1096, 1252, 1416
		<code>\@glo@label</code> . . .	84, 86, 94, 95, 97– 107, 114, 182, 183, 412, 418, 429, 430, 505, 510, 512–518, 522, 559, 560, 630–632, 680, 711, 712, 755, 756, 761, 763– 769, 773, 811, 812, 884, 885, 933, 965, 1010–1012, 1017, 1019–1025, 1030, 1066, 1067, 1165, 1166, 1171–1180, 1185, 1221, 1222, 1322, 1323, 1328– 1337, 1342, 1378, 1379, 1448, 1449
		<code>\@glo@list</code> 107, 518, 769, 1025, 1180, 1337
		<code>\@glo@long</code>	74, 86, 102, 105, 499, 505, 515, 517, 749, 755, 766, 768, 1005,

1011, 1022, 1024, 1160, 1166, 1177, 1179, 1317, 1323, 1334, 1336	\@glo@see 83, 102, 106, 107, 504, 515, 518, 754, 766, 769, 1010, 1022, 1025, 1164, 1176, 1180, 1321, 1333, 1337
\@glo@longaccess 427, 429, 430, 629–631, 882–884	\@glo@seeautonumberlist 12, 13, 84, 468, 504, 717, 754, 971, 1010, 1126, 1165, 1281, 1322
\@glo@longpl 86, 102, 105, 505, 515, 517, 611, 613, 615, 617, 618, 621, 622, 657, 755, 766, 768, 863, 864, 866, 867, 869, 871, 874, 875, 910, 1011, 1022, 1024, 1166, 1177, 1179, 1323, 1334, 1336	\@glo@short 75, 86, 102, 105, 499, 505, 515, 517, 660, 750, 755, 766, 768, 914, 1005, 1011, 1022, 1024, 1160, 1166, 1177, 1179, 1317, 1323, 1334, 1336
\@glo@longpluralaccess 427, 429, 430, 629–631, 882–884	\@glo@shortaccess 427, 429, 430, 629–631, 657–660, 882–884, 910–912, 914
\@glo@name 17, 81, 87, 101, 104, 105, 470, 503, 506, 515–517, 719, 753, 756, 765, 767, 768, 973, 1009, 1012, 1021, 1023, 1024, 1128, 1164, 1167, 1176, 1178, 1179, 1283, 1320, 1324, 1333, 1335, 1336	\@glo@shortpl 86, 102, 105, 505, 515, 517, 611, 613, 615, 617, 618, 621, 622, 657, 660, 755, 766, 768, 863– 865, 867, 869, 871, 874, 875, 910, 914, 1011, 1022, 1024, 1166, 1177, 1179, 1323, 1334, 1336
\@glo@no@assign@sortkey . . 200, 202, 252, 573, 592, 824, 844, 1080, 1100, 1235, 1255, 1392, 1421	\@glo@shortpluralaccess 427, 429, 430, 629–631, 882–884
\@glo@nonumberlist 85, 504, 505, 755, 1011, 1165, 1322	\@glo@sort 17, 19, 26–28, 82, 102, 104, 108, 109, 470, 471, 475, 503, 515, 517, 519, 520, 719, 721, 724, 725, 753, 766, 767, 770, 771, 973–975, 979, 1009, 1022, 1023, 1026, 1027, 1128, 1130, 1133, 1134, 1164, 1176, 1178, 1181, 1283, 1284, 1288, 1289, 1321, 1333, 1335, 1338
\@glo@numfmt 217, 580, 664, 832, 917, 1087, 1243, 1400	\@glo@sortedinsert 235, 235, 585, 586, 837, 838, 1093, 1248, 1249, 1410
\@glo@parent 84, 102–104, 108, 109, 235, 471, 504, 515, 516, 519, 520, 585, 720, 754, 766, 767, 770, 771, 837, 974, 1010, 1022, 1023, 1026, 1027, 1093, 1129, 1165, 1176–1178, 1181, 1248, 1249, 1322, 1333–1335, 1338, 1410	\@glo@sortentries 232, 233, 234, 584, 587, 588, 836, 838, 839, 1092, 1094, 1095, 1247, 1250, 1251, 1408, 1409
\@glo@plural 82, 101, 104, 503, 515–517, 659, 753, 766– 768, 912, 1009, 1022–1024, 1164, 1176, 1178, 1321, 1333, 1335	\@glo@sorthandler@case 233, 236, 586, 587, 838, 839, 1093, 1095, 1249, 1250, 1408, 1411
\@glo@pluralaccess 426, 429, 629–631, 882–884	\@glo@sorthandler@letter 233, 236, 586, 587, 838, 839, 1093, 1094, 1249, 1250, 1408, 1410
\@glo@prefix 13, 84, 102, 108, 109, 132, 133, 217, 468, 504, 515, 519, 520, 535, 580, 664, 717, 754, 766, 770, 771, 786, 832, 917, 972, 1010, 1022, 1026, 1027, 1039, 1040, 1087, 1126, 1165, 1176, 1181, 1194, 1243, 1281, 1322, 1333, 1338, 1351, 1400	\@glo@sorthandler@nocase 233, 236, 586, 588, 838, 839, 1094, 1095, 1249, 1251, 1408, 1411
\@glo@range 217, 580, 664, 832, 917, 1087, 1243, 1399, 1400	

<code>\@glo@sorthandler@word</code>	766, 769, 1009, 1022, 1025, 1164, 1176, 1179, 1321, 1333, 1336
. 232, 236, 586, 587, 838, 1093, 1094, 1249, 1250, 1408, 1410	
<code>\@glo@sortinghandler</code>	<code>\@glo@symbolpluralaccess</code>
234, 235, 584, 586, 836, 838, 1092, 1093, 1247, 1249, 1409, 1410 426, 429, 629–631, 882–884
<code>\@glo@sortinglist</code> 234, 235, 238, 239, 584–586, 588, 836, 837, 840, 1092, 1093, 1095, 1247–1249, 1251, 1409, 1410, 1412	<code>\@glo@text</code> 82, 101, 104, 106, 144–148, 173–175, 503, 515, 517, 518, 542–545, 555, 556, 617, 659, 680, 753, 766, 767, 769, 794–796, 807, 808, 869, 912, 933, 1009, 1022, 1023, 1025, 1048–1050, 1061, 1062, 1164, 1176, 1178, 1179, 1203– 1205, 1217, 1218, 1321, 1333, 1335, 1336, 1360–1362, 1374, 1375
<code>\@glo@sortmacro@case</code>	<code>\@glo@textaccess</code> 426, 429, 628, 630, 631, 657–659, 882–884, 910–913
. 238, 587, 839, 1095, 1250, 1412	<code>\@glo@thislabel</code> 107, 519, 769, 770, 1026, 1180, 1337
<code>\@glo@sortmacro@def</code>	<code>\@glo@thislettergrp</code> 232, 243, 245, 246, 589, 590, 841, 842, 1096, 1097, 1252, 1253, 1408, 1415–1417
. 238, 588, 840, 1095, 1251, 1412	<code>\@glo@thisvalue</code> 75, 76, 500, 750, 1006, 1160, 1161, 1317, 1318
<code>\@glo@sortmacro@def@do</code>	<code>\@glo@tmp</code> 77, 94, 95, 218, 510, 580, 761, 832, 1007, 1017, 1088, 1161, 1171, 1172, 1243, 1318, 1328, 1329, 1400
. 239, 588, 840, 1095, 1251, 1412	<code>\@glo@type</code> 11, 83, 101–103, 105, 106, 183, 184, 201, 209, 222, 223, 225, 234, 235, 239–241, 244, 250, 251, 271, 467, 471, 503, 515–518, 560, 561, 572, 576, 577, 583–585, 588, 589, 591, 592, 601, 612, 614, 615, 617, 619, 621, 623, 625, 684, 685, 717, 720, 753, 765–769, 812, 824, 828, 829, 834–837, 840, 841, 843, 844, 853, 864, 866, 868, 870, 872, 874, 876, 878, 937, 938, 971, 974, 975, 1009, 1022–1025, 1067, 1079, 1083, 1084, 1090–1093, 1095, 1096, 1098, 1099, 1108, 1126, 1129, 1164, 1176, 1177, 1179, 1180, 1222, 1223, 1235, 1239, 1246– 1248, 1251, 1252, 1254, 1255, 1263, 1280, 1321, 1333, 1334, 1336, 1337, 1379, 1380, 1392, 1396, 1403, 1404, 1409, 1410, 1412, 1413, 1416, 1419, 1420, 1429
<code>\@glo@sortmacro@letter</code>	
. 237, 587, 839, 1094, 1250, 1411	
<code>\@glo@sortmacro@lettercase</code> 237	
<code>\@glo@sortmacro@nocase</code>	
. 238, 587, 839, 1095, 1251, 1412	
<code>\@glo@sortmacro@standard</code>	
. 238, 587, 839, 1094, 1250, 1411	
<code>\@glo@sortmacro@use</code>	
. 239, 588, 840, 1096, 1251, 1413	
<code>\@glo@sortmacro@word</code>	
. 237, 586, 838, 1094, 1250, 1411	
<code>\@glo@sortmacro@wordcase</code> 237	
<code>\@glo@sorttype</code> 204, 240, 244, 245, 252, 574, 588, 589, 592, 826, 840, 841, 844, 1081, 1096, 1100, 1237, 1251, 1252, 1255, 1394, 1413, 1416, 1417, 1421	
<code>\@glo@storeentry</code>	
. . 17, 19, 108, 469–471, 519, 719, 720, 770, 973–975, 1026, 1128, 1129, 1181, 1283, 1284, 1338	
<code>\@glo@suffix</code> 132, 133, 217, 535, 580, 664, 786, 832, 917, 1039, 1040, 1087, 1194, 1243, 1351, 1400	
<code>\@glo@symbol</code> 74, 83, 101, 106, 499, 503, 515, 518, 616, 620, 749, 753, 766, 769, 869, 873, 1005, 1009, 1022, 1025, 1160, 1164, 1176, 1179, 1317, 1321, 1333, 1336	
<code>\@glo@symbolaccess</code> 426, 429, 629–631, 660, 882–884, 913	
<code>\@glo@symbolplural</code> 83, 101, 106, 503, 515, 518, 753,	

<code>\@glo@types</code>	37,	<code>\@glolist@</code>	
69, <u>77</u> , 79, 113, 114, 184, 185,		. 103, 516, 767, 1023, 1177, 1334	
200–202, 293, 411, 481, 497,		<code>\@gloname</code>	
501, 521, 522, 561, 572, 573,		. 105, 517, 768, 1024, 1179, 1336	
627, 711, 731, 747, 751, 772,		<code>\@glossary@default@style</code>	
773, 812, 824, 825, 880, 965,	 11, <u>11</u> , 13, 14, 223,	
986, 1003, 1006, 1007, 1029,		261, 289, 467, 468, 583, 597,	
1030, 1067, 1078–1080, 1120,		625, 717, 718, 835, 849, 878,	
1140, 1157, 1158, 1161, 1162,		971, 972, 1090, 1103, 1118,	
1184, 1223, 1234–1236, 1276,		1119, 1126, 1127, 1246, 1259,	
1296, 1314, 1315, 1318, 1319,		1274, 1280, 1281, 1403, 1424, 1440	
1341, 1380, 1391, 1392, 1441, 1448		<code>\@glossaryentryfield</code>	
<code>\@glo@useri</code>	85,	107, 108, 109, 519, 520, 770,	
102, 105, 505, 515, 517, 755,		1026, 1027, 1180, 1181, 1337, 1338	
766, 768, 1011, 1022, 1024,		<code>\@glossarysection</code>	
1166, 1176, 1179, 1323, 1333, 1336	 57, <u>58</u> , 489, 490, 739,	
<code>\@glo@useriaccess</code>		740, 995, 1149, 1150, 1306, 1307	
427, 429, 430, 629–631, 882–884		<code>\@glossarystyle</code>	
<code>\@glo@userii</code>	85,	223, 224,	
102, 105, 505, 515, 517, 755,		250, 583, 591, 835, 843, 1090,	
766, 768, 1011, 1022, 1024,		1091, 1099, 1246, 1254, 1403, 1420	
1166, 1177, 1179, 1323, 1334, 1336		<code>\@glossarysubentryfield</code>	
<code>\@glo@useriiaccess</code>	427, 107, 108, 109, 519, 520,	
429, 430, 629–631, 882, 883, 885		770, 771, 1026, 1027, 1181, 1338	
<code>\@glo@useriii</code>	85,	<code>\@gls</code>	143,
102, 105, 505, 515, 517, 755,		<u>143</u> , 542, 794, 1047, 1203, 1360	
766, 768, 1011, 1022, 1024,		<code>\@gls@</code>	115, 117,
1166, 1177, 1179, 1323, 1334, 1336		143, <u>143</u> , 420–422, 523, 524,	
<code>\@glo@useriiiaccess</code>	427,	542, 681, 682, 774, 775, 794,	
429, 430, 629–631, 882, 883, 885		934, 935, 1031, 1032, 1047,	
<code>\@glo@useriv</code>	86,	1185, 1187, 1203, 1342, 1344, 1360	
102, 105, 505, 515, 517, 755,		<code>\@gls@automake@immediate</code>	
766, 768, 1011, 1022, 1024,	 200, 201, 572,	
1166, 1177, 1179, 1323, 1334, 1336		823, 824, 1078, 1079, 1234, 1391	
<code>\@glo@userivaccess</code>		<code>\@gls@link</code>	128, <u>128</u> , 129,
427, 429, 430, 629–631, 883–885		532, 533, 783, 1037, 1192, 1349	
<code>\@glo@userv</code>	86,	<code>\@gls@Hcounter</code> 131, 132, 534, 785,	
102, 105, 505, 515, 517, 755,		1038, 1039, 1193, 1194, 1350, 1351	
766, 768, 1011, 1022, 1024,		<code>\@gls@ReturnAfterFi</code>	264,
1166, 1177, 1179, 1323, 1334, 1336		598, 599, 850, 1105, 1261, 1426	
<code>\@glo@uservaccess</code>		<code>\@gls@access@display</code> ..	<u>434</u> , 633, 887
428–430, 629, 630, 632, 883–885		<code>\@gls@actualchar</code> 109, 136, 138, 191,	
<code>\@glo@uservi</code>	86,	520, 537, 538, 540, 565, 667,	
102, 105, 505, 515, 517, 755,		770, 771, 788, 789, 791, 816,	
766, 768, 1011, 1022, 1024,		817, 920, 1026, 1027, 1042,	
1166, 1177, 1179, 1323, 1334, 1336		1044, 1045, 1071, 1181, 1197,	
<code>\@glo@userviaccess</code>		1199, 1227, 1338, 1354, 1356, 1384	
428–430, 629, 630, 632, 883–885		<code>\@gls@addpredefinedattributes</code> .	
<code>\@glodesc</code> <u>62</u> , 187, 196, 493,	
. 105, 518, 768, 1025, 1179, 1336		562, 569, 743, 814, 821, 998,	

999, 1068, 1076, 1153, 1154,
 1224, 1231, 1310, 1311, 1381, 1388
 \@gls@adjustmode 184,
 184, 560, 812, 1067, 1222, 1379
 \@gls@after
 22, 472, 473, 722, 977, 1131, 1286
 \@gls@assign@symbol@field 26
 \@gls@assign@symbolplural@field 26
 \@gls@automake . . . 197, 202, 569,
 570, 573, 821, 822, 825, 1076,
 1077, 1080, 1232, 1235, 1389, 1392
 \@gls@automake@immediate
 . 198, 200, 570–572, 822–824,
 1077, 1078, 1233, 1234, 1390, 1391
 \@gls@automake@types 200, 200, 202,
 1078, 1080, 1234, 1235, 1391, 1392
 \@gls@before
 . . . 22, 472, 722, 977, 1131, 1286
 \@gls@between
 . . . 298, 684, 685, 938, 1121, 1122
 \@gls@body 175, 556, 807,
 808, 1062, 1217, 1218, 1374, 1375
 \@gls@checkactual
 134, 138, 194, 536,
 540, 567, 787, 791, 819, 1041,
 1044, 1045, 1074, 1195, 1199,
 1200, 1230, 1352, 1356, 1357, 1387
 \@gls@checkboxar 134, 137, 194, 536,
 539, 568, 787, 790, 819, 1041,
 1043, 1044, 1074, 1195, 1198,
 1199, 1230, 1352, 1355, 1356, 1387
 \@gls@checkedmkidx 133–
 140, 193–195, 535–541, 566–
 568, 786–793, 818–820, 1040–
 1046, 1073–1075, 1195–1201,
 1229–1231, 1352–1358, 1386–1388
 \@gls@checkesc
 134, 139, 195, 787, 820,
 1041, 1074, 1196, 1230, 1353, 1387
 \@gls@checkescactual 134,
 135, 194, 536–538, 567, 787–
 789, 819, 1040, 1042, 1074,
 1195, 1197, 1229, 1352, 1354, 1386
 \@gls@checkescbar 134, 136, 195,
 536, 538, 568, 787, 789, 819,
 1041–1043, 1074, 1195, 1197,
 1198, 1230, 1352, 1354, 1355, 1387
 \@gls@checkesclevel 134,
 136, 195, 538, 539, 787, 789,
 790, 820, 1041, 1043, 1074,
 1196, 1198, 1230, 1353, 1355, 1387
 \@gls@checkescquote 134,
 135, 194–196, 536, 537, 567–
 569, 787, 788, 819, 820, 1040–
 1042, 1074, 1075, 1195–1197,
 1229–1231, 1352–1354, 1386–1388
 \@gls@checklevel 134, 138,
 195, 536, 539, 568, 787, 790,
 791, 819, 1041, 1044, 1074,
 1196, 1199, 1230, 1353, 1356, 1387
 \@gls@checkmkidxchars
 108, 132, 133, 194, 203,
 216, 218, 519, 535, 536, 567,
 573, 579, 581, 664, 770, 786,
 787, 819, 825, 831, 832, 917,
 1026, 1039, 1040, 1074, 1080,
 1087, 1088, 1181, 1194, 1195,
 1229, 1236, 1242, 1244, 1338,
 1351, 1352, 1386, 1393, 1399, 1401
 \@gls@checkquote 134,
 134, 193, 194, 536, 537, 566,
 567, 787, 788, 818, 819, 1040,
 1041, 1073, 1074, 1195, 1196,
 1228, 1229, 1352, 1353, 1385, 1386
 \@gls@classI 188, 562, 563,
 814, 1069, 1224, 1225, 1381, 1382
 \@gls@classII
 . 188, 563, 814, 1069, 1225, 1382
 \@gls@codepage
 . 68, 225, 496, 584, 746, 836,
 1002, 1092, 1157, 1247, 1314, 1404
 \@gls@counter 126,
 130–132, 183, 184, 207, 217,
 218, 532–535, 560, 576, 580,
 664, 783–785, 812, 828, 832,
 917, 1036, 1038, 1039, 1067,
 1083, 1087, 1088, 1191, 1193,
 1194, 1222, 1239, 1243, 1244,
 1348, 1350, 1351, 1379, 1396, 1400
 \@gls@counterwithin
 15, 15, 469, 718,
 719, 972, 973, 1127, 1128, 1282
 \@gls@ctr
 60, 61, 491, 742, 997, 1152, 1309
 \@gls@currentlettergroup
 244, 246, 588, 590, 840, 842,
 1096, 1097, 1251, 1253, 1416, 1417
 \@gls@debugfalse
 7, 465, 715, 969, 1123, 1278

<code>\@gls@debugtrue</code>	7, 465, 714, 715, 968, 969, 1123, 1124, 1278	<code>\@gls@deprecated@acr@warning</code>	288, 866, 868, 870, 872, 874, 876, 1118, 1274, 1439
<code>\@gls@declareoption</code>	12, 12, 13, 14, 21, 24, 25, 32, 35–38, 40, 42, 43, 467, 468, 471–474, 478, 480, 481, 483, 484, 717, 718, 721–724, 728, 730, 731, 733, 734, 971, 972, 975–978, 982, 984–987, 989, 1126, 1127, 1130–1133, 1137, 1139–1144, 1280, 1281, 1285, 1287, 1292, 1294–1297, 1299	<code>\@gls@do@acronymsdef</code>	21, 21, 22, 47, 48, 80, 471, 472, 485, 502, 721, 722, 735, 752, 976, 990, 991, 1008, 1130, 1131, 1145, 1163, 1285, 1302, 1320
<code>\@gls@default</code>	120, 525, 776, 1033, 1187, 1188, 1344, 1345	<code>\@gls@do@automake@err</code>	39, 40, 986, 987, 1141, 1142, 1296, 1297
<code>\@gls@default@value</code>	74–76, 82, 87, 88, 101, 102, 104, 105, 287, 417, 418, 499, 500, 503, 506, 507, 515–518, 620, 679, 680, 749, 750, 753, 756, 757, 765–768, 873, 933, 1005, 1006, 1009, 1012, 1013, 1022–1025, 1118, 1160, 1161, 1164, 1167, 1168, 1176, 1178, 1179, 1274, 1317, 1318, 1321, 1324, 1325, 1333, 1335, 1336, 1439	<code>\@gls@do@glsunset</code>	127, 127, 144–148, 1036, 1048–1050, 1191, 1203–1205, 1348, 1360–1362
<code>\@gls@define@value</code>	74–76, 82, 87, 88, 101, 102, 104, 105, 287, 417, 418, 499, 500, 503, 506, 507, 515–518, 620, 679, 680, 749, 750, 753, 756, 757, 765–768, 873, 933, 1005, 1006, 1009, 1012, 1013, 1022–1025, 1118, 1160, 1161, 1164, 1167, 1168, 1176, 1178, 1179, 1274, 1317, 1318, 1321, 1324, 1325, 1333, 1335, 1336, 1439	<code>\@gls@do@indexdef</code>	43, 44, 47, 48, 80, 484, 485, 502, 734, 735, 752, 989–991, 1008, 1144, 1145, 1163, 1299, 1302, 1320
<code>\@gls@defaulttype</code>	77, 77, 1007, 1161, 1318	<code>\@gls@do@numbersdef</code>	43, 47, 48, 80, 483–485, 502, 734, 735, 752, 989–991, 1008, 1144, 1145, 1163, 1299, 1302, 1320
<code>\@gls@deffile</code>	90–92, 508, 509, 758, 759, 1014, 1015, 1169, 1170, 1326, 1327	<code>\@gls@do@symbolsdef</code>	42, 80, 483, 502, 733, 752, 989, 1008, 1143, 1163, 1299, 1320
<code>\@gls@define@glossaryentrycounter</code>	15, 15, 48, 251, 253, 469, 485, 592, 593, 718, 719, 735, 844, 845, 972, 973, 990, 1099, 1100, 1127, 1145, 1255, 1256, 1282, 1302, 1420, 1421	<code>\@gls@do@symbolssdef</code>	47, 48, 485, 735, 990, 991, 1145, 1302
<code>\@gls@define@glossarysubentrycounter</code>	15, 48, 252, 253, 469, 485, 592, 593, 719, 735, 844, 845, 973, 991, 1099, 1100, 1128, 1145, 1255, 1256, 1282, 1302, 1420, 1421	<code>\@gls@doautomake</code>	40, 40, 200, 202, 482, 572, 573, 732, 824, 825, 987, 1079, 1080, 1142, 1234, 1235, 1297, 1391, 1392
<code>\@gls@defsort</code>	17–20, 106, 470, 471, 518, 719–721, 769, 973–975, 1025, 1128–1130, 1179, 1283, 1284, 1336	<code>\@gls@docheckquotedef</code>	193–196, 566–569, 818–821, 1073–1075, 1228–1231, 1385–1388
<code>\@gls@defsortcount</code>	17–20, 79, 470, 471, 502, 719–721, 752, 973–975, 1008, 1128–1130, 1162, 1283, 1284, 1319	<code>\@gls@docloadedfalse</code>	6, 464, 714, 968, 1123, 1277
		<code>\@gls@docloadedtrue</code>	6, 464, 714, 968, 1123, 1277
		<code>\@gls@dodedeflistparser</code>	202, 573, 824, 1079, 1080, 1235, 1392
		<code>\@gls@doentrycounterdef</code>	48, 485, 735, 990, 991, 1145, 1302
		<code>\@gls@doentrydef</code>	531, 532, 782, 783
		<code>\@gls@dolast</code>	219, 220, 581, 833, 1089, 1244, 1245, 1401
		<code>\@gls@donext</code>	219, 220, 581, 833, 1089, 1244, 1245, 1401
		<code>\@gls@donext@def</code>	182, 183, 560, 811, 812, 1066, 1222, 1379
		<code>\@gls@dosubentrycounterdef</code>	48, 485, 735, 991, 1145, 1302

<code>\@gls@dothiswrite</code>	570, 571, 822, 823, 1076–1078, 1232–1234	<code>\@gls@fixbraces</code>	107, 219, 518, 581, 769, 833, 1025, 1088, 1180, 1244, 1337, 1401
<code>\@gls@elem</code>	684, 937, 1121	<code>\@gls@forbidtexext</code> . .	77, 79, 501, 751, 752, 1007, 1162, 1318, 1319
<code>\@gls@enablesavenonumberlist</code> . .	85, 91, 504, 508, 755, 759, 1011, 1015, 1165, 1169, 1322, 1326	<code>\@gls@fulllist</code> 37, 731, 986, 1140, 1296	
<code>\@gls@encapchar</code>	136, 137, 191, 217, 219, 538, 539, 565, 580, 581, 664, 667, 789, 790, 816, 817, 832, 833, 917, 920, 1043, 1044, 1071, 1087, 1088, 1197–1199, 1227, 1243, 1244, 1354–1356, 1384, 1400, 1401	<code>\@gls@get@counterprefix</code>	218, 580, 832, 1088, 1243, 1400
<code>\@gls@entry@count</code>	116, 117, 117, 523, 524, 774, 775, 1031, 1032, 1186, 1343	<code>\@gls@getbody</code>	175, 556, 807, 1062, 1217, 1374
<code>\@gls@entry@field</code> 94, 95, 115, 141, 172, 173–182, 430–432, 510, 511, 522, 555–559, 632, 633, 761, 773, 807–811, 885, 886, 1017, 1030, 1061–1065, 1171, 1172, 1185, 1202, 1216–1221, 1328, 1329, 1342, 1359, 1373–1378		<code>\@gls@getcounter</code>	80, 502, 752, 1008, 1163, 1320
<code>\@gls@escbsdq</code>	133, 134, 192, 535, 536, 566, 668, 786, 787, 817, 818, 921, 1040, 1072, 1195, 1228, 1352, 1385	<code>\@gls@getcounterprefix</code>	218, 218, 578– 580, 830–832, 1088, 1243, 1400
<code>\@gls@expand@fields</code>	88, 88, 89, 507, 757, 1013, 1167, 1168, 1324, 1325	<code>\@gls@getgrouptitle</code>	203, 259, 259, 298, 399, 573, 596, 685, 825, 848, 938, 1081, 1102, 1122, 1236, 1258, 1393, 1423
<code>\@gls@expandonce</code>	89, 507, 757, 1013, 1168, 1325	<code>\@gls@glossary</code> 210, 210, 211, 577, 829, 1084, 1239, 1240, 1396, 1397	
<code>\@gls@extramakeindexopts</code>	196, 201, 569, 572, 821, 824, 1075, 1079, 1231, 1235, 1388, 1392	<code>\@gls@glslink@hyper@update@hook</code>	126, 126, 1036, 1191, 1348
<code>\@gls@fetch@lettergrp</code>	232, 233, 242, 246, 1408, 1415, 1417	<code>\@gls@gobbleopt</code>	78, 501, 751, 1007, 1162, 1319
<code>\@gls@fetchfield</code>	76, 93, 500, 509, 750, 760, 1006, 1016, 1161, 1171, 1318, 1328	<code>\@gls@grptitle</code>	259, 295, 298, 596, 684, 685, 848, 937, 938, 1102, 1121, 1122, 1258, 1423
<code>\@gls@field@link</code>	96, 149, 149, 150–164, 445– 450, 511, 512, 545–551, 645– 648, 762, 797–802, 898–901, 1018, 1050–1056, 1172, 1173, 1206–1212, 1329, 1330, 1363–1369	<code>\@gls@hyp@opt</code>	96, 117–119, 127, 128, 143–171, 268–270, 420–423, 511, 512, 524, 532, 542–555, 599, 600, 681–683, 762, 775, 783, 794– 806, 851, 852, 934–936, 1018, 1032, 1037, 1047–1060, 1106, 1107, 1172, 1173, 1186, 1187, 1191, 1192, 1203–1216, 1262, 1263, 1329, 1330, 1343, 1344, 1348, 1349, 1360–1373, 1427, 1428
<code>\@gls@fieldaccess@display</code> .	434, 434, 435–437, 634–636, 887–889	<code>\@gls@hyp@opt@cs</code> .	127, 128, 532, 783, 1037, 1191, 1192, 1348, 1349
<code>\@gls@firsttok</code>	244, 245, 589, 841, 1096, 1252, 1416	<code>\@gls@hypergroup</code>	297, 298, 684, 937, 1121
		<code>\@gls@if@measuring</code>	109, 110, 1027, 1181, 1182, 1338, 1339
		<code>\@gls@ifinlist</code>	60, 61, 491, 741, 742, 997, 1152, 1309
		<code>\@gls@ifnotmeasuring</code>	520, 771
		<code>\@gls@igtype</code>	81, 503, 753, 1009, 1163, 1320

<code>\@gls@increment@currcount</code>	802–806, 1050, 1056–1061, 115, 116 , 522, 523, 773, 774, 1030, 1031, 1185, 1186, 1342, 1343
<code>\@gls@indexdef</code>	43 , 43 , 484, 734, 989, 1144, 1299
<code>\@gls@initnonumberlist</code> 85 , 85 , 102, 504, 515, 754, 755, 766, 1010, 1011, 1022, 1165, 1176, 1322, 1333
<code>\@gls@islistofacronyms</code> 22, 472, 722, 976, 1131, 1286
<code>\@gls@keylist</code>	461, 657, 910
<code>\@gls@keymap</code>	85, 92, 92 , 93–95, 417, 428, 505, 508– 510, 630, 679, 755, 759–761, 883, 932, 1011, 1015–1017, 1166, 1170–1172, 1323, 1327–1329
<code>\@gls@label</code>	203, 204, 207, 214, 217, 573, 574, 576, 579, 580, 825, 828, 830–832, 1081, 1083, 1086, 1087, 1236, 1239, 1242, 1243, 1393, 1396, 1398–1400
<code>\@gls@labelsfile</code>	37, 38, 481, 731, 986, 1140, 1141, 1296
<code>\@gls@langmod</code>	36, 198, 199, 225, 570, 571, 822, 823, 984, 1076, 1077, 1091, 1139, 1232, 1233, 1247, 1294, 1389, 1390, 1404
<code>\@gls@levelchar</code> 109, 137, 138, 191, 520, 538, 539, 565, 667, 771, 789, 790, 816, 817, 920, 1027, 1043, 1044, 1071, 1181, 1198, 1199, 1227, 1338, 1355, 1356, 1384	
<code>\@gls@link</code>	128, 130 , 144 – 149 , 165 – 172 , 443 – 445 , 533, 542 – 545 , 551 – 555 , 643 , 644 , 783, 784, 794–797, 802–807, 896–898, 1037, 1038, 1048–1050, 1056–1061, 1192, 1193, 1203–1206, 1212–1216, 1349 , 1350, 1360 – 1363 , 1369 – 1373
<code>\@gls@link@checkfirsthyper</code> 129 , 143 – 148 , 533, 542 – 545 , 784, 794–796, 1037, 1047–1050, 1192, 1203–1205, 1349, 1360–1362	
<code>\@gls@link@label</code> 130, 533, 613, 618, 784, 865, 871, 1038, 1193, 1350	
<code>\@gls@link@nocheckfirsthyper</code> 149 , 149 , 164–172, 545, 551–555, 797,	802–806, 1050, 1056–1061, 1206, 1212–1216, 1363, 1369–1373
	<code>\@gls@link@opts</code> 130, 533, 613, 618, 784, 865, 871, 1038, 1193, 1350
	<code>\@gls@list</code> . . 684, 685, 937, 938, 1121
	<code>\@gls@listsuffix</code> 60, 491, 741, 997, 1152, 1309
	<code>\@gls@loadlist</code> 13 , 13 , 14, 288, 468, 625, 718, 878, 972, 1118, 1126, 1127, 1274, 1281, 1439
	<code>\@gls@loadlong</code> . . 13 , 13 , 14, 288, 468, 625, 717, 718, 878, 972, 1118, 1126, 1127, 1274, 1281, 1439
	<code>\@gls@loadshellesc</code> 45, 47, 1300–1302
	<code>\@gls@loadsuper</code> 13 , 13 , 14, 288, 468, 625, 718, 878, 972, 1118, 1126, 1127, 1274, 1281, 1439
	<code>\@gls@loadtree</code> 14 , 14 , 288, 468, 625, 718, 878, 972, 1118, 1127, 1274, 1281, 1439
	<code>\@gls@local@increment@currcount</code> 115, 116 , 522, 523, 773, 774, 1030, 1031, 1185, 1186, 1342, 1343
	<code>\@gls@loclist</code> . 204, 205, 241–243, 245, 246, 574, 575, 589, 590, 826, 827, 841, 842, 1081, 1082, 1097, 1237, 1238, 1252, 1253, 1394, 1395, 1413–1415, 1417, 1418
	<code>\@gls@map</code> 92, 93, 508, 509, 759, 760, 1015, 1016, 1170, 1171, 1327, 1328
	<code>\@gls@missinglang@warn</code> 24, 24 , 25, 52, 474, 487, 723, 737, 978, 993, 1132, 1147, 1287, 1304
	<code>\@gls@missingnumberlist</code> 87 , 105, 506, 517, 756, 768, 1012, 1024, 1166, 1179, 1323, 1336
	<code>\@gls@noaccess</code> 428 , 434, 630, 633, 634, 883, 887
	<code>\@gls@noexpand@field</code> 87
	<code>\@gls@noexpand@fields</code> 87 , 89, 506, 507, 756, 757, 1012, 1013, 1167, 1168, 1324, 1325
	<code>\@gls@nohyperlist</code> 23 , 23 , 24 , 81, 129, 473, 502, 503, 533, 723, 753, 784, 977, 1009, 1038, 1132, 1163, 1192, 1286, 1320, 1349
	<code>\@gls@noidx@addtorefs</code> . 204, 207, 207
	<code>\@gls@noidx@do</code> 244, 245 , 589, 840, 841, 1096, 1097, 1252, 1416, 1417

<code>\@gls@noidx@getgrouptitle</code>	1076–1078, 1232–1234, 1389–1391
203, <u>259</u> , 573, 596, 825, 848,	
1081, 1102, 1236, 1258, 1393, 1423	
<code>\@gls@noidx@sanitizesort</code>	
. 26, 206, 475, 575, 724, 827,	
979, 1082, 1133, 1238, 1288, 1395	
<code>\@gls@noidx@setsanitizesort</code>	
. 30, <u>30</u> , 206,	
477, 575, 726, 727, 827, 981,	
1082, 1135, 1136, 1238, 1291, 1395	
<code>\@gls@noidx@loclist@finalsep</code> 205,	
575, 826, 827, 1082, 1237, 1394	
<code>\@gls@noidx@loclist@prev</code>	
. 205, 247, 575,	
590, 591, 827, 842, 843, 1082,	
1098, 1237, 1253, 1254, 1394, 1418	
<code>\@gls@noidx@loclist@sep</code>	
. 205, 247, 574, 575,	
590, 591, 826, 842, 843, 1082,	
1098, 1237, 1253, 1254, 1394, 1418	
<code>\@gls@noist@toolate</code> 191,	
193, <u>196</u> , 1071, 1072, 1076,	
1227, 1228, 1231, 1384, 1385, 1388	
<code>\@gls@nofef@warn</code> . 203, <u>207</u> , 241,	
244, 573, 576, 589, 825, 828,	
841, 1080, 1083, 1096, 1236,	
1238, 1252, 1393, 1395, 1413, 1416	
<code>\@gls@notranslatorhook</code> <u>31</u>	
<code>\@gls@numberlink</code>	
. 263, 598, 850, 1104, 1260, 1425	
<code>\@gls@numbersdef</code> 43,	
<u>43</u> , 483, 484, 734, 989, 1144, 1299	
<code>\@gls@numlist@lastsep</code> . 182, 183,	
560, 811, 812, 1066, 1222, 1379	
<code>\@gls@numlist@nextsep</code> . 182, 183,	
560, 811, 812, 1066, 1222, 1379	
<code>\@gls@numlist@sep</code> 182, 183,	
560, 811, 812, 1066, 1222, 1379	
<code>\@gls@old@chapter</code> 48,	
485, 735, 991, 1146, 1302, 1303	
<code>\@gls@old@print@noidx@glossary</code>	
. <u>244</u> , 1416	
<code>\@gls@oldnewglossaryentryposthook</code>	
. 429, 630, 884	
<code>\@gls@oldnewglossaryentryprehook</code>	
. 428, 630, 883	
<code>\@gls@onlypremakeg</code> 48, <u>48</u> ,	
49, 485, 735, 736, 991, 1146, 1303	
<code>\@gls@order</code>	
. 197–199, 570, 571, 821–823,	
1076–1078, 1232–1234, 1389–1391	
<code>\@gls@org@LT@output</code> . . . 316, 693, 947	
<code>\@gls@org@glsnoidxdisplayloc</code>	205,
. 206, 575, 827, 1082, 1238, 1395	
<code>\@gls@org@glsseeformat</code> . . . 205,	
206, 575, 827, 1082, 1238, 1395	
<code>\@gls@override@glossary</code>	
. <u>41</u> , 42, 482, 483,	
733, 988, 989, 1143, 1298, 1299	
<code>\@gls@patchtabularx</code>	
. 111, 520, 771, 1028, 1183, 1340	
<code>\@gls@preglossaryhook</code>	
. <u>222</u> , 224, 582, 583, 834,	
835, 1090, 1091, 1246, 1247, 1403	
<code>\@gls@prevlevel</code> 329, 330,	
412–415, 674, 675, 700, 701,	
712, 713, 927, 928, 953, 954,	
966, 967, 1444, 1445, 1449, 1450	
<code>\@gls@provide@newglossary</code>	
. <u>77</u> , 79, 501, 751, 752,	
1006–1008, 1161, 1162, 1318, 1319	
<code>\@gls@quotechar</code>	
. 135–138, 191, 193–195,	
536–540, 565–568, 667, 788–	
791, 816–820, 920, 1041–1045,	
1071, 1073, 1075, 1196–1199,	
1227–1231, 1353–1356, 1384–1388	
<code>\@gls@reference</code> 203, 204, 207, <u>247</u> ,	
573, 574, 576, 591, 825, 826,	
828, 843, 1081, 1083, 1098,	
1236, 1239, 1254, 1393, 1395, 1418	
<code>\@gls@removedoption</code> . . <u>29</u> , 34, 35,	
41, 980, 984, 988, 1135, 1138,	
1139, 1142, 1143, 1290, 1294, 1298	
<code>\@gls@removespaces</code> 263, 264, 598,	
850, 1104, 1105, 1260, 1261, 1426	
<code>\@gls@renewglossary</code>	
197, <u>210</u> , 569, 577, 821, 829,	
1076, 1084, 1231, 1240, 1388, 1397	
<code>\@gls@replacementtext</code>	
. 433, 434, 633, 886, 887	
<code>\@gls@requested@languages</code>	
. 33, <u>33</u> , 51,	
983, 992, 1138, 1147, 1293, 1303	
<code>\@gls@rest</code> 175,	
556, 807, 808, 1062, 1218, 1375	
<code>\@gls@restore@glslocal</code> <u>127</u> , 131,	
1036, 1038, 1191, 1193, 1348, 1350	

<code>\@gls@restoreat</code> .. 91, 508, 759, 1015, 1169, 1326	<code>\@gls@setupsort@clear</code> .. 20, 975, 1130, 1284
<code>\@gls@roman</code> 64, 493, 494, 665, 744, 918, 999, 1000, 1154, 1311	<code>\@gls@setupsort@def</code> 17, 470, 720, 974, 1129, 1283
<code>\@gls@run@output@dir</code> .. 39, 40, 45, 46, 1297, 1300, 1301	<code>\@gls@setupsort@none</code> 19, 471, 721, 975, 1129, 1284
<code>\@gls@run@output@file</code> ... 45, 46, 198, 199, 1300, 1301, 1389–1391	<code>\@gls@setupsort@standard</code> ... 17, 469, 470, 719, 973, 974, 1128, 1283
<code>\@gls@run@unrestricted@shell</code> .. 44, 45, 198–200, 1300, 1389–1391	<code>\@gls@setupsort@use</code> 18, 470, 720, 974, 1129, 1283
<code>\@gls@runshell</code> 44, 44, 45, 47, 198, 199, 1300, 1301, 1389, 1391	<code>\@gls@sort</code> .. 232, 233, 242, 246, 590, 842, 1097, 1253, 1408, 1409, 1415, 1417
<code>\@gls@sanitized@tmp</code> .. 133, 535, 536, 786, 1040, 1195, 1352	<code>\@gls@sort@A</code> .. 236, 586, 838, 1093, 1094, 1249, 1410, 1411
<code>\@gls@sanitized@desc</code> .. 25, 474, 479, 724, 729	<code>\@gls@sort@B</code> .. 236, 586, 838, 1093, 1094, 1249, 1410, 1411
<code>\@gls@sanitizename</code> .. 26, 474, 724, 978, 1133, 1288	<code>\@gls@startswithexpandonce</code> 88, 89, 507, 757, 1013, 1168, 1325
<code>\@gls@sanitizesort</code> .. 17, 26, 470, 474, 719, 724, 973, 978, 1128, 1133, 1283, 1288	<code>\@gls@storenonumberlist</code> ... 85, 105, 504, 517, 754, 755, 768, 1011, 1024, 1165, 1179, 1322, 1336
<code>\@gls@sanitizesymbol</code> ... 26, 474, 479, 480, 724, 729, 978, 1133, 1288	<code>\@gls@symbolsdef</code> .. 42, 42, 483, 733, 989, 1143, 1299
<code>\@gls@save@glslocal</code> 126, 127, 130, 1036, 1038, 1191, 1193, 1348, 1350	<code>\@gls@this</code> .. 212, 578, 830, 1085, 1240, 1397
<code>\@gls@save@entrycounter</code> 130, 131, 184, 534, 560, 784, 785, 812, 1038, 1067, 1193, 1222, 1350, 1379	<code>\@gls@this@Hloc</code> .. 218, 580, 832, 1088, 1243, 1400
<code>\@gls@savenonumberlist</code> .. 84, 85, 85, 206, 504, 754, 755, 1010, 1011, 1083, 1165, 1238, 1322, 1395	<code>\@gls@this@entryname</code> .. 38, 731, 986, 1140, 1141, 1296
<code>\@gls@see@noindex</code> .. 9, 9, 10, 84, 466, 467, 504, 716, 754, 970, 1010, 1125, 1165, 1279, 1280, 1322	<code>\@gls@this@field</code> .. 76, 500, 750, 1006, 1161, 1318
<code>\@gls@setacrstyle</code> 34, 47, 48, 479, 480, 484, 485, 729, 735, 983, 990, 991, 1138, 1145, 1294, 1302	<code>\@gls@this@label</code> 73, 219, 220, 238, 239, 498, 581, 588, 749, 833, 840, 1004, 1089, 1095, 1159, 1244, 1245, 1251, 1316, 1401, 1412
<code>\@gls@setcounter</code> ... 79, 80, 502, 752, 1008, 1162, 1163, 1319, 1320	<code>\@gls@this@lang</code> .. 51, 992, 1147, 1303, 1304
<code>\@gls@setdefault@glslink@opts</code> .. 129, 130, 533, 784, 1038, 1193, 1350	<code>\@gls@this@list</code> .. 182, 183, 560, 811, 812, 1066, 1222, 1379
<code>\@gls@setsort</code> .. 17–20, 130, 184, 470, 471, 534, 560, 719–721, 784, 812, 974, 975, 1038, 1067, 1128–1130, 1193, 1222, 1283, 1284, 1350, 1379	<code>\@gls@this@loc</code> .. 218, 580, 832, 1088, 1243, 1400
<code>\@gls@setupshortcuts</code> .. 47, 48, 484, 485, 735, 990, 991, 1145, 1302	<code>\@gls@this@val</code> .. 93, 509, 760, 1016, 1171, 1328
	<code>\@gls@title</code> .. 56, 489, 739, 995, 1149, 1306
	<code>\@gls@tmp</code> .. 9, 51, 64, 65, 89, 133, 211, 264, 466, 471, 486, 494, 507, 535, 536, 577, 684, 685, 715, 720, 737, 744, 745,

<code>\@glsdesc</code>	<code>\@glsminrange</code>
. 154, 547, 799, 1052, 1208, 1365 187, 187, 188, 562, 563,
<code>\@glsdesc@</code>	665, 666, 813–815, 918, 919,
646, 799, 899, 1052, 1208, 1365	1068, 1069, 1224, 1225, 1381, 1382
<code>\@glsdescplural</code>	<code>\@glsname</code>
155,	. 153, 547, 798, 1052, 1207, 1364
547, 799, 1052, 1053, 1208, 1365	<code>\@glsname@</code>
<code>\@glsdescplural@</code> . 155, 447, 547,	153, 447, 547,
646, 799, 899, 1053, 1208, 1365	645, 798, 899, 1052, 1207, 1364
<code>\@glsdisp</code>	<code>\@glsnavhypertarget</code>
148, 148, 297, 297, 684, 937, 1121
149, 545, 796, 1050, 1205, 1362	<code>\@glsnextpages</code>
<code>\@glsentry</code> . 37, 38, 113, 114, 117,	223, 252, 583, 593, 835, 844,
481, 521, 522, 524, 731, 772–	1091, 1100, 1246, 1256, 1403, 1421
775, 986, 1029–1032, 1140,	<code>\@glsnodesc</code>
1141, 1184, 1186, 1296, 1341, 1343	86,
<code>\@glsentrytitlecase</code> ... 178, 178,	101, 103, 105, 505, 515, 518,
557, 558, 809, 1064, 1219, 1376	756, 765, 766, 768, 1012, 1021,
<code>\@glsfirst</code>	1022, 1024, 1025, 1166, 1176,
150,	1177, 1179, 1323, 1333, 1334, 1336
546, 797, 1050, 1051, 1206, 1363	<code>\@glsnoname</code>
<code>\@glsfirst@</code> 86, 101, 104, 105, 505, 515–
150, 446, 546,	517, 755, 765, 767, 768, 1011,
645, 797, 898, 1051, 1206, 1363	1021, 1023, 1024, 1166, 1176,
<code>\@glsfirstletter</code>	1178, 1179, 1323, 1333, 1335, 1336
186, 186,	<code>\@glsnonextpages</code>
561, 813, 1067, 1068, 1223, 1380	223, 252, 583, 592, 835, 844,
<code>\@glsfirstplural</code>	1091, 1100, 1246, 1255, 1403, 1421
. 152, 546, 798, 1051, 1207, 1364	<code>\@glsnumberformat</code>
<code>\@glsfirstplural@</code>	126,
..... 152, 446, 446, 546,	130, 183, 184, 207, 217, 532,
645, 798, 898, 1051, 1207, 1364	533, 560, 576, 580, 664, 783,
<code>\@glsnumber</code> . 262, 262, 598,	784, 812, 828, 831, 832, 917,
849, 850, 1104, 1259, 1260, 1425	1036, 1038, 1067, 1083, 1087,
<code>\@glsisacronymlistfalse</code>	1191, 1193, 1222, 1239, 1243,
... 23, 473, 722, 977, 1131, 1286	1348, 1350, 1379, 1396, 1399, 1400
<code>\@glsisacronymlisttrue</code>	<code>\@glsopenfile</code> 197, 197, 209,
... 23, 473, 722, 977, 1131, 1286	569, 577, 821, 828, 1076, 1084,
<code>\@glslink</code>	1231, 1232, 1239, 1388, 1389, 1396
110, 111, 130, 142, 142, 143,	<code>\@glsorder</code>
183, 295, 534, 541, 542, 560,	35,
684, 785, 793, 794, 812, 937,	201, 480, 572, 729, 824, 984,
1027, 1028, 1038, 1047, 1067,	1079, 1139, 1235, 1294, 1391, 1392
1121, 1182, 1193, 1202, 1203,	<code>\@glspl</code>
1222, 1339, 1350, 1359, 1360, 1379	145,
<code>\@glslocalreset</code>	146, 543, 795, 1048, 1204, 1361
112,	<code>\@glspl@</code>
113, 115, 521, 522, 771–773,	116, 118,
1028–1030, 1183–1185, 1340–1342	146, 146, 421, 422, 523, 524,
<code>\@glslocalunset</code>	543, 681–683, 774, 775, 795,
112,	935, 936, 1031, 1032, 1048,
112, 115, 521, 522, 772, 773,	1186, 1187, 1204, 1343, 1344, 1361
1029, 1030, 1184, 1185, 1341, 1342	<code>\@glsplural</code>
<code>\@glslocref</code> ... 207, 214, 216, 217,	151, 546,
576, 578–580, 664, 828, 830–	798, 1051, 1206, 1207, 1363, 1364
832, 917, 1083, 1086, 1087,	<code>\@glsplural@</code>
1239, 1241–1243, 1396, 1398–1400	151, 446, 546,
	645, 798, 898, 1051, 1207, 1364

<code>\@glsprefix@record@hook</code>	<code>\@glsuserv@</code>
. 419, 420–423, 681–683, 934–937	162, 450, 550,
<code>\@glsreset</code>	647, 802, 901, 1055, 1211, 1368
111,	<code>\@glsuservi</code>
113, 115, 520–522, 771–773,	. 163, 550, 802, 1056, 1211, 1368
1028–1030, 1183–1185, 1340–1342	<code>\@glsuservi@</code>
<code>\@glssee</code> 107, 219, 518, 581, 769, 833,	163, 450, 550,
1025, 1089, 1180, 1244, 1337, 1401	648, 802, 901, 1056, 1211, 1368
<code>\@glsshowaccsupp</code>	<code>\@glswidestname</code>
7, 8, 9, 425,	362, 371,
465, 466, 628, 715, 716, 881,	412, 412, 413, 414, 674, 675,
969, 970, 1123, 1124, 1278, 1279	712, 713, 928, 965, 966, 1449, 1450
<code>\@glsshowtarget</code>	<code>\@glswritefiles</code>
7, 8, 8,	. 40, 209, 482, 576, 732, 828,
465, 466, 541, 714, 715, 793,	987, 1083, 1142, 1239, 1298, 1396
968–970, 1123, 1124, 1278, 1279	<code>\@glsxtr@doaccsupp</code>
<code>\@glsymbol</code>	424, 627, 881
156, 548,	<code>\@glsxtr@record</code>
799, 1053, 1208, 1209, 1365, 1366	420, 681, 934
<code>\@glsymbol@</code>	<code>\@gobble</code>
156, 448, 548,	7, 17–19, 37, 38, 92,
646, 799, 899, 1053, 1209, 1366	110, 133, 185, 186, 189, 201,
<code>\@glsymbolplural</code>	203, 302, 465, 470, 471, 480,
. 157, 548, 800, 1053, 1209, 1366	481, 508, 509, 520, 535, 561,
<code>\@glsymbolplural@</code> 157, 448, 548,	563, 564, 572, 573, 663, 666,
646, 800, 899, 1053, 1209, 1366	667, 714, 715, 719–721, 730,
<code>\@glstarget</code>	731, 759, 760, 771, 786, 812,
142,	813, 815, 824, 825, 916, 919,
142, 143, 255, 297, 542, 594,	920, 940, 968, 969, 974, 975,
684, 793, 794, 846, 937, 1047,	985, 986, 1015, 1016, 1027,
1101, 1121, 1203, 1257, 1360, 1422	1040, 1067, 1070, 1079, 1080,
<code>\@glstext</code>	1123, 1128, 1129, 1140, 1141,
. 149, 545, 797, 1050, 1206, 1363	1170, 1182, 1195, 1223, 1225,
<code>\@glstext@</code>	1226, 1235, 1236, 1278, 1283,
149, 445, 545,	1284, 1295, 1296, 1327, 1339,
645, 797, 898, 1050, 1206, 1363	1352, 1380, 1382, 1383, 1392, 1393
<code>\@glsunset</code>	<code>\@gobblethree</code>
112,
112, 115, 521, 522, 772, 773,	7, 465, 715, 969, 1123, 1278
1029, 1030, 1183–1185, 1340–1342	<code>\@idxitem</code>
<code>\@glsuseri</code>	406, 708, 962, 1445, 1446
. 158, 548, 800, 1054, 1209, 1366	<code>\@ifclassloaded</code>
<code>\@glsuseri@</code>
158, 448, 548,	6, 14, 57, 464, 468, 489,
647, 800, 900, 1054, 1209, 1366	714, 718, 739, 968, 972, 995,
<code>\@glsuserii</code>	1123, 1127, 1150, 1277, 1281, 1307
. 159, 549, 800, 1054, 1210, 1367	<code>\@ifglossaryexists</code>
<code>\@glsuserii@</code>	70, 70,
159, 449, 549,	221, 497, 582, 747, 748, 834,
647, 800, 900, 1054, 1210, 1367	1003, 1089, 1158, 1245, 1315, 1402
<code>\@glsuseriii</code>	<code>\@ifl@t@r</code> 29, 476, 726, 980, 1135, 1290
160,	<code>\@ifnextchar</code>
549, 801, 1054, 1055, 1210, 1367	. 79, 127, 502, 532, 752, 783,
<code>\@glsuseriii@</code>	1008, 1037, 1162, 1191, 1319, 1348
160, 449, 549,	<code>\@ifpackageloaded</code>
647, 801, 900, 1055, 1210, 1367	6,
<code>\@glsuseriv</code>	11, 13, 31–33, 51, 68, 110, 182,
161, 549,	193, 424, 464, 467, 478, 479,
801, 1055, 1210, 1211, 1367, 1368	486, 496, 520, 559, 566, 627,
<code>\@glsuseriv@</code>	714, 717, 727–729, 736, 747,
161, 449, 550,	771, 811, 818, 880, 968, 971,
647, 801, 900, 1055, 1211, 1368	
<code>\@glsuserv</code>	
. 162, 550, 801, 1055, 1211, 1368	

972, 982, 983, 992, 1002, 1027, 1066, 1073, 1123, 1126, 1136– 1138, 1147, 1157, 1182, 1221, 1228, 1277, 1280, 1281, 1292, 1293, 1304, 1314, 1339, 1378, 1385	761, 769, 884, 933, 1017, 1025, 1171, 1172, 1180, 1328, 1329, 1337
<code>\@ifstar</code> 70, 78, 94, 127, 186, 265, 497, 501, 510, 532, 561, 599, 747, 751, 760, 761, 783, 813, 851, 1003, 1007, 1016, 1017, 1037, 1067, 1068, 1105, 1158, 1162, 1171, 1172, 1191, 1223, 1261, 1315, 1319, 1328, 1329, 1348, 1380, 1426	<code>\@newglossaryentryprehook</code> 94, 95, 100, 102, 107, 417, 428, 510, 514, 515, 519, 630, 679, 761, 765, 766, 769, 883, 933, 1017, 1021, 1022, 1025, 1171, 1172, 1176, 1177, 1180, 1328, 1329, 1333, 1334, 1337
<code>\@ifundefined</code> 50, 306, 317, 330, 331, 338, 413, 414, 486, 675, 684, 688, 694, 701, 705, 712, 713, 736, 928, 937, 942, 948, 954, 955, 958, 959, 966, 992, 1146, 1303, 1449, 1450	<code>\@nil</code> 22, 107, 132–134, 175, 194, 195, 217, 219, 233, 244, 245, 262– 264, 473, 518, 535, 536, 556, 567, 568, 580, 581, 589, 590, 598, 664, 722, 769, 786, 787, 807, 819, 820, 831, 833, 841, 842, 849, 850, 917, 977, 1025, 1039–1041, 1062, 1074, 1087, 1088, 1096, 1097, 1104, 1105, 1131, 1180, 1194–1196, 1217, 1229, 1230, 1243, 1244, 1252, 1253, 1259–1261, 1286, 1337, 1351–1353, 1374, 1386, 1387, 1399, 1401, 1409, 1416, 1425, 1426
<code>\@ignored@glossaries</code> 37, 80, 81, 81, 481, 502, 503, 731, 752, 753, 986, 1008, 1009, 1140, 1163, 1296, 1320	<code>\@nnil</code> . 22, 220, 473, 581, 722, 833, 977, 1089, 1131, 1244, 1286, 1401
<code>\@input@</code> 225, 584, 835, 1091, 1247, 1404	<code>\@no@makeglossaries</code> 36, 37, 202, 204, 206, 480, 481, 573, 574, 576, 730, 824, 826, 828, 985, 1080, 1081, 1083, 1139, 1140, 1235, 1237, 1238, 1295, 1392, 1394, 1395
<code>\@istfilename</code> 54, 201, 488, 572, 738, 824, 994, 1079, 1148, 1235, 1305, 1391, 1392	<code>\@no@post@desc</code> 53, 74, 488, 669, 738, 922, 993, 1148, 1305
<code>\@makecol</code> 316, 693, 694, 947	<code>\@nopostdesc</code> 53, 223, 487, 583, 738, 835, 993, 1091, 1148, 1246, 1305, 1403
<code>\@makeglossaries@warn@noprintglossary</code> 202, 202, 824, 825, 1079, 1080, 1235, 1236, 1392	<code>\@onelevel@sanitize</code> 9, 26, 27, 38, 64, 92, 109, 133, 186, 190, 218, 221, 245, 264, 466, 475, 494, 508, 520, 535, 536, 561, 562, 564, 581, 582, 589, 665, 666, 715, 724, 725, 731, 744, 759, 770, 786, 813, 816, 832, 834, 841, 918, 920, 970, 979, 986, 999, 1015, 1026, 1040, 1068, 1070, 1088, 1089, 1096, 1105, 1124, 1133, 1141, 1154, 1170, 1181, 1195, 1223, 1224, 1226, 1244, 1245, 1252, 1260, 1279, 1288, 1296,
<code>\@makeglossary</code> 37, 197, 201, 480, 481, 569, 572, 730, 731, 821, 824, 985, 1076, 1079, 1140, 1231, 1235, 1295, 1388, 1392	
<code>\@minus</code> . . 302, 325, 374, 406, 687, 697, 708, 940, 951, 962, 1442, 1445	
<code>\@mkboth</code> 57, 489, 490, 739, 740, 995, 1150, 1307	
<code>\@ne</code> 110, 1027, 1182, 1339	
<code>\@newglossary</code> 77, 79, 80, 501, 502, 751, 752, 1007, 1008, 1161–1163, 1318–1320	
<code>\@newglossaryentry@defcounters</code> 106, 107, 114, 518, 519, 522, 769, 773, 1025, 1030, 1180, 1185, 1337, 1342	
<code>\@newglossaryentryposthook</code> 94, 95, 106, 107, 418, 429, 510, 518, 519, 630, 680,	

1311, 1327, 1338, 1352, 1380, 1381, 1383, 1401, 1402, 1416, 1426	<code>\@plus</code> . . . 302, 325, 374, 406, 687, 697, 708, 940, 951, 962, 1442, 1445
<code>\@onlypreamble</code> 79, 90, 101, 116, 120, 202, 206, 502, 508, 514, 523, 525, 573, 575, 752, 758, 765, 774, 776, 825, 827, 1008, 1014, 1021, 1031, 1033, 1080, 1083, 1163, 1168, 1176, 1186, 1188, 1235, 1238, 1320, 1325, 1333, 1343, 1345, 1392, 1395	<code>\@print@glossary</code> 222, <u>224</u> , 582, 584, 834, 835, 1090, 1091, 1245, 1247, 1402, 1404
<code>\@onlypremakeg</code> . . . 48, 53–55, 61, 62, 66, 80, 193, 485, 488, 492, 495, 502, 566, 735, 738, 742, 743, 745, 752, 818, 991, 994, 997, 998, 1001, 1008, 1073, 1146, 1148, 1149, 1152, 1153, 1155, 1163, 1228, 1303, 1305, 1306, 1309, 1310, 1312, 1320, 1385	<code>\@print@noidx@glossary</code> 222, <u>239</u> , 582, 588, 834, 840, 1090, 1096, 1246, 1251, 1402, 1413
<code>\@org@Gls@acrentryname</code> <u>174</u> , 1062, 1217, 1374	<code>\@printgloss@checkexists</code> <u>221</u> , 222, 223, 582, 583, 834, 835, 1090, 1245, 1246, 1402, 1403
<code>\@org@glossaryentrynumbers</code> 223, <u>224</u> , 583, 835, 1090, 1091, 1246, 1247, 1403, 1404	<code>\@printgloss@checkexists@allowignored</code> <u>221</u> , 582, 834, 1090, 1245, 1402
<code>\@org@Gls@assign@descplural</code> 612, 619, 621, 623, 657, 660, 661, 864, 871, 872, 874, 876, 911, 913, 914	<code>\@printgloss@checkexists@noignored</code> 221, <u>221</u> , 222, 582, 834, 1089, 1090, 1245, 1246, 1402
<code>\@org@Gls@assign@firstpl</code> 612–615, 617, 619, 621–623, 657–661, 863, 864, 866–872, 874, 876, 910–914	<code>\@printgloss@setsort</code> 202, 204, <u>222</u> , 223, 573, 574, 582, 583, 824, 826, 834, 835, 1080, 1081, 1090, 1235, 1237, 1246, 1392, 1394, 1403
<code>\@org@Gls@assign@plural</code> 612–615, 617, 619, 621–623, 657–661, 864, 866–872, 874, 876, 911–914	<code>\@printglossary</code> . . . 222, <u>222</u> , 582, 834, 1090, 1245, 1246, 1402, 1403
<code>\@org@Gls@assign@symbolplural</code> 612–615, 617, 621–623, 658, 659, 661, 864, 866–870, 874, 876, 911, 912, 914	<code>\@roman</code> 64, 493, 665, 744, 918, 999, 1154, 1311
<code>\@org@Gls@numberformat</code> 182, 559, 811, 1066, 1221, 1378	<code>\@secondofthree</code> 127, 143, 144, 147, 165, 167, 169, 171, 443, 532, 542–544, 551–553, 555, 643, 783, 794, 796, 803– 806, 896, 1037, 1047–1049, 1057–1060, 1191, 1203, 1205, 1212, 1213, 1215, 1216, 1348, 1360, 1362, 1369, 1370, 1372, 1373
<code>\@org@newglossaryentryprehook</code> 100, 514, 765, 1021, 1176, 1333	<code>\@secondoftwo</code> 28, 31–34, 51, 92, 93, 110, 142–145, 148, 164–166, 168–170, 213, 302, 415, 416, 443–445, 475, 476, 478, 479, 486, 508, 509, 542, 543, 545, 551, 553, 554, 643, 644, 662, 679, 725, 727–729, 736, 759, 760, 793–796, 802, 803, 805, 896–898, 915, 932, 940, 979, 980, 982, 983, 992, 1015, 1016, 1027, 1028, 1047, 1048, 1050, 1056, 1057, 1059, 1086, 1134, 1136–1138, 1147, 1170, 1171, 1182, 1203–1205, 1212–1215, 1241, 1289, 1290, 1292, 1293, 1304, 1327, 1328, 1339, 1360–1362, 1369–1372, 1398
<code>\@outputpage</code> 316, 693, 694, 947	
<code>\@p@glossarysection</code> 57, <u>58</u> , 489, 490, 739, 740, 995, 996, 1149, 1150, 1306, 1307	
<code>\@pgls</code> 420, <u>420</u> , 681, 934	
<code>\@pgls@</code> 420, <u>420</u> , 681, 934	
<code>\@pglsp1</code> 420, <u>420</u> , 681, 934	
<code>\@pglsp1@</code> 420, <u>421</u> , 681, 934	

<code>\@set@glo@numformat</code> . . .	132, 217, 535, 580, 664, 786, 832, 917, 1039, 1087, 1194, 1243, 1351, 1400		741, 742, 815, 997, 998, 1070, 1152, 1153, 1225, 1308, 1310, 1382
<code>\@sglsaddkey</code>	94, 95, 510, 761, 1017, 1172, 1329	<code>\@xdyattributes</code> <u>60</u> , 61, 62, 188, 491, 492, 562, 663, 665, 741, 742, 814, 916, 918, 997, 998, 1069, 1151–1153, 1224, 1308–1310, 1381	
<code>\@sglsaddstoragekey</code>	94, 510, 760, 1016, 1171, 1328	<code>\@xdycounters</code>	60–62, 189, 491, 492, 563, 741, 742, 815, 997, 998, 1070, 1152, 1153, 1225, 1309, 1310, 1382
<code>\@tabacckludge</code>	28, 475, 725, 979, 1134, 1289	<code>\@xdycrossrefhook</code>	<u>65</u> , 189, 494, 563, 745, 815, 1000, 1070, 1155, 1225, 1312, 1382
<code>\@text@composite@x</code>	28, 475, 725, 979, 1134, 1289	<code>\@xdylanguage</code>	<u>67</u> , 225, 496, 584, 746, 836, 1002, 1091, 1157, 1247, 1314, 1404
<code>\@thirdofthree</code> 128, 145, 147, 166, 168, 170, 172, 444, 532, 543, 544, 551, 553–555, 643, 783, 795, 796, 803–805, 807, 897, 1037, 1048, 1049, 1057–1059, 1061, 1192, 1204, 1205, 1213– 1216, 1349, 1361, 1362, 1370–1373		<code>\@xdylettergroups</code> 68, <u>68</u> , 191, 497, 565, 667, 747, 816, 920, 1002, 1003, 1071, 1157, 1227, 1314, 1384	
<code>\@this@attr</code>	189, 563, 564, 815, 1070, 1225, 1226, 1382, 1383	<code>\@xdylocationclassorder</code>	66, <u>66</u> , 189, 495, 563, 666, 745, 815, 919, 1001, 1070, 1156, 1225, 1313, 1382
<code>\@this@childlabel</code>	234, 235, 585, 837, 1092, 1248, 1409	<code>\@xdylocref</code>	<u>60</u> , 61, 191, 491, 492, 564, 663, 667, 741, 742, 816, 916, 920, 997, 998, 1071, 1152, 1226, 1309, 1383
<code>\@this@counter</code>	62, 492, 742, 998, 1153, 1310	<code>\@xdynumbergrouporder</code>	68, 186, <u>186</u> , 497, 561, 562, 747, 813, 1003, 1068, 1157, 1223, 1224, 1314, 1380, 1381
<code>\@this@ctr</code>	189, 563, 815, 1070, 1225, 1382	<code>\@xdyrequiredstyles</code> . 67, <u>67</u> , 187, 496, 562, 664, 746, 814, 918, 1002, 1068, 1156, 1224, 1313, 1381	
<code>\@this@key</code>	93, 509, 760, 1016, 1171, 1328	<code>\@xdysortrules</code>	66, <u>66</u> , 191, 495, 565, 667, 746, 816, 920, 1001, 1071, 1156, 1227, 1313, 1384
<code>\@this@label</code>	234, 585, 836, 837, 1092, 1248, 1409	<code>\@xdystyle</code> . . . 187, 562, 664, 665, 814, 918, 1068, 1069, 1224, 1381	
<code>\@this@scs</code>	49, 485, 486, 736, 991, 1146, 1303	<code>\@xdyuseralphabets</code> . . 63, <u>63</u> , 188, 493, 562, 665, 743, 814, 918, 999, 1069, 1154, 1224, 1311, 1381	
<code>\@tmp</code>	494, 665, 918	<code>\@xdyuserlocationdefs</code> . . . 65, <u>65</u> , 189, 494, 495, 563, 663, 666, 744, 745, 815, 917, 919, 1000, 1001, 1069, 1155, 1225, 1312, 1382	
<code>\@use@ption</code>	44, 484, 734, 990, 1145, 1300	<code>\@xdyuserlocationnames</code> 65, <u>65</u> , 66, 494, 495, 664, 744, 745, 917, 1000, 1001, 1155, 1156, 1312, 1313	
<code>\@warn@nomakeglossaries</code>	200, 225, 572, 584, 823, 836, 1078, 1092, 1234, 1247, 1391, 1404		
<code>\@wrglossary@pageformat</code> . . 212, <u>213</u> , 578, 830, 1085, 1241, 1398			
<code>\@wrglossarynumberhook</code> . . . 212, <u>213</u> , 216, 578, 579, 830, 831, 1085, 1087, 1241, 1242, 1398, 1399			
<code>\@write@glslabelsname</code>	38, <u>38</u> , 731, 986, 1141, 1296		
<code>\@xdy@main@language</code>	35, 36, 480, 570, 571, 584, 730, 822, 823, 836, 984, 1139, 1294		
<code>\@xdyattributelist</code>	<u>60</u> , 62, 189, 491, 492, 563,		

<code>\@xfor@nextelement</code>	<code>\Aclp</code>
. 220, 581, 833, 1089, 1244, 1401	. 284, 611, 863, 1118, 1273, 1439
<code>\@xp@gl@getcounterprefix</code>	<code>\aclp</code>
. 214, 216, 217,	. 284, 611, 863, 1118, 1273, 1439
1086, 1087, 1242, 1243, 1398–1400	<code>\acp</code>
<code>\@xp@glshypernumber</code> 285, 611, 863, 1118, 1273, 1439
. 262, 262, 1104, 1259, 1425	<code>\acrfootnote</code>
<code>\</code> 107, 108,	. 285, 612, 613,
133, 139, 185, 192, 262, 263,	618, 865, 871, 1118, 1274, 1439
519, 535, 561, 565, 566, 598,	<code>\ACRfull</code>
667, 668, 670, 671, 677, 678,	. 269, 600, 852, 1107, 1262, 1428
770, 786, 791, 792, 813, 817,	<code>\Acrfull</code>
850, 920, 921, 923, 924, 930,	. 268, 284, 600, 611, 851, 863,
931, 1026, 1040, 1045, 1067,	1107, 1118, 1262, 1273, 1428, 1439
1071, 1072, 1104, 1105, 1180,	<code>\acrfull</code>
1181, 1195, 1200, 1223, 1227,	. 268, 268, 284,
1228, 1260, 1337, 1338, 1352,	599, 611, 851, 863, 1106, 1107,
1357, 1380, 1384, 1385, 1425, 1426	1118, 1262, 1273, 1427, 1428, 1439
<code>\^</code> 28, 38, 475, 725, 731,	<code>\ACRfullfmt</code>
979, 986, 1134, 1141, 1289, 1296	. 268, 269,
<code>_</code> 207–209, 226–233, 239–243,	271, 280, 281, 456, 458, 600,
248–250, 324, 348, 351–368,	602, 608, 609, 653, 655, 852,
370–392, 395, 398–406, 1405–	853, 860, 861, 906, 908, 1106–
1408, 1413, 1414, 1416, 1418, 1419	1108, 1114, 1116, 1262–1264,
<code>\‘</code> 28, 475, 725, 979, 1134, 1289	1270, 1272, 1427–1429, 1435, 1437
<code>\ </code> 134, 136, 139,	<code>\acrfullfmt</code>
195, 536, 538, 568, 787, 789,	. 267, 268,
791, 819, 1041–1043, 1045,	271, 280, 281, 456, 458, 599,
1074, 1195, 1197, 1198, 1200,	601, 608, 609, 653, 654, 851,
1230, 1352, 1354, 1355, 1357, 1387	853, 859, 861, 906, 908, 1106–
<code>\~</code> 28, 476, 725, 979, 1134, 1289	1108, 1114, 1116, 1262, 1264,
	1270, 1271, 1427–1429, 1435, 1437
	<code>\acrfullformat</code>
	. 181, 267, 267, 559, 599, 600,
	611, 624, 811, 851, 863, 877,
	1065, 1106, 1221, 1262, 1378, 1427
	<code>\ACRfullpl</code>
	. 270, 600, 852, 1107, 1263, 1428
	<code>\Acrfullpl</code>
	. 269, 284, 600, 611, 852, 863,
	1107, 1118, 1263, 1273, 1428, 1439
	<code>\acrfullpl</code>
	. 269,
	269, 284, 600, 611, 852, 863,
	1107, 1118, 1263, 1273, 1428, 1439
	<code>\ACRfullplfmt</code>
	. 268, 270,
	272, 280, 282, 457, 459, 601,
	602, 608, 609, 653, 655, 852–
	854, 860, 861, 906, 908, 1106–
	1108, 1115, 1116, 1262–1264,
	1270, 1272, 1427–1429, 1436, 1437
	<code>\Acrfullplfmt</code>
	. 268, 270,
	272, 280, 282, 457, 459, 600,
<code>\A</code> 259, 1102, 1258, 1423	
<code>\a</code> 28, 476, 725, 980, 1134, 1290	
<code>\AA</code> 28, 476, 725, 980, 1134, 1290	
<code>\aa</code> 28, 476, 725, 980, 1134, 1290	
<code>\Ac</code> 285, 611, 863, 1118, 1273, 1439	
<code>\ac</code> 284, 611, 863, 1118, 1273, 1439	
<code>access (key)</code> 426	
<code>\accsuppglossaryentryfield</code>	
. 451, 628, 649, 881, 902	
<code>\accsuppglossarysubentryfield</code>	
. 451, 628, 649, 881, 902	
<code>\Acf</code> 284, 611, 863, 1118, 1273, 1439	
<code>\acf</code> 284, 611, 863, 1118, 1273, 1439	
<code>\Acfp</code> 284, 611, 863, 1118, 1273, 1439	
<code>\acfp</code> 284, 611, 863, 1118, 1273, 1439	
<code>\Acl</code> 284, 611, 863, 1118, 1273, 1439	
<code>\acl</code> 284, 611, 863, 1118, 1273, 1439	

602, 608, 609, 653, 655, 852,
854, 860, 861, 906, 908, 1106–
1108, 1115, 1116, 1262–1264,
1270, 1272, 1427–1429, 1436, 1437

`\acrfullplfmt` 268, 269,
272, 280, 282, 456, 459, 600,
602, 608, 609, 653, 655, 852,
854, 860, 861, 906, 908, 1106–
1108, 1114, 1116, 1262–1264,
1270, 1272, 1427–1429, 1436, 1437

`\aclinkfootnote` 285,
286, 612, 865, 1118, 1274, 1439

`\aclinkfullformat`
. 267, 267, 268, 599–
601, 851–853, 1106, 1262, 1427

`\ACRlong` 169, 554, 805, 1059, 1215, 1372

`\Acrlong` 169,
169, 284, 553, 611, 805, 863,
1059, 1118, 1214, 1273, 1371, 1439

`\acrlong` 168, 284, 553, 611, 804, 863,
1058, 1118, 1214, 1273, 1371, 1439

`\ACRlongpl`
. 171, 555, 806, 1060, 1216, 1373

`\Acrlongpl` 171,
171, 284, 554, 611, 806, 863,
1060, 1118, 1216, 1273, 1373, 1439

`\acrlongpl`
170, 284, 554, 611, 806, 863,
1060, 1118, 1215, 1273, 1372, 1439

`\acronymformat` 270, 601, 616, 659,
853, 869, 912, 1107, 1263, 1428

`\acrnolinkfootnote`
. 286, 612, 865, 1118, 1274, 1439

acronym (option) 21

`\acronymentry`
271, 272, 274, 275, 277, 278,
280–283, 452–455, 457–461,
601–611, 649–656, 853–863,
903–905, 907–910, 1108–1113,
1115–1117, 1264, 1266–1269,
1271–1273, 1429–1434, 1436–1438

`\acronymfont` 123, 164–168,
175, 181, 270, 270, 272, 274–
278, 280–283, 440, 441, 443–
445, 452–461, 529, 530, 551–
553, 556, 559, 601–611, 613,
614, 616–622, 641–644, 649–
656, 658–660, 780, 781, 802–
804, 808, 811, 853–863, 865,
867, 868, 870–873, 875, 894–
898, 903–911, 913, 1034, 1035,
1056–1058, 1062, 1065, 1107,
1109–1117, 1189, 1190, 1212–
1214, 1218, 1221, 1263, 1264,
1266–1273, 1346, 1347, 1369–
1371, 1375, 1378, 1428, 1430–1438

acronymlists (option) 23

`\acronymname` 21, 22, 49, 52,
471, 472, 486, 487, 721, 722,
736, 737, 976, 991, 993, 1130,
1131, 1146, 1148, 1285, 1303, 1305

acronyms (option) 21

`\acronymsort` 271, 272, 274, 275, 277,
278, 280, 281, 283, 452–455,
457, 458, 460, 461, 601–611,
649–651, 654, 656, 853–863,
903–905, 907–910, 1108–1113,
1115–1117, 1264, 1266–1269,
1271, 1273, 1429–1434, 1436–1438

`\acronymtype` 21, 21, 22,
271, 272, 471, 472, 601, 602,
611–625, 657–660, 721, 722,
853, 854, 863–878, 910–913,
975, 976, 1108, 1109, 1130,
1131, 1263, 1264, 1285, 1429, 1430

`\acrpluralsuffix` 266, 271, 274–
276, 280–282, 452, 453, 457–
460, 599, 601, 603–605, 608–
615, 617–619, 621, 622, 624,
649, 650, 654, 656–661, 851,
853, 855–857, 860–863, 865–
867, 869–872, 874, 875, 877,
878, 903, 907–914, 1106, 1108,
1110–1112, 1115–1117, 1261,
1264, 1266–1268, 1271, 1272,
1427, 1429, 1431–1433, 1436–1438

`\ACRshort`
. 165, 551, 803, 1057, 1212, 1369

`\Acrshort` 165,
165, 283, 551, 611, 803, 863,
1056, 1117, 1212, 1273, 1369, 1438

`\acrshort`
164, 283, 551, 611, 802, 863,
1056, 1117, 1212, 1273, 1369, 1438

`\ACRshortpl`
. 167, 552, 804, 1058, 1214, 1371

`\Acrshortpl` 167,
167, 284, 552, 611, 804, 863,
1058, 1117, 1213, 1273, 1370, 1439

<code>\acrshortpl</code>	314	<code>altlong4col-booktabs (style)</code>	314
166, 284, 552, 611, 803, 863,		<code>altlong4colborder (style)</code>	311
1057, 1117, 1213, 1273, 1370, 1439		<code>altlong4colheader (style)</code>	311
acrstyles:		<code>altlong4colheaderborder (style)</code> .	312
<code>dua</code>	278, 455	<code>altlongragged4col (style)</code>	320
<code>dua-desc</code>	280, 457	<code>altlongragged4col-booktabs</code>	
<code>footnote</code>	281, 458	(style)	315
<code>footnote-desc</code>	282, 460	<code>altlongragged4colborder (style)</code> .	322
<code>footnote-sc</code>	282, 459	<code>altlongragged4colheader (style)</code> .	321
<code>footnote-sc-desc</code>	283, 460	<code>altlongragged4colheaderborder</code>	
<code>footnote-sm</code>	282, 460	(style)	322
<code>footnote-sm-desc</code>	283, 460	<code>\altnewglossary</code>	
<code>long-sc-short</code>	275	. . . 79, 502, 752, 1008, 1163, 1319	
<code>long-sc-short-desc</code>	277, 453	<code>altsuper4col (style)</code>	336
<code>long-short</code>	273, 452	<code>altsuper4colborder (style)</code>	337
<code>long-short-desc</code>	276, 453	<code>altsuper4colheader (style)</code>	336
<code>long-sm-short</code>	276	<code>altsuper4colheaderborder (style)</code>	337
<code>long-sm-short-desc</code>	277, 454	<code>altsuperragged4col (style)</code>	342
<code>long-sp-short</code>	274	<code>altsuperragged4colborder (style)</code>	343
<code>long-sp-short-desc</code>	277	<code>altsuperragged4colheader (style)</code>	343
<code>sc-short-long</code>	276	<code>altsuperragged4colheaderborder</code>	
<code>sc-short-long-desc</code>	278, 454	(style)	343
<code>short-long</code>	275, 452	<code>alttree (style)</code>	412
<code>short-long-desc</code>	278, 454	<code>alttreegroup (style)</code>	414
<code>sm-short-long</code>	276	<code>alttreehypergroup (style)</code>	415
<code>sm-short-long-desc</code>	278, 455	<code>\andname</code> 50, 220, 486, 581, 736, 833,	
<code>\Acs</code> ... 283, 611, 863, 1117, 1273, 1438		992, 1089, 1146, 1245, 1303, 1402	
<code>\acs</code> ... 283, 611, 863, 1117, 1273, 1438		<code>\AnyTrackedLanguages</code>	
<code>\Acsp</code> .. 284, 611, 863, 1117, 1273, 1439	 52, 416, 487, 662,	
<code>\acsp</code> .. 284, 611, 863, 1117, 1273, 1439		679, 737, 915, 932, 993, 1147, 1304	
<code>\add@accent@</code>		<code>\appto</code> 24, 29, 85, 94, 95, 213,	
... 28, 475, 725, 979, 1134, 1289		417, 418, 428, 473, 476, 505,	
<code>\addcontentsline</code>		510, 630, 679, 680, 723, 726,	
... 59, 491, 741, 997, 1151, 1308		755, 761, 883, 932, 933, 977,	
<code>\addglossarytocaptions</code>		980, 1011, 1017, 1086, 1132,	
..... 50, 51, 486, 736,		1135, 1166, 1171, 1172, 1241,	
737, 992, 1146, 1147, 1303, 1304		1286, 1290, 1323, 1328, 1329, 1398	
<code>\addtolength</code>		<code>\AtBeginDocument</code>	21,
... 414, 675, 713, 928, 966, 1450		68, 91, 110, 184, 204, 472, 496,	
<code>\advance</code>	103, 110,	508, 520, 560, 574, 721, 747,	
131, 316, 470, 471, 516, 534,		759, 771, 812, 826, 976, 1002,	
693, 720, 767, 785, 947, 974,		1015, 1027, 1067, 1081, 1130,	
975, 1023, 1027, 1039, 1129,		1157, 1169, 1182, 1222, 1236,	
1178, 1182, 1193, 1335, 1339, 1350		1285, 1314, 1326, 1339, 1379, 1393	
<code>\AE</code> 28, 476, 725, 980, 1134, 1290		<code>\AtEndDocument</code>	37,
<code>\ae</code> 28, 476, 725, 980, 1134, 1290		40, 91, 116, 203, 206, 209, 225,	
<code>altlist (style)</code>	304	298, 481, 482, 508, 523, 573,	
<code>altlistgroup (style)</code>	304	576, 584, 684, 731, 732, 759,	
<code>altlisthypergroup (style)</code>	304	774, 825, 828, 836, 937, 986,	
<code>altlong4col (style)</code>	311	987, 1015, 1031, 1080, 1083,	

	1091, 1121, 1140, 1142, 1169, 1186, 1236, 1239, 1247, 1296, 1297, 1326, 1343, 1393, 1396, 1404		351, 360, 361, 363, 364, 366– 374, 398, 399, 401, 402, 404, 476, 725, 980, 1134, 1288, 1289
automake (option)	39	\c@equation 131, 534, 785, 1039, 1193, 1350
automakegloss (option)	40	\c@glossaryentry 15, 469, 718, 972, 1127, 1282
automakeglosslite (option)	40	\c@glossarysubentry 15, 469, 719, 973, 1128, 1282
B			
\b	28, 476, 725, 980, 1134, 1290	\c@page	212, 213, 216, 578, 579, 830, 831, 1085–1087, 1241, 1242, 1397–1399
\begin	190, 240, 244, 303, 306– 312, 314, 315, 317–322, 324– 344, 564, 588, 666, 687–691, 693–708, 815, 840, 919, 940, 942–962, 1070, 1096, 1226, 1252, 1383, 1413, 1416, 1442–1445	\capitalisewords 178, 809, 1063, 1219, 1376
\BeginAccSupp	425, 628, 881	\catcode	38, 91, 139, 508, 731, 759, 791, 986, 1015, 1045, 1141, 1169, 1200, 1296, 1326, 1357
\begingroup	7, 9, 38, 211, 215, 224, 263, 464, 466, 577, 579, 598, 714, 715, 731, 829, 831, 850, 968, 970, 986, 1084, 1086, 1104, 1123, 1124, 1141, 1240, 1242, 1247, 1260, 1277, 1279, 1296, 1397, 1399, 1404, 1425	\centering	376–378
\bfseries	307, 309–311, 313, 314, 318– 320, 322, 332, 334–337, 339– 344, 689–692, 695–697, 702– 708, 942–946, 948–951, 955–962	\cGls	118, 118 , 524, 775, 1032, 1187, 1344
\bgroup	28, 100, 139, 182, 217, 234, 263, 475, 514, 559, 583, 585, 725, 765, 791, 811, 835, 837, 979, 1021, 1045, 1066, 1087, 1090, 1092, 1105, 1134, 1176, 1200, 1221, 1243, 1248, 1260, 1289, 1333, 1357, 1378, 1400, 1409, 1426	\cglS	117, 117 , 118, 524, 775, 1032, 1186, 1187, 1343, 1344
\bool	27, 30, 70, 74, 141, 229–231, 249, 323, 324, 352, 353, 355, 357, 361–364, 366–372, 385–389, 398–401, 405, 1202, 1359, 1407	\cGlsformat	116, 118 , 523, 524, 774, 775, 1031, 1032, 1186, 1187, 1343, 1344
\boolean	624, 877	\cglSformat	115, 117 , 523, 524, 774, 775, 1031, 1032, 1185, 1187, 1342, 1344
\boolfalse	41, 482, 732, 988, 1142, 1143, 1298	\cGlspl	119, 119 , 524, 525, 775, 1032, 1187, 1344
\booltrue	482, 732	\cglSpl	118, 118 , 119, 524, 775, 1032, 1187, 1344
\bottomrule	313, 314, 692, 945, 946	\cGlsplformat	116, 119 , 523, 525, 774, 776, 1031, 1033, 1186, 1187, 1343, 1344
\box	316, 693, 694, 947	\cglSplformat	116, 119 , 523, 524, 774, 775, 1031, 1032, 1186, 1187, 1343, 1344
C			
\c	27, 28, 208, 228, 348, 349,	\changes	182
		\char	260, 596, 848, 1103, 1258, 1423
		\cleardoublepage	59, 490, 740, 741, 996, 1151, 1308
		\clearpage	59, 490, 740, 741, 996, 1151, 1308
		\closeout	38, 91, 191, 193, 197, 209, 481, 508, 565, 566, 569, 577, 731, 759, 816, 818, 821, 829, 986, 1015, 1071, 1072, 1076, 1084, 1141,

1169, 1227, 1228, 1232, 1239, 1296, 1326, 1384, 1385, 1389, 1396	<code>\cslet</code> 84, 85, 100, 107, 239, 505, 514, 519, 588, 755, 765, 770, 840, 1010, 1011, 1021, 1026, 1095, 1165, 1176, 1180, 1251, 1322, 1333, 1337, 1412
<code>\compatglossarystyle</code> 668–678, 922–932	<code>\csname</code> 16, 19, 28, 44, 49, 51, 52, 58, 61, 62, 64, 65, 67, 69, 73, 79, 80, 87, 88, 94–96, 98, 99, 103– 110, 112, 113, 130–132, 144– 148, 165–172, 182, 184, 188, 189, 194–197, 203, 207, 209, 211–213, 217–219, 222, 223, 225, 235, 261, 265, 289–294, 413, 414, 418, 429, 430, 433, 437, 443–445, 462, 463, 469– 471, 475, 484, 486, 487, 490, 492, 494, 496–498, 501, 502, 506, 510–513, 516–521, 531, 533–535, 542–545, 551–555, 560, 562–564, 567–569, 574, 576–578, 580, 581, 583–585, 591, 594, 595, 597, 599, 625– 628, 630–633, 636, 643, 644, 661–664, 675, 680, 684, 685, 712, 713, 719–721, 725, 734, 736, 737, 740, 742, 744, 746, 747, 749, 751, 752, 756, 757, 760–764, 767–772, 782, 784– 786, 794–796, 802–807, 811, 812, 814, 815, 819–821, 825, 828–830, 832–837, 843, 846– 849, 851, 864, 878–881, 884– 886, 889, 896–898, 914–917, 928, 933, 937, 938, 965, 966, 973–975, 979, 990–993, 996, 998, 1000, 1002–1004, 1007, 1008, 1012, 1013, 1016–1020, 1023–1027, 1029, 1038, 1039, 1048–1050, 1056–1061, 1066, 1067, 1069, 1070, 1074–1076, 1081, 1083–1085, 1087, 1088, 1090, 1091, 1093, 1103, 1105, 1119–1121, 1128–1130, 1134, 1145–1148, 1150, 1151, 1153– 1155, 1157–1159, 1162, 1163, 1167, 1171–1175, 1177–1182, 1184, 1193, 1194, 1203–1205, 1212–1216, 1222, 1224–1226, 1229–1232, 1236, 1239–1241, 1243, 1244, 1246–1248, 1259,
<code>\compatglossarystyle</code> 668–678, 922–932	
compatible-2.07 (option) <u>41</u>	
compatible-3.07 (option) <u>40</u>	
<code>\compatibleglossentry</code> <u>255</u> , <u>425</u> , <u>594</u> , <u>595</u> , <u>628</u> , <u>846</u> , <u>847</u> , <u>881</u>	
<code>\compatiblesubglossentry</code> <u>258</u> , <u>425</u> , <u>595</u> , <u>628</u> , <u>847</u> , <u>881</u>	
<code>\copy</code> 316, 693, 694, 947	
<code>\count@</code> 245, 589, 841, 1096, 1097, 1252, 1416, 1417	
counter (key) <u>83</u>	
counter (option) <u>23</u>	
counters:	
glossaryentry <u>15</u>	
glossarysubentry <u>15</u>	
counterwithin (option) <u>15</u>	
<code>\cs</code> 27, 45–47, 207–209, 226, 227, 229– 233, 240, 241, 243, 248, 249, 296, 297, 324, 346–350, 352– 354, 356–368, 370–372, 374– 392, 395, 398–400, 402, 403, 405, 1288, 1300, 1301, 1404– 1409, 1413, 1416, 1418, 1419	
<code>\csdef</code> 25, 94–96, 106, 107, 114, 115, 234, 235, 261, 273, 474, 510–512, 518, 519, 522, 523, 585, 597, 602, 603, 668, 724, 761, 762, 769, 770, 773, 836, 837, 849, 854, 922, 978, 1017, 1018, 1025, 1026, 1030, 1092, 1093, 1104, 1109, 1133, 1171– 1173, 1180, 1185, 1248, 1259, 1265, 1287, 1288, 1328–1330, 1337, 1342, 1409, 1410, 1425, 1430	
<code>\csedef</code> 117, 212, 523, 578, 774, 830, 1031, 1085, 1186, 1241, 1343, 1398	
<code>\csgdef</code> 20, 56, 77, 81, 115, 116, 221, 248, 489, 501, 502, 522, 523, 582, 591, 739, 751, 752, 773, 774, 834, 843, 975, 995, 1007, 1008, 1030, 1031, 1089, 1098, 1130, 1149, 1161, 1163, 1185, 1186, 1245, 1254, 1284, 1306, 1318, 1320, 1342, 1343, 1402, 1418, 1419	

1261, 1274–1276, 1282, 1284, 1289, 1300, 1303–1305, 1307– 1312, 1314–1316, 1319, 1320, 1324, 1328–1332, 1334–1339, 1341, 1350, 1351, 1360–1362, 1369–1373, 1379, 1381–1383, 1386–1389, 1393, 1396–1398, 1400, 1401, 1403, 1404, 1410, 1424, 1426, 1440–1442, 1449, 1450	<code>\CustomAcronymFields</code>
<code>\csshow</code> 293, 627, 880, 1120, 1276, 1441 288, 624, 877, 878
<code>\csuse</code> 52, 55, 56, 77, 85, 87, 88, 95, 96, 172, 198, 199, 235, 244, 245, 247, 251, 252, 273, 418, 487, 489, 501, 506, 507, 511, 512, 531, 532, 570, 571, 585, 587– 589, 591, 592, 597, 603, 669– 678, 680, 737, 739, 751, 756, 757, 762, 782, 822, 823, 837, 839–841, 843, 844, 849, 855, 923–927, 929–933, 993, 994, 1007, 1010, 1012, 1013, 1018, 1061, 1077, 1078, 1093, 1094, 1096–1099, 1109, 1148, 1149, 1162, 1165, 1167, 1168, 1172, 1173, 1216, 1232–1234, 1248, 1250–1252, 1254, 1255, 1265, 1305, 1306, 1318, 1322, 1324, 1325, 1329, 1330, 1373, 1389– 1391, 1410, 1416–1418, 1420, 1430	<code>\CustomNewAcronymDef</code> 288, 624, 625, 877, 878
<code>\csxdef</code> 105, 116, 517, 523, 768, 774, 864, 1024, 1031, 1179, 1186, 1336, 1343	D
<code>\currentglossary</code> 15, 16, 56, 223, 295, 297, 298, 346, 469, 489, 583, 718, 719, 739, 835, 973, 994, 1091, 1121, 1127, 1128, 1149, 1246, 1282, 1306, 1403	<code>\d</code> 28, 475, 725, 979, 1134, 1289
<code>\currentglssubentry</code> 16, 254, 469, 593, 719, 845, 973, 1100, 1128, 1256, 1282, 1421	<code>\datatool</code> 27, 226, 229–231, 1288, 1289, 1404, 1407
<code>\CurrentOption</code> 44, 417, 424, 484, 627, 679, 734, 880, 932, 990, 1144, 1145, 1300	<code>\datatoolasciistart</code> 229, 1406
<code>\CurrentTrackedLanguage</code> 52, 416, 417, 487, 662, 679, 737, 916, 932, 993, 1147, 1304	<code>\datatoolctrlboundary</code> 229, 231, 1406, 1407
<code>\CurrentTrackedTag</code> 52, 416, 487, 662, 679, 737, 915, 932, 993, 1147, 1304	<code>\day</code> 187, 191, 562, 565, 664, 667, 814, 817, 918, 920, 1068, 1071, 1224, 1227, 1381, 1384
	<code>debug (option)</code> 7
	<code>\DeclareAcronymList</code> 21, 22, 22, 23, 271, 272, 471– 473, 601, 602, 612, 614, 615, 617, 619, 621, 623, 625, 721– 723, 853, 854, 864, 866, 868, 870, 872, 874, 876, 878, 976, 977, 1108, 1109, 1130–1132, 1263, 1264, 1285, 1286, 1429, 1430
	<code>\DeclareCurrentRelease</code> 4, 295, 299, 301, 305, 312, 317, 323, 330, 338, 344, 415–417, 424, 463, 464, 714, 967, 1276, 1442, 1445
	<code>\DeclareListParser</code> 202, 573, 824, 1079, 1235, 1392
	<code>\DeclareOption</code> 12, 417, 424, 467, 627, 679, 717, 880, 932, 971, 1126, 1280
	<code>\DeclareOptionX</code> 12, 467, 717, 971, 1126, 1280
	<code>\DeclareRelease</code> 4, 295, 299, 301, 305, 312, 317, 323, 330, 338, 344, 415–417, 424, 463, 464, 714, 967, 1276, 1442, 1445
	<code>\DeclareRobustCommand</code> 53, 487, 620, 738, 872, 993, 1148, 1305
	<code>\def</code> 11, 13, 15–20, 22, 27–29, 35, 36, 38–40, 42, 47– 49, 52, 53, 56, 60, 63–68, 73, 78–86, 89, 98, 100, 102–107, 110, 111, 115–120, 126, 130– 140, 143–172, 182–184, 187, 191, 193–200, 203–205, 209, 212, 214, 216–219, 222, 223,

234, 238, 244, 245, 247, 250,
252, 259–265, 268–271, 298–
301, 329, 330, 412–415, 417,
420–423, 426–429, 443–450,
467–472, 475, 476, 480, 482–
489, 491, 493–499, 501–505,
507, 513–520, 523–525, 531–
555, 560, 562, 565–572, 574–
576, 578–581, 583, 584, 588–
592, 596–601, 612–617, 619–
621, 623, 625, 628–630, 643–
648, 657–661, 664, 667, 669,
674, 675, 679, 681–686, 700,
701, 712, 713, 717–722, 724–
726, 730–733, 735–739, 741,
743–749, 751–755, 757, 763,
765–771, 773–776, 782–807,
811, 812, 814, 816, 818–823,
826–828, 830–836, 840–844,
848–853, 864, 866–874, 876,
878, 881–884, 896–901, 911–
914, 917, 918, 920, 922, 928,
932–936, 938, 939, 953, 954,
965–967, 971–977, 979, 980,
984, 986–991, 993–995, 997,
999–1004, 1007–1011, 1013,
1020–1028, 1030–1033, 1036,
1038–1061, 1066–1068, 1071,
1073–1083, 1085–1090, 1092,
1095, 1096, 1098–1100, 1102–
1108, 1121, 1126–1131, 1133–
1135, 1139, 1140, 1142, 1143,
1145, 1146, 1148, 1149, 1152,
1154–1157, 1159, 1162–1166,
1168, 1174, 1176–1180, 1182,
1183, 1185–1188, 1191, 1193–
1201, 1203–1216, 1221, 1222,
1224, 1227–1234, 1236, 1237,
1239, 1241–1244, 1246, 1247,
1251–1255, 1258–1263, 1280–
1286, 1288–1290, 1294–1299,
1302, 1303, 1305, 1306, 1309,
1311–1314, 1316, 1319–1323,
1325, 1331, 1333–1337, 1339,
1340, 1342–1345, 1348, 1350–
1358, 1360–1373, 1378, 1379,
1381, 1384–1391, 1393, 1394,
1396, 1398–1401, 1403, 1409,
1412, 1416–1421, 1423–1426,
1428, 1429, 1444, 1445, 1449, 1450

`\def@glsexdycheckbackslash`
. 140, 540,
541, 792, 793, 1046, 1201, 1358
`\DefaultNewAcronymDef` 285,
461, 611, 612, 657, 863, 865, 910
`\defglssdisplay` 125, 531, 782
`\defglssdisplayfirst` . . . 125, 531, 782
`\defglssentryfmt`
. . . 77, 79, 81, 273, 285, 501,
502, 531, 602, 611, 612, 614,
616, 618, 620, 622, 624, 751–
753, 782, 854, 863, 865, 867,
868, 870, 873, 875, 877, 1007,
1008, 1109, 1118, 1161–1163,
1265, 1274, 1318–1320, 1430, 1439
`\define@boolkey` . . 10, 12–16, 21,
29, 30, 34, 35, 40, 41, 126, 127,
252, 467–469, 471, 476, 477,
479, 480, 482, 532, 592, 716–
719, 721, 726, 729, 730, 732,
783, 844, 971–973, 976, 980,
981, 983, 984, 987, 988, 1036,
1099, 1125–1128, 1130, 1135,
1138, 1139, 1142, 1191, 1255,
1280–1282, 1285, 1290, 1291,
1293, 1294, 1297, 1298, 1348, 1420
`\define@choicekey`
. . . 7, 9–11, 16, 30–32, 35, 39,
42, 44, 84, 251, 252, 465–467,
469, 477, 478, 480, 481, 483,
504, 591, 592, 714, 716, 717,
719, 727–729, 732, 733, 754,
843, 844, 968, 970, 971, 973,
981, 982, 984, 986, 988, 989,
1010, 1099, 1123, 1125, 1128,
1136, 1137, 1139, 1141, 1143,
1144, 1165, 1254, 1255, 1277,
1279, 1280, 1282, 1291, 1292,
1294, 1296, 1298, 1300, 1322, 1420
`\define@key` 11, 12, 15, 23, 24,
31, 33, 36, 81–86, 94, 95, 126,
183, 250, 252, 417, 426–428,
467, 469, 473, 477, 480, 503–
505, 510, 532, 560, 591, 592,
628, 629, 679, 717, 719, 723,
727, 730, 753–755, 761, 783,
812, 843, 844, 882, 883, 932,
971, 973, 977, 981, 983, 984,
1009–1011, 1017, 1036, 1067,
1098–1100, 1126, 1127, 1132,

1136, 1138, 1139, 1163–1166, 1171, 1172, 1191, 1222, 1254, 1255, 1280, 1282, 1286, 1287, 1291, 1293–1295, 1320–1323, 1328, 1329, 1348, 1379, 1419, 1421	<code>\do</code> 33, 44, 49, 51, 60, 62, 69, 92, 93, 133, 182, 187– 189, 200–202, 212, 219, 234, 272, 479, 484, 485, 491, 492, 497, 508, 509, 535, 560, 562, 563, 572, 573, 578, 581, 585, 602, 612, 614, 616, 617, 619, 622, 623, 625, 664, 684, 685, 729, 734, 736, 742, 747, 759, 760, 786, 811, 814, 815, 824, 825, 830, 833, 836, 837, 854, 864, 867, 868, 870, 872, 875, 876, 878, 918, 937, 938, 983, 990–992, 997, 998, 1003, 1015, 1016, 1040, 1066, 1068–1070, 1078–1080, 1085, 1089, 1092, 1109, 1121, 1138, 1144, 1146, 1147, 1152, 1153, 1157, 1158, 1170, 1171, 1195, 1222, 1224, 1225, 1234, 1235, 1240, 1244, 1248, 1265, 1293, 1300, 1303, 1309, 1310, 1314, 1315, 1327, 1328, 1352, 1379, 1381, 1382, 1391, 1392, 1397, 1401, 1409, 1430
<code>\DefineAcronymSynonyms</code> 47, <u>283</u> , 285, 485, 611, 735, 863, 990, 1117, 1118, 1145, 1273, 1302, 1438, 1439	<code>\do@glo@storeentry</code> 17–20, 106, 469–471, 518, 719–721, 769, 973–975, 1025, 1128–1130, 1180, 1283, 1284, 1337
<code>\delimN</code> <u>55</u> , 190, 202, 247, 263, 489, 564, 573, 590, 598, 666, 739, 816, 824, 842, 850, 920, 994, 1070, 1080, 1098, 1104, 1149, 1226, 1235, 1253, 1260, 1306, 1383, 1392, 1418, 1425	<code>\do@gls@link@checkfirsthyper</code> 128, 130, 143–149, 164–172, 443– 445, 533, 542–545, 551–555, 643, 644, 783, 784, 794–797, 802–806, 896, 897, 1037, 1038, 1047–1050, 1056–1061, 1192, 1193, 1203–1206, 1212–1216, 1349, 1350, 1360–1363, 1369–1373
<code>\delimR</code> <u>55</u> , 190, 262, 263, 489, 564, 598, 666, 739, 816, 850, 920, 994, 1070, 1104, 1149, 1226, 1260, 1306, 1383, 1425	<code>\do@gls@xdycheckbackslash</code> 133, 535, 786, 1040, 1195, 1352
<code>description (key)</code> <u>81</u>	<code>\do@glsdisablehyperinlist</code> <u>129</u> , 130, 533, 784, 1038, 1192, 1193, 1349, 1350
<code>description (option)</code> <u>34</u>	<code>\do@gls@haschildren</code> 73, 498, 499, 748, 749, 1004, 1159, 1316
<code>descriptionaccess (key)</code> <u>426</u>	<code>\do@write@glslabelsname</code> 38, <u>38</u> , 731, 986, 1140, 1141, 1296
<code>\DescriptionDUANewAcronymDef</code> <u>286</u> , 614, 616, 867, 868	<code>\doifglossarynoexistsordo</code> <u>72</u> , 78, 498, 501, 748, 751, 1004, 1007, 1159, 1162, 1316, 1319
<code>\DescriptionFootnoteNewAcronymDef</code> <u>286</u> , <u>461</u> , 613, 614, 658, 865, 866, 911	<code>\dtl@ifsingle</code> 596, 848
<code>\descriptionname</code> <u>49</u> , 52, 307, 309–311, 313, 314, 318–320, 322, 332, 334–337, 339–344, 486, 487, 689–692, 695–697, 702–704, 706–708, 736, 737, 942–946, 948–951, 955–962, 991, 993, 1146, 1148, 1303, 1305	
<code>\DescriptionNewAcronymDef</code> <u>286</u> , <u>461</u> , 616, 617, 658, 869, 870, 912	
<code>descriptionplural (key)</code> <u>82</u>	
<code>descriptionpluralaccess (key)</code> <u>427</u>	
<code>\DH</code> 29, 476, 726, 980, 1135, 1290	
<code>\dh</code> 29, 476, 726, 980, 1135, 1290	
<code>\dim</code> 226, 348, 352–356, 360–373, 399, 401–405	
<code>\dimen@</code> 275, 316, 411, 412, 604, 693, 711, 712, 856, 947, 965, 1111, 1266, 1432, 1448, 1449	
<code>\directlua</code> 45, 990, 1145, 1300	
<code>\disable@keys</code> 47, 484, 734, 990, 1145, 1302	
<code>disablemakegloss (option)</code> <u>36</u>	

<code>\dtl@insertinto</code>	759, 760, 782, 785–793, 812,
. 235, 586, 838, 1093, 1249, 1410	813, 816, 818–820, 822–824,
<code>\dtl@sortresult</code>	828, 831, 832, 836, 838, 841,
838, 1093, 1094, 1249, 1410, 1411	848, 850, 851, 863, 865, 867,
<code>\DTLassignlettergroup</code> ...	869, 871, 873, 875, 910–913,
<code>\dtlcompare</code>	916–918, 920, 991, 992, 997–
. 236, 586, 838, 1094, 1249, 1411	1002, 1007, 1012, 1013, 1015,
<code>\dtlicompare</code>	1016, 1038–1046, 1067, 1071,
. 236, 586, 838, 1094, 1249, 1411	1073–1079, 1083, 1091, 1093,
<code>\DTLifinlist</code>	1094, 1097, 1103, 1105, 1146,
. 81, 129, 503, 533, 753, 784,	1147, 1151–1156, 1162, 1167,
1009, 1038, 1163, 1192, 1320, 1349	1169, 1171, 1193–1201, 1223,
<code>\DTLifint</code>	1227–1235, 1239, 1247, 1249,
. 259, 596, 848, 1103, 1258, 1423	1252, 1258, 1260, 1261, 1303,
<code>\dtlletterindexcompare</code>	1304, 1308–1313, 1319, 1324,
. 236, 586, 838, 1093, 1249, 1410	1326, 1328, 1350–1358, 1380,
<code>\dtlnonlettergroup</code>	1384–1388, 1392, 1396, 1404,
. 231, 1407	1410, 1411, 1417, 1423, 1426
<code>\dtlnumbergroup</code>	<code>\egroup</code>
. 231, 1407	28,
<code>\DTLsortedletter</code>	101, 183, 218, 235, 264, 475,
. 232, 1408	514, 560, 583, 585, 725, 765,
<code>\DTLsortlettercasehandler</code>	812, 835, 837, 979, 1021, 1066,
. 230	1088, 1091, 1092, 1105, 1134,
<code>\DTLsortletterhandler</code> ...	1176, 1222, 1243, 1248, 1260,
. 230, 1407	1289, 1333, 1379, 1400, 1410, 1426
<code>\DTLsortwordcasehandler</code>	<code>\else</code>
. 230, 231, 1407	8, 14–
<code>\DTLsortwordhandler</code>	16, 20–24, 26, 29–31, 38, 40,
. 229, 231, 1406, 1407	44, 47–49, 53, 54, 57, 59–69,
<code>\DTLsubstituteall</code>	77, 84, 87, 103, 104, 107–110,
. 133, 536, 786, 1040, 1195, 1352	115, 116, 127, 129–140, 175,
<code>\dtlwordindexcompare</code>	186, 187, 190–200, 209, 211,
. 236, 586, 838, 1093, 1249, 1410	213, 214, 216–220, 223, 245,
<code>dua (acrstyle)</code>	252, 254, 260, 262–264, 275,
. 278, 455	287, 289, 303, 307, 308, 310,
<code>dua (option)</code>	313, 314, 316, 318, 319, 321,
. 34	331, 333, 335, 339, 341, 343,
<code>dua-desc (acrstyle)</code>	408, 409, 411, 413, 414, 434,
. 280, 457	465, 466, 468, 469, 471–474,
<code>\DUANewAcronymDef</code>	476, 477, 481–485, 488–497,
. 287, 622, 623, 875, 876	504, 506, 516, 517, 519, 520,
	523, 533–545, 556, 561, 562,
	564–571, 576–581, 583, 589,
	592–594, 596, 598, 604, 614,
	615, 617, 619, 620, 622–625,
	633, 664–669, 673–675, 684,
	687, 689, 690, 692–696, 701–
	703, 705–707, 709–713, 715,
	718–724, 726, 727, 731–736,
	738–747, 754, 756, 767, 768,
	770, 771, 774, 784–796, 807,
E	
<code>\eappto</code>	502, 503, 519, 578
<code>\edef</code> ...	49, 51, 60–65, 67, 79, 87,
88, 91, 93, 131–140, 185, 186,	
191, 193–195, 202, 209, 225,	
236, 245, 260, 263–265, 471,	
472, 485, 487, 491–498, 501–	
503, 506–509, 512–515, 519,	
531, 533–541, 559–561, 565–	
568, 570, 571, 573, 576, 579,	
580, 582, 584–586, 589, 593,	
596, 598, 599, 611, 613, 614,	
616, 618, 620, 622, 657–660,	
663–665, 667, 684, 736, 737,	
741–746, 751, 752, 756, 757,	

808, 813, 814, 816, 818–823,
828–833, 835, 841, 844–846,
848, 850, 856, 866–868, 870,
872–875, 877, 878, 887, 917,
918, 920–922, 926–928, 937,
940, 942–950, 955–957, 959–
961, 963–967, 969, 972–978,
980, 981, 986, 987, 990, 991,
994–1003, 1007, 1010, 1012,
1023, 1024, 1026, 1027, 1031,
1036–1046, 1062, 1068, 1069,
1071–1073, 1075–1079, 1084–
1089, 1091, 1096, 1100, 1101,
1103–1105, 1111, 1118, 1121,
1124, 1127–1133, 1135, 1136,
1141, 1142, 1144–1146, 1148–
1158, 1161, 1165, 1167, 1177,
1178, 1180–1182, 1185, 1186,
1191–1201, 1218, 1223, 1224,
1226–1234, 1239–1244, 1246,
1252, 1255–1257, 1259–1261,
1266, 1274, 1278, 1279, 1281,
1282, 1285–1288, 1290, 1291,
1296–1298, 1300, 1302, 1303,
1305–1315, 1318, 1322, 1324,
1334, 1335, 1337–1339, 1342,
1343, 1348–1358, 1375, 1380,
1381, 1383–1391, 1396–1401,
1403, 1417, 1420–1422, 1424–
1426, 1432, 1439, 1440, 1446–1450
\emph . 219, 265, 581, 599, 833, 851,
1089, 1105, 1244, 1261, 1401, 1426
\empty 263, 424, 598,
627, 850, 881, 1105, 1260, 1426
\encodingdefault
. . . 28, 475, 725, 979, 1134, 1289
\end . . 190, 240, 241, 244, 303, 306–
312, 314, 315, 317–322, 325–
344, 564, 589, 666, 687–691,
693–708, 815, 840, 919, 940,
942–962, 1070, 1096, 1226,
1252, 1383, 1413, 1416, 1442–1445
\end@doifinlist
. . . 60, 491, 741, 997, 1152, 1309
\end@getprefix
. 218, 580, 832, 1088, 1243, 1400
\end@glis@islistofacronyms . . 22,
472, 473, 722, 976, 977, 1131, 1286
\EndAccSupp 425, 628, 881
\endcsname 16,
19, 28, 44, 49, 51, 52, 58, 61,
62, 64, 65, 67, 69, 73, 79, 80,
87, 88, 94–96, 98, 99, 103–
110, 112, 113, 130–132, 144–
148, 165–172, 182, 184, 188,
189, 194–197, 203, 207, 209,
211–213, 217–219, 222, 223,
225, 235, 261, 265, 289–294,
413, 414, 418, 429, 430, 433,
437, 443–445, 462, 463, 469–
471, 475, 484, 486, 487, 490,
492, 494, 496–498, 501, 502,
506, 510–513, 516–521, 531,
533–535, 542–545, 551–555,
560, 562–564, 567–569, 574,
576–578, 580, 581, 583–585,
591, 594, 595, 597, 599, 625–
628, 630–633, 636, 643, 644,
661–664, 675, 680, 684, 685,
712, 713, 719–721, 725, 734,
736, 737, 740, 742, 744, 746,
747, 749, 751, 752, 756, 757,
760–764, 767–772, 782, 784–
786, 794–796, 802–807, 811,
812, 814, 815, 819–821, 825,
828–830, 832–837, 843, 846–
849, 851, 864, 878–881, 884–
886, 889, 896–898, 914–917,
928, 933, 937, 938, 965, 966,
973–975, 979, 990–993, 996,
998, 1000, 1002–1004, 1007,
1008, 1012, 1013, 1016–1020,
1023–1027, 1029, 1038, 1039,
1048–1050, 1056–1061, 1066,
1067, 1069, 1070, 1074–1076,
1081, 1083–1085, 1087, 1088,
1090, 1091, 1093, 1103, 1105,
1119–1121, 1128–1130, 1134,
1145–1148, 1150, 1151, 1153–
1155, 1157–1159, 1162, 1163,
1167, 1171–1175, 1177–1182,
1184, 1193, 1194, 1203–1205,
1212–1216, 1222, 1224–1226,
1229–1232, 1236, 1239–1241,
1243, 1244, 1246–1248, 1259,
1261, 1274–1276, 1282, 1284,
1289, 1300, 1303–1305, 1307–
1312, 1314–1316, 1319, 1320,
1324, 1328–1332, 1334–1339,
1341, 1350, 1351, 1360–1362,

1369–1373, 1379, 1381–1383,
 1386–1389, 1393, 1396–1398,
 1400, 1401, 1403, 1404, 1410,
 1424, 1426, 1440–1442, 1449, 1450
`\endfoot` 307–311,
 313, 314, 318–320, 322, 689–
 692, 695–697, 942–946, 948–951
`\endgroup` 7, 9, 38, 211,
 216, 224, 263, 464, 466, 577,
 579, 598, 714, 716, 731, 829,
 831, 850, 968, 970, 986, 1084,
 1087, 1105, 1123, 1124, 1141,
 1240, 1242, 1247, 1260, 1277,
 1279, 1296, 1397, 1399, 1404, 1426
`\endhead` 307–311,
 313, 314, 318–320, 322, 689–
 692, 695–697, 942–946, 948–951
`\endtheglossary`
 . . 6, 7, 464, 714, 968, 1123, 1277
`entrycounter` (option) 15
`\entryname` 49, 52, 307,
 309–311, 313, 314, 318–320,
 322, 332, 334–337, 339–344,
 486, 487, 689–692, 695–697,
 702–704, 706–708, 736, 737,
 942–946, 948–951, 955–962,
 991, 993, 1146, 1148, 1303, 1305
 environments:
 `theglossary` 255
`\epreto` 110, 1027, 1182, 1339
`\equal` . . 31, 48, 59, 131, 201, 477,
 485, 490, 534, 572, 596, 684,
 727, 735, 740, 785, 824, 848,
 937, 981, 991, 996, 1038, 1079,
 1121, 1136, 1145, 1151, 1193,
 1235, 1291, 1302, 1308, 1350, 1392
`esclocations` (option) 13
`\everypar` 399, 401, 405
`\exp` 18, 19, 27, 47, 207, 226–
 232, 242, 243, 249, 298, 324,
 325, 345, 348, 351–354, 359,
 361–365, 371–373, 398–400,
 404, 405, 1283, 1284, 1288,
 1289, 1301, 1405–1408, 1414–1416
`\expandafter` . . . 9, 19, 27, 28, 44,
 49, 51, 61, 62, 64–69, 73, 79–
 81, 87, 88, 92–96, 98, 99, 103–
 113, 125, 129, 131–140, 173–
 175, 178, 183, 185, 186, 189,
 193–195, 197, 209, 211, 212,
 214, 216, 217, 220, 223, 224,
 233, 235, 245, 256, 261, 263,
 265, 289–294, 400, 406, 418,
 429, 430, 433, 434, 461–463,
 466, 470, 471, 475, 484–487,
 492, 494–498, 501–503, 506–
 513, 516–521, 531, 533–541,
 556, 560, 561, 563, 564, 566–
 569, 576–581, 583, 585, 589,
 590, 594, 595, 597–599, 613,
 618, 625–628, 630–634, 657,
 661–664, 666, 667, 680, 684,
 685, 712, 715, 719–721, 724,
 725, 734, 736, 737, 742, 744,
 746, 747, 749, 751–753, 756,
 757, 759–764, 767–772, 782,
 784–792, 807–809, 812, 813,
 815, 818–821, 828–833, 835,
 837, 841, 842, 846, 847, 849–
 851, 864, 865, 871, 878–881,
 884–887, 910, 914–917, 919,
 920, 933, 937, 938, 965, 970,
 974, 975, 979, 990–992, 998,
 1000–1004, 1007–1009, 1012,
 1013, 1015–1020, 1023–1029,
 1036, 1038–1046, 1061, 1062,
 1064, 1066, 1067, 1070, 1073–
 1076, 1083–1087, 1089, 1091,
 1093, 1096, 1097, 1101, 1103,
 1105, 1119–1122, 1124, 1128–
 1130, 1133, 1134, 1145–1147,
 1152–1159, 1162, 1163, 1167,
 1168, 1170–1175, 1177–1184,
 1190–1201, 1217–1219, 1222,
 1223, 1225, 1226, 1229–1231,
 1239–1243, 1245–1249, 1252,
 1253, 1257, 1259–1261, 1274–
 1276, 1279, 1284, 1288, 1289,
 1300, 1303, 1304, 1309–1316,
 1319, 1320, 1324, 1325, 1327–
 1332, 1334–1341, 1347–1358,
 1374–1376, 1379, 1380, 1382,
 1383, 1386–1388, 1396–1401,
 1403, 1404, 1409, 1410, 1416,
 1422, 1424, 1426, 1440–1442, 1449
`\expandonce` 87–
 89, 99, 133, 134, 194–196, 212,
 213, 236, 271, 506, 507, 535,
 536, 567–569, 578, 586, 594,
 595, 601, 611, 613, 615, 617,

618, 621, 622, 628, 756, 757,
764, 786, 787, 819, 820, 830,
838, 846, 847, 853, 863–867,
869, 871, 874, 875, 881, 1012,
1013, 1020, 1040, 1041, 1074,
1075, 1085, 1093, 1094, 1108,
1167, 1168, 1175, 1195, 1196,
1229–1231, 1241, 1249, 1264,
1324, 1325, 1332, 1352, 1353,
1386–1388, 1398, 1410, 1411, 1429

`\ExplSyntaxOff` 6,
12, 19, 27, 30, 47, 70, 74, 110,
141, 209, 234, 244, 250, 260,
299, 325, 406, 412, 968, 1102,
1103, 1122, 1182, 1202, 1258,
1259, 1277, 1284, 1289, 1302,
1339, 1359, 1409, 1416, 1419, 1424

`\ExplSyntaxOn` 4, 11,
16, 27, 30, 45, 70, 74, 110, 141,
207, 226, 239, 248, 259, 260,
296, 323, 344, 412, 967, 1102,
1103, 1122, 1182, 1202, 1258,
1276, 1283, 1288, 1300, 1339,
1359, 1404, 1413, 1418, 1423, 1424

F

`\fi` 7, 8, 10, 11,
14–17, 20–24, 26, 29–31, 33–
36, 38–40, 42–44, 47–49, 52–
54, 57, 59–69, 77, 80, 84, 87,
99, 103–110, 115, 116, 127,
129–140, 175, 184, 186, 187,
190, 192–198, 200–202, 209–
214, 216–221, 223–225, 245,
251–254, 260–264, 275, 285,
287, 289, 303, 307, 308, 310,
313, 314, 316, 318, 320, 321,
331, 333, 335, 339, 341, 343,
400, 406, 408, 409, 411–414,
425, 434, 464–477, 479, 481–
485, 487–497, 502, 504, 506,
516–520, 523, 533–545, 556,
560–562, 564, 566–570, 572,
573, 577–584, 589, 592–594,
596–599, 604, 611, 612, 614,
615, 617–625, 627, 628, 634,
663–669, 673–675, 684, 687,
689, 690, 692–696, 701–703,
705–707, 709–724, 726–728,
731–736, 738–747, 752, 754,
756, 764, 766–771, 774, 784–
797, 808, 812–814, 816, 818–
825, 828–836, 841, 844–846,
848–850, 856, 863, 864, 866–
868, 870, 872–878, 880, 881,
887, 916–918, 920–922, 926–
928, 937, 940, 942–950, 955–
957, 959–961, 963–978, 980,
981, 983–991, 993–1003, 1007,
1008, 1010, 1012, 1020, 1022–
1027, 1030, 1031, 1036–1046,
1062, 1067–1069, 1071–1073,
1075–1080, 1084–1092, 1097,
1099–1101, 1103–1105, 1111,
1118, 1119, 1121, 1123–1133,
1135, 1136, 1138, 1139, 1141–
1146, 1148–1158, 1161, 1163,
1165, 1167, 1175, 1177–1179,
1181, 1182, 1185, 1186, 1191–
1201, 1218, 1222–1224, 1226,
1228, 1229, 1231, 1233–1235,
1239–1247, 1252, 1255–1257,
1259–1261, 1266, 1273, 1274,
1277–1283, 1285–1288, 1290,
1291, 1293–1300, 1302, 1303,
1305–1315, 1318, 1320, 1322,
1324, 1332, 1334–1336, 1338,
1339, 1342, 1343, 1348–1358,
1375, 1379–1381, 1383, 1385,
1386, 1388, 1390–1392, 1396–
1404, 1417, 1420–1422, 1424–
1426, 1432, 1439, 1440, 1446–1450

`\findrootlanguage` 67, 496, 746

`first` (key) 82

`firstaccess` (key) 426

`\firstacronymfont`
. . . . 125, 270, 274, 275, 281,
442, 443, 452, 453, 458, 531,
601, 603, 604, 609, 613, 616,
618, 620, 643, 649, 650, 654,
781, 782, 853, 855, 856, 861,
865, 869, 871, 873, 896, 902,
903, 907, 1036, 1107, 1110,
1111, 1115, 1116, 1190, 1191,
1263, 1265–1267, 1271, 1347,
1348, 1428, 1431, 1432, 1436, 1437

`firstplural` (key) 82

`firstpluralaccess` (key) 426

`\fmtversion`
. . . . 29, 476, 726, 980, 1135, 1290

<code>\footnote</code>	281, 286, 458, 609, 612, 654, 861, 865, 907, 1115, 1116, 1118, 1271, 1274, 1436, 1437, 1439	492, 501, 512, 516, 577, 592, 593, 684, 720, 731, 742, 751, 763, 767, 829, 844, 937, 974, 986, 998, 1007, 1019, 1023, 1084, 1100, 1121, 1129, 1141, 1153, 1162, 1174, 1177, 1178, 1240, 1255, 1256, 1296, 1310, 1319, 1331, 1334, 1335, 1397, 1421
<code>footnote (acrstyle)</code>	281, 458	
<code>footnote (option)</code>	34	
<code>footnote-desc (acrstyle)</code>	282, 460	
<code>footnote-sc (acrstyle)</code>	282, 459	
<code>footnote-sc-desc (acrstyle)</code>	283, 460	
<code>footnote-sm (acrstyle)</code>	282, 460	
<code>footnote-sm-desc (acrstyle)</code>	283, 460	
<code>\FootnoteNewAcronymDef</code>	287, 461, 618, 619, 659, 871, 872, 913	
<code>\footnotesize</code>	8, 466, 715, 969, 1124, 1279	
<code>\forallallacronyms</code>	69, 497, 747, 1003, 1158, 1314	
<code>\forallallglossaries</code>	69, 69, 209, 222, 411, 497, 576, 582, 711, 747, 828, 834, 965, 1003, 1083, 1090, 1157, 1158, 1239, 1245, 1246, 1314, 1315, 1396, 1402, 1403, 1448	
<code>\forallallglsentries</code>	37, 69, 113, 114, 117, 184, 185, 481, 497, 521, 522, 524, 561, 731, 747, 772–774, 812, 986, 1003, 1029–1031, 1067, 1140, 1158, 1184, 1186, 1223, 1296, 1315, 1341, 1343, 1380	
<code>\ForEachTrackedDialect</code>	52, 416, 487, 662, 679, 737, 915, 932, 993, 1147, 1304	
<code>\forallrglsentries</code>	69, 69, 73, 107, 238, 412, 497, 498, 519, 588, 711, 747, 749, 770, 840, 965, 1003, 1004, 1026, 1095, 1158, 1159, 1180, 1251, 1315, 1316, 1337, 1412, 1448	
<code>\forallrlistcsloop</code>	227, 234, 240, 244, 248, 585, 589, 836, 840, 1092, 1096, 1248, 1252, 1405, 1409, 1413, 1416, 1419	
<code>\forallrlistloop</code>	205, 247, 575, 590, 827, 842, 1082, 1098, 1237, 1238, 1253, 1394, 1395, 1418	
G		
<code>\g</code>	207, 208, 249, 250	
<code>\gdef</code>	38, 62, 78, 79, 98, 103, 104, 211, 252, 253, 298, 470,	
<code>\Genacrfullformat</code>	124, 125, 271, 272, 274, 275, 281, 442, 442, 452, 453, 458, 530, 531, 602–604, 609, 642, 643, 649, 650, 654, 781, 853–856, 861, 895, 896, 902, 903, 907, 1035, 1036, 1108, 1110, 1111, 1115, 1190, 1264– 1267, 1271, 1347, 1429–1432, 1436	
<code>\genacrfullformat</code>	124, 124, 125, 271, 272, 274, 275, 281, 442, 442, 452, 458, 530, 531, 601–604, 609, 642, 649, 650, 654, 781, 782, 853–856, 861, 895, 896, 902, 903, 907, 1035, 1036, 1108, 1110, 1111, 1115, 1190, 1264– 1266, 1271, 1347, 1429–1432, 1436	
<code>\GenericAcronymFields</code>	271, 272, 274, 275, 277, 278, 280, 281, 283, 452–458, 460, 461, 601– 611, 649–651, 653, 654, 656, 853–862, 902–907, 909, 910, 1108–1115, 1117, 1264–1266, 1268–1271, 1273, 1429–1436, 1438	
<code>\Genplacrfullformat</code>	124, 125, 272, 274, 275, 281, 442, 443, 452, 453, 458, 530, 531, 602–604, 609, 642, 643, 649, 650, 654, 781, 782, 854– 856, 861, 895, 896, 902, 903, 907, 1035, 1036, 1108–1111, 1116, 1190, 1191, 1264–1267, 1271, 1347, 1348, 1429–1432, 1437	
<code>\genplacrfullformat</code>	124, 125, 125, 272, 274, 275, 281, 441, 442, 443, 452, 453, 458, 530, 531, 602–604, 609, 642, 643, 649, 650, 654, 781, 782, 854– 856, 861, 895, 896, 902, 903, 907, 1035, 1036, 1108–1111, 1116, 1190, 1191, 1264–1267,	

1271, 1347, 1348, 1429–1432, 1437
 \GetTitleStringDisableCommands
 302, 940
 \GetTitleStringSetup 302, 940
 \glo@desc 669, 922
 \glo@do@compare 236, 237, 586,
 838, 1093, 1094, 1249, 1410, 1411
 \glo@grabfirst
 233, 244, 589, 590, 841, 842,
 1096, 1097, 1252, 1253, 1409, 1416
 \glo@label
 . 73, 107, 498, 519, 749, 770,
 1004, 1026, 1159, 1180, 1316, 1337
 \glo@list
 . 107, 519, 770, 1026, 1180, 1337
 \glo@name
 . 256, 594, 846, 1101, 1257, 1422
 \glo@parent
 . . 73, 498, 749, 1004, 1159, 1316
 \glo@type
 . 107, 519, 770, 1026, 1180, 1337
 \glo@value
 . . 92, 508, 759, 1015, 1170, 1327
 \global 19, 87, 90, 100,
 106, 113, 211, 224, 245, 246,
 253, 316, 470, 471, 506, 508,
 514, 518, 521, 577, 583, 589,
 590, 593, 694, 720, 721, 756,
 758, 765, 769, 772, 829, 835,
 841, 842, 845, 947, 974, 975,
 1012, 1014, 1021, 1025, 1029,
 1084, 1091, 1097, 1100, 1129,
 1130, 1167, 1169, 1176, 1180,
 1184, 1240, 1247, 1252, 1253,
 1256, 1284, 1324, 1326, 1333,
 1337, 1341, 1397, 1404, 1417, 1421
 \glo@linkprefix 130, 131,
 183, 255, 534, 560, 594, 785,
 812, 846, 1038, 1067, 1101,
 1193, 1222, 1257, 1350, 1379, 1422
 \glosortentrieswarning
 24, 24, 234,
 473, 474, 584, 723, 836, 977,
 978, 1092, 1132, 1247, 1287, 1409
 \glossaries 70, 74,
 228, 229, 241, 296–298, 345–
 350, 352–354, 357, 358, 363,
 364, 366–368, 383, 384, 386,
 388, 389, 395–400, 402–405, 412
 \GlossariesWarning 7, 10, 24,
 24, 25, 29–31, 36, 37, 39, 41,
 42, 51, 56, 60, 72, 76, 77, 84,
 87, 90–92, 117–119, 182, 196,
 198, 199, 203, 205–208, 211,
 213, 218, 221, 223, 288, 424,
 465, 467, 473, 474, 476, 477,
 480–483, 489, 491, 498, 500,
 504, 506, 524, 525, 531, 532,
 559, 570, 571, 573–577, 580,
 582, 583, 595, 597, 627, 663,
 714–716, 723, 724, 726, 727,
 730–733, 739, 741, 748, 751,
 754, 756, 758, 759, 775, 782,
 783, 811, 822, 823, 825–829,
 832, 834, 835, 847, 849, 876,
 881, 916, 968–970, 977, 978,
 980, 981, 985, 986, 988, 992,
 995, 997, 1004, 1006, 1007,
 1010, 1012, 1014, 1015, 1032,
 1066, 1076–1078, 1080–1085,
 1088, 1090, 1118, 1123–1125,
 1132, 1133, 1135, 1136, 1139–
 1141, 1143, 1147, 1149, 1151,
 1158, 1161, 1165, 1166, 1169,
 1170, 1187, 1221, 1231–1234,
 1236–1238, 1240, 1241, 1244–
 1246, 1274, 1278, 1280, 1287,
 1290, 1291, 1295, 1297, 1298,
 1304, 1306, 1308, 1315, 1318,
 1322, 1323, 1326, 1327, 1344,
 1378, 1388–1391, 1393–1395,
 1397, 1398, 1400, 1402, 1403, 1439
 \GlossariesWarningNoLine 7, 8, 24,
 24, 45, 202, 204, 209, 225, 298,
 465, 473, 474, 572–574, 576,
 584, 684, 714, 715, 723, 825,
 826, 828, 836, 937, 968, 969,
 977, 978, 1080, 1081, 1083,
 1084, 1092, 1121, 1123, 1124,
 1132, 1236, 1239, 1247, 1278,
 1287, 1300, 1392, 1393, 1396, 1404
 \glossary
 41, 42, 42, 482, 483, 664, 732,
 733, 917, 988, 1143, 1298, 1299
 \glossaryentry 217, 219,
 580, 581, 664, 832, 833, 917,
 1087, 1088, 1243, 1244, 1400, 1401
 glossaryentry (counter) 15
 \glossaryentryfield 258, 451, 594,

595, 649, 669–674, 676–678,
846, 847, 902, 922–927, 929–931

`\glossaryentrynumbers`
. 12, 12, 190, 223, 224,
242, 243, 246, 252, 253, 467,
468, 564, 583, 590, 592, 593,
666, 717, 816, 835, 841, 842,
844, 845, 919, 971, 1070, 1090,
1091, 1097, 1100, 1126, 1226,
1246, 1247, 1253, 1255, 1256,
1280, 1281, 1383, 1403, 1404,
1414, 1415, 1417, 1418, 1420, 1421

`\glossaryheader` . . . 190, 240, 244,
255, 299, 303, 304, 306–311,
313, 314, 317–322, 324, 326–
329, 331, 333, 334, 338, 340,
342, 398, 407–412, 415, 564,
589, 594, 666, 685, 687–692,
694–703, 705–707, 709–713,
815, 840, 846, 919, 938, 940–
946, 948–957, 959–967, 1070,
1096, 1101, 1226, 1252, 1257,
1383, 1413, 1416, 1422, 1442–1450

`\glossarymark` 57, 57, 489,
739, 740, 995, 1150, 1306, 1307

`\glossaryname`
20, 49, 51, 52, 471, 486, 487,
721, 736, 737, 975, 991–993,
1130, 1146–1148, 1285, 1303–1305

`\glossarypostamble`
. 56, 190, 240, 244, 489,
564, 589, 666, 739, 815, 841,
919, 995, 1070, 1096, 1149,
1226, 1252, 1306, 1383, 1413, 1416

`\glossarypreamble`
. 56, 189, 240, 244, 489,
563, 588, 666, 739, 815, 840,
919, 994, 1070, 1096, 1149,
1225, 1251, 1306, 1382, 1413, 1416

`\glossarysection`
. 56, 189, 240, 241, 244, 489,
563, 588, 666, 739, 815, 840,
919, 995, 1070, 1096, 1149,
1225, 1251, 1306, 1382, 1413, 1416

`\glossarystyle` 261, 597, 849

`glossarysubentry` (counter) 15

`\glossarysubentryfield`
. 258, 451, 595, 596,
649, 669–678, 847, 902, 922–931

`\glossarytitle` 189, 222, 223, 240,
241, 244, 250, 563, 583, 588,
591, 666, 815, 834, 835, 840,
843, 919, 1070, 1090, 1091,
1096, 1098, 1225, 1246, 1251,
1254, 1382, 1403, 1413, 1416, 1419

`\glossarytoctitle`
11, 20–22, 43, 49, 52, 57, 189,
223, 240, 241, 244, 250, 251,
467, 471, 472, 483, 484, 486,
487, 489, 563, 583, 588, 591,
592, 666, 717, 721, 722, 734,
736, 737, 739, 815, 834, 840,
843, 844, 919, 971, 975, 976,
989, 991, 993, 995, 1070, 1090,
1096, 1099, 1126, 1130, 1131,
1144, 1146, 1148, 1149, 1225,
1246, 1251, 1254, 1255, 1280,
1285, 1299, 1303, 1305, 1306,
1382, 1403, 1413, 1416, 1419, 1420

`\glossentry` . . . 107, 224, 243, 246,
255, 299, 303–306, 308, 309,
318, 319, 321, 331, 333, 335,
338, 340, 342, 398, 407, 409,
410, 413, 519, 583, 590, 595,
685, 687–690, 694–696, 701–
703, 705–707, 709–712, 770,
835, 842, 847, 938, 940–944,
948–950, 955–957, 959–964,
966, 1026, 1091, 1097, 1101,
1181, 1246, 1253, 1257, 1338,
1403, 1415, 1417, 1422, 1446–1449

`\GLOSSentrydesc` 256, 382, 383

`\GlossEntryDesc` 257, 382, 383

`\Glossentrydesc` . . . 141, 256, 382,
383, 451, 595, 648, 847, 901,
1102, 1202, 1257, 1258, 1359, 1423

`\glossentrydesc` 141, 256,
256, 300, 303–306, 308–310,
318, 319, 321, 331, 333, 335,
338–340, 342, 358, 382, 407–
411, 413, 414, 451, 595, 648,
686–690, 694–696, 701–703,
705–707, 709–713, 846, 901,
939–944, 948–950, 955–957,
959–961, 963–966, 1102, 1202,
1257, 1258, 1359, 1423, 1446–1450

`\GLOSSentryname` 256, 379, 380

`\GlossEntryName` 256, 379, 380

`\Glossentryname`
141, 256, 378, 379, 594, 846,

1102, 1202, 1257, 1359, 1422, 1423
 \glossentryname 141, 255, 255, 256,
 258, 300, 302–306, 308, 309,
 318, 319, 321, 331, 333, 335,
 338, 340, 342, 357, 378, 379,
 407–410, 413, 414, 450, 451,
 594, 648, 685–690, 694–696,
 701–703, 705–707, 709–713,
 846, 901, 939–944, 948–950,
 955–957, 959–961, 963, 964,
 966, 1101, 1102, 1202, 1257,
 1258, 1359, 1422, 1423, 1446–1450
 \glossentrynameother 141, 1202, 1359
 \GLOSSentrysymbol 257, 380, 381
 \GlossEntrySymbol 257, 381
 \Glossentrysymbol 141, 257,
 380, 381, 451, 595, 648, 847,
 902, 1102, 1202, 1258, 1359, 1423
 \glossentrysymbol 141, 257,
257, 300, 309, 310, 321, 335,
 342, 358, 380, 381, 407–410,
 413, 414, 451, 595, 648, 685,
 686, 690, 696, 703, 707, 709–
 713, 847, 902, 939, 944, 950,
 957, 961, 963, 964, 966, 1102,
 1202, 1258, 1359, 1423, 1446–1450
 \GLS . . . 145, 543, 794, 1048, 1204, 1361
 \Gls 118, 144,
144, 265, 285, 524, 542, 599,
 611, 775, 794, 851, 863, 1032,
 1048, 1105, 1118, 1187, 1203,
 1261, 1273, 1344, 1360, 1426, 1439
 \gls . . . 41, 117, 143, 144, 203, 254,
 265, 284, 483, 524, 542, 573,
 593, 599, 611, 733, 775, 794,
 825, 845, 851, 863, 988, 1032,
 1047, 1048, 1080, 1101, 1105,
 1118, 1143, 1187, 1203, 1236,
 1256, 1261, 1273, 1298, 1344,
 1360, 1393, 1422, 1426, 1427, 1439
 \gls@@delimN . . . 263, 1104, 1260, 1425
 \gls@accessibility
 425, 425, 628, 881, 882
 \gls@accsupp@engine 425, 425, 628, 881
 \gls@Alphpage 211,
212, 216, 578, 579, 829–831,
 1085, 1087, 1240–1242, 1397, 1399
 \gls@alphpage 211,
212, 216, 578, 579, 829–831,
 1085, 1087, 1240–1242, 1397, 1399
 \gls@arabicpage 211,
212, 216, 578, 579, 829–831,
 1085, 1086, 1240–1242, 1397, 1399
 \gls@assign@desc 100,
100, 106, 514, 518, 765, 769,
 1021, 1025, 1176, 1179, 1333, 1336
 \gls@assign@desc@field 26
 \gls@assign@descplural 612, 619,
 621, 623, 657, 660, 661, 864,
 871, 872, 874, 876, 911, 913, 914
 \gls@assign@descplural@field . . 26
 \gls@assign@field
 89, 89, 94, 95, 100, 102, 104–
 106, 418, 507, 510, 514, 515,
 517, 518, 680, 757, 761, 765–
 769, 933, 1013, 1017, 1021–
 1025, 1168, 1171, 1172, 1176–
 1179, 1325, 1328, 1329, 1333–1336
 \gls@assign@firstpl 612–615, 617,
 619, 621–623, 657–661, 863,
 864, 866–872, 874, 876, 910–914
 \gls@assign@name@field 26
 \gls@assign@plural 612–615,
 617, 619, 621–623, 657–661,
 864, 866–872, 874, 876, 911–914
 \gls@assign@symbolplural
 612–615, 617,
 621–623, 658, 659, 661, 864,
 866–870, 874, 876, 911, 912, 914
 \gls@assign@type@field 26
 \gls@automake@makegloss . . . 38,
 39, 40, 200, 986, 987, 1079,
 1141, 1142, 1234, 1296, 1297, 1391
 \gls@automake@nr
 38, 39, 40, 47, 200, 481, 572,
 732, 823, 986, 987, 1078, 1141,
 1142, 1234, 1296, 1297, 1301, 1391
 \gls@automake@option . 38, 39, 40,
 986, 987, 1141, 1142, 1296, 1297
 \gls@automake@val 38–40, 481, 732,
 986, 987, 1141, 1142, 1296, 1297
 \gls@begindocdefs
 91, 91, 508, 759, 1015, 1169, 1326
 \gls@checkisacronymlist
 . 23, 129, 473, 533, 722, 784,
 977, 1037, 1131, 1192, 1286, 1349
 \gls@checkseeallowed
 84, 84, 202, 203,
 504, 507, 572, 573, 754, 824,

825, 1010, 1079, 1081, 1165,
 1235, 1236, 1321, 1322, 1392, 1393
 \gls@checkseeallowed@preambleonly
 84,
 504, 507, 754, 1010, 1165, 1322
 \gls@codepage ... 36, 68, 198, 199,
 225, 480, 496, 570, 571, 584,
 730, 746, 747, 822, 823, 836,
 984, 1002, 1076, 1077, 1092,
 1139, 1157, 1232, 1233, 1247,
 1294, 1295, 1314, 1389, 1390, 1404
 \gls@debug@nr 7,
 465, 483, 714, 733, 968, 1123, 1277
 \gls@debug@val 7,
 465, 483, 714, 733, 968, 1123, 1277
 \gls@defdocnewglossaryentry 89,
 91, 115, 507, 508, 522, 758,
 759, 773, 1014, 1015, 1030,
 1168, 1169, 1185, 1325, 1326, 1342
 \gls@defglossaryentry ... 89–92,
 101, 101, 507, 508, 514, 758,
 759, 765, 1014, 1015, 1021,
 1168–1170, 1176, 1325–1327, 1333
 \gls@delimN 263, 263, 1104, 1260, 1425
 \gls@delimR 262, 263, 1104, 1260, 1425
 \gls@disablepagerefexpansion ..
 211, 212,
 216, 577–579, 829–831, 1084,
 1085, 1087, 1240, 1242, 1397, 1399
 \gls@do@addxdyattribute
 ... 62, 492, 742, 998, 1153, 1310
 \gls@docdef@addlabel 90, 91,
 758, 759, 1014, 1015, 1169, 1326
 \gls@docdef@labellist
 ... 90, 90, 758, 1014, 1169, 1326
 \gls@docclearpage 58, 59,
 490, 491, 740, 741, 996, 1151, 1308
 \gls@doentryfmt
 .. 77, 501, 751, 1007, 1162, 1318
 \gls@dostubst 133,
 536, 786, 787, 1040, 1195, 1352
 \gls@dotoc@title .. 223, 224, 250,
 583, 591, 835, 843, 1090, 1091,
 1098, 1099, 1246, 1254, 1403, 1419
 \gls@end@sanitizesort
 27, 28, 475, 724,
 725, 979, 1133, 1134, 1288, 1289
 \gls@endcheck
 88, 89, 507, 757, 1013, 1168, 1325
 \gls@glossary .. 41, 42, 210, 217–
 219, 483, 577, 580, 581, 733,
 829, 832, 833, 988, 989, 1084,
 1087, 1088, 1143, 1239, 1243,
 1244, 1298, 1299, 1396, 1400, 1401
 \gls@glsdefs@addlabel
 90, 92, 758, 759,
 1014, 1015, 1169, 1170, 1326, 1327
 \gls@glsdefs@labellist
 ... 90, 90, 758, 1014, 1169, 1326
 \gls@gobbleopt .. 78, 79, 501, 502,
 751, 752, 1007, 1008, 1162, 1319
 \gls@grplabel ... 295, 684, 937, 1121
 \gls@hypergroup@rerun
 298, 298, 684, 937, 1121
 \gls@if@measuring 109,
 110, 111, 1027, 1028, 1182, 1339
 \gls@ifnotmeasuring
 110, 111, 112, 211, 520, 521,
 771, 772, 1027–1029, 1084,
 1182–1184, 1240, 1339–1341, 1397
 \gls@inlinepostchild 299–
 301, 669, 685, 686, 922, 938, 939
 \gls@inlinesep 299,
 300, 669, 685, 686, 922, 938, 939
 \gls@inlinesubsep 299,
 300, 669, 685, 686, 922, 938, 939
 \gls@islistofacronyms 22,
 472, 473, 722, 976, 977, 1131, 1286
 \gls@istfilebase 53, 53,
 198, 199, 488, 570, 571, 738,
 822, 823, 993, 994, 1076, 1077,
 1148, 1232, 1233, 1305, 1389, 1390
 \gls@kernelglossredefs@nr
 42, 988, 1143, 1298
 \gls@kernelglossredefs@val
 42, 988, 1143, 1298
 \gls@label
 . 265, 599, 851, 1105, 1261, 1426
 \gls@level 87, 103, 104, 241,
 242, 245, 246, 506, 516, 589,
 590, 756, 767, 841, 1012, 1023,
 1097, 1167, 1177, 1178, 1252,
 1253, 1324, 1334, 1335, 1414, 1417
 \gls@local@stepcounter
 110, 110, 1027, 1182, 1339
 \gls@measuring@reset .. 110, 111,
 1027, 1028, 1182, 1183, 1339, 1340
 \gls@mfirstuc@nr
 44, 989, 990, 1144, 1300

<code>\gls@mfirstuc@val</code>	44, 989, 1144, 1300	<code>\gls@orgnumber</code>	215, 216, 579, 831, 1086, 1242, 1399
<code>\gls@newacronym</code> 266, 266, 1106, 1261, 1427	<code>\gls@orgroman</code>	216, 579, 831, 1086, 1087, 1242, 1399
<code>\gls@noidx@nonumberlist</code>	84, 206, 1010, 1083, 1165, 1238, 1322, 1395	<code>\gls@orgromannumeral</code>	215, 216, 579, 831, 1086, 1242, 1399
<code>\gls@noidxglossary</code> 203, 207, 573, 576, 825, 828, 1080, 1083, 1236, 1239, 1393, 1395	<code>\gls@orgthe</code>	215, 216, 579, 831, 1086, 1242, 1399
<code>\gls@nonumberlist@nr</code> 84, 504, 754, 1010, 1165, 1322	<code>\gls@original@glossary</code> 41, 42, 482, 483, 732, 733, 988, 1143, 1298, 1299
<code>\gls@nonumberlist@val</code> 84, 504, 754, 1010, 1165, 1322	<code>\gls@original@makeglossary</code> 41, 42, 482, 483, 732, 733, 988, 989, 1143, 1298, 1299
<code>\gls@nosetquote</code> 101, 191, 193, 196, 514, 565, 566, 569, 765, 816, 818, 821, 1021, 1071, 1073, 1075, 1176, 1227, 1228, 1231, 1333, 1384, 1385, 1388	<code>\gls@printglossary@begin</code> 223, 224, 1246, 1247, 1403, 1404
<code>\gls@number</code> 216, 579, 831, 1086, 1242, 1399	<code>\gls@printglossary@end</code> 224, 224, 1247, 1404
<code>\gls@numberedsection@nr</code>	11, 251, 467, 591, 592, 717, 843, 971, 1099, 1125, 1254, 1255, 1280, 1420	<code>\gls@protected@pagefmts</code> 133, 211, 212, 535, 578, 786, 829, 830, 1040, 1085, 1195, 1240, 1241, 1352, 1397, 1398
<code>\gls@numberedsection@val</code> 11, 251, 467, 591, 717, 843, 971, 1099, 1125, 1254, 1280, 1420	<code>\gls@romanpage</code> 211, 212, 216, 578, 579, 829–831, 1085, 1087, 1240–1242, 1397–1399
<code>\gls@numberpage</code> 211, 212, 216, 578, 579, 829–831, 1085, 1086, 1240–1242, 1397, 1399	<code>\gls@romanpage</code> 211, 212, 216, 578, 579, 829–831, 1085, 1086, 1240–1242, 1397–1399
<code>\gls@org@glossaryentryfield</code> 224, 583, 835, 1091, 1246, 1403	<code>\gls@save@numberlist</code>	... 12, 220, 467, 468, 582, 717, 833, 971, 1089, 1126, 1245, 1280, 1281, 1402
<code>\gls@org@glossarysubentryfield</code> 224, 583, 835, 1091, 1246, 1247, 1403	<code>\gls@seen@index@nr</code> 9, 466, 716, 970, 1125, 1279
<code>\gls@org@insert</code> 616, 618, 620, 868, 870, 871, 873	<code>\gls@seen@index@val</code> 9, 466, 716, 970, 1125, 1279
<code>\gls@org@label</code> 109–111, 1027, 1028, 1182, 1339	<code>\gls@set@xr@key</code> 83, 83, 504, 754, 1010, 1164, 1321
<code>\gls@org@link</code> 110, 111, 1027, 1028, 1182, 1339	<code>\gls@start@measuring</code> 110, 110, 111, 369, 1027, 1028, 1182, 1183, 1339, 1340
<code>\gls@org@refstepcounter</code>	.. 109, 110, 111, 1027, 1028, 1182, 1339	<code>\gls@stop@measuring</code>	... 111, 111, 373, 1028, 1182, 1183, 1339, 1340
<code>\gls@org@target</code> 110, 111, 1027, 1028, 1182, 1339	<code>\gls@suffixF</code>	54, 55, 190, 192, 488, 564, 566, 666, 668, 738, 816–818, 920, 921, 994, 1070–1072, 1149, 1226, 1228, 1306, 1383, 1385
<code>\gls@org@alph</code> 216, 579, 831, 1086, 1087, 1242, 1399	<code>\gls@suffixFF</code>	55, 55, 190, 192, 488, 564, 566, 666–668, 738, 816, 818, 920, 921, 994, 1070–1072, 1149, 1226, 1228, 1306, 1383, 1385
<code>\gls@org@alph</code> 215, 216, 579, 831, 1086, 1087, 1242, 1399		
<code>\gls@org@arabic</code> 215, 216, 579, 831, 1086, 1242, 1399		

<code>\gls@text</code>	125, 531,	736, 851, 857, 860–862, 865,
	782, 1036, 1190, 1191, 1347, 1348	992, 1106, 1112, 1115–1117,
<code>\gls@the</code>	216, 579, 831, 1086, 1242, 1399	1146, 1261, 1267, 1268, 1271,
<code>\gls@thissty</code>		1272, 1303, 1427, 1433, 1436–1438
	. . . 33, 479, 729, 983, 1138, 1293	<code>\glsacrshortcutsfalse</code>
<code>\gls@tmp</code>	200, 202, 209,	. . . 47, 485, 735, 990, 1145, 1302
	287, 576, 620, 828, 873, 1078,	<code>\glsacrshortcutstrue</code>
	1080, 1083, 1118, 1234, 1235,	. . . 47, 484, 735, 990, 1145, 1302
	1239, 1274, 1391, 1392, 1396, 1439	<code>\glsacsospace</code>
<code>\gls@tmplen</code>		274, <u>275</u> ,
	. 140, 141, 355, 411–414, 541,	277, 603, 604, 606, 855–857,
	675, 711–713, 793, 928, 965,	1110–1112, 1266, 1268, 1431–1433
	966, 1047, 1201, 1358, 1448–1450	<code>\glsadd</code> . . . 41, 184, <u>184</u> , 185, 483,
<code>\gls@tr@set@acronym@toctitle</code> . .		560, 561, 733, 812, 988, 1067,
 21, 22, 472,	1143, 1222, 1223, 1298, 1379, 1380
	721, 722, 976, 1130, 1131, 1285	<code>\glsaddall</code>
<code>\gls@tr@set@main@toctitle</code> <u>184</u> , 561, 812, 1067, 1223, 1380
	. . . 20, 471, 721, 975, 1130, 1285	<code>\glsaddallunused</code>
<code>\gls@tr@set@numbers@toctitle</code> . .		. <u>185</u> , 561, 812, 1067, 1223, 1380
	. . . 43, 484, 734, 989, 1144, 1299	<code>\glsaddkey</code>
<code>\gls@tr@set@symbols@toctitle</code> <u>94</u> , 510, 761, 1017, 1172, 1329
	. . . 43, 483, 734, 989, 1144, 1299	<code>\GlsAddLetterGroup</code>
<code>\gls@translate@nr</code> <u>68</u> , 497, 747, 1003, 1157, 1314
	. . . 32, 478, 728, 982, 1137, 1292	<code>\glsaddprotectedpagefmt</code> . . <u>212</u> ,
<code>\gls@translate@val</code>		<u>213</u> , 578, 830, 1085, 1241, 1398
	. . . 32, 478, 728, 982, 1137, 1292	<code>\GlsAddSortRule</code> <u>66</u> ,
<code>\gls@wrglossary</code>	211,	495, 496, 746, 1001, 1156, 1313
	<u>211</u> , 577, 829, 1084, 1240, 1397	<code>\glsaddstoragekey</code>
<code>\gls@xdy@set@lang</code> <u>93</u> , 510, 760, 1016, 1171, 1328
 <u>35</u> , 198, 199, 225, 984,	<code>\GlsAddXdyAlphabet</code>
	1076, 1077, 1091, 1139, 1232,	. . . <u>63</u> , 493, 743, 999, 1154, 1311
	1233, 1247, 1294, 1389, 1390, 1404	<code>\GlsAddXdyAttribute</code>
<code>\gls@xdystring</code>	133, 535, 61, <u>61</u> , 62, 63, 492,
	536, 786, 787, 1040, 1195, 1352	493, 663, 742, 743, 916, 917,
<code>\gls@xindy@glsnumbersfalse</code>		997–999, 1152–1154, 1309–1311
	. . . 36, 480, 730, 985, 1139, 1295	<code>\GlsAddXdyCounters</code> <u>60</u> , 61,
<code>\gls@xindy@glsnumberstrue</code>		62, 80, 491, 492, 502, 663, 741–
	. . . 35, 480, 730, 984, 1139, 1294	743, 752, 916, 997, 998, 1008,
<code>\gls@xr@key</code> 9, 10, 84, <u>84</u> , 466, 467,		1152, 1153, 1163, 1309, 1310, 1320
	504, 716, 754, 970, 1010, 1125,	<code>\GlsAddXdyLocation</code>
	1164, 1165, 1279, 1280, 1321, 1322 <u>65</u> , 494, 495, 663, 745,
<code>\glsaccessdisplay</code> <u>437</u> , 636, 889		917, 1000, 1001, 1155, 1156, 1312
<code>\glsaccessibility</code>		<code>\GlsAddXdyStyle</code>
	<u>425</u> , 433, 434, 628, 633, 881, 886	. . <u>67</u> , 496, 746, 1002, 1156, 1313
<code>\glsaccsupp</code> 433, 434, <u>434</u> , 633, 886, 887		<code>\glsautomakefalse</code> 39, 40, 200, 481,
<code>\glsacronymtrue</code>		482, 572, 732, 824, 987, 1078,
	. . . 21, 472, 722, 976, 1131, 1285	1079, 1141, 1142, 1234, 1297, 1391
<code>\glsacrpluralsuffix</code>		<code>\glsautomaketrue</code> 39,
	. . 50, <u>50</u> , 266, 276, 280–282,	40, 481, 732, 987, 1141, 1142, 1297
	486, 599, 605, 608–610, 612,	<code>\glsautoprefix</code> <u>10</u> , 11,
		251, 467, 592, 717, 844, 971,

1099, 1125, 1126, 1255, 1280, 1420
`\glsbackslash` 46, [185](#),
 561, 813, 1067, 1223, 1301, 1380
`\glscapitalisewords` 178, [178](#), 379–
 382, 809, 1063, 1064, 1219, 1376
`\glscapscase`
 120, 121, 123, 124, 143–
 148, 164–172, 279, 437, 439–
 445, 455, 456, 525, 526, 528–
 530, 542–545, 551–555, 607,
 616, 620, 636, 637, 639–644,
 652, 776, 777, 779–781, 794–
 796, 802–807, 859, 869, 873,
 889, 890, 892–898, 905, 1033–
 1035, 1047–1050, 1056–1061,
 1114, 1188–1190, 1203–1205,
 1212–1216, 1269, 1270, 1345–
 1347, 1360–1362, 1369–1373, 1435
`\glscategory` 433, 633, 886
`\glsclearpage` 58, [59](#), 490, 491, 740,
 741, 996, 1150, 1151, 1307, 1308
`\glsclosebrace` 65, [185](#),
 190, 191, 495, 561, 564, 666,
 667, 745, 813, 816, 919, 920,
 1000, 1001, 1067, 1070, 1071,
 1155, 1223, 1226, 1312, 1380, 1383
`\glscompositor` 54,
[54](#), 65, 192, 488, 494, 566, 666,
 668, 738, 744, 817, 919, 921,
 994, 1000, 1072, 1148, 1149,
 1155, 1228, 1305, 1306, 1312, 1385
`\glscounter` 23, [23](#), 48, 60, 79,
 105, 131, 473, 485, 491, 502,
 517, 534, 663, 723, 735, 741,
 752, 768, 785, 916, 977, 991,
 997, 1008, 1024, 1038, 1132,
 1145, 1152, 1162, 1179, 1193,
 1286, 1302, 1309, 1319, 1336, 1350
`\glscurrententrylabel`
 141, 221, 224,
 582, 583, 834, 835, 1089, 1091,
 1202, 1245–1247, 1359, 1402, 1403
`\glscurrentfieldvalue`
 76, [77](#), 500, 501,
 750, 751, 1006, 1161, 1317, 1318
`\glscustomtext`
 120, 122, 124, 143–148, 164–
 172, 278, 279, 437, 440, 442–
 445, 455, 456, 525, 528–530,
 542–545, 551–555, 607, 612,
 613, 616, 618, 620, 636, 639,
 641–644, 652, 776, 778–781,
 794–796, 802–807, 859, 865,
 868–871, 873, 889, 892, 894,
 896–898, 905, 906, 1033–1035,
 1047–1050, 1056–1061, 1113,
 1114, 1188–1190, 1203–1205,
 1212–1216, 1269, 1270, 1345–
 1347, 1360–1362, 1369–1373, 1435
`\GlsDeclareNoHyperList` [23](#),
 24, 473, 723, 977, 1132, 1286, 1287
`\glsdefaultshortaccess`
 461, [461](#), 657–660, 910–913
`\glsdefaultttype` 21,
[21](#), 35, 36, 56, 67, 69, 77, 78,
 102, 120, 209, 222, 471, 489,
 496, 497, 501, 515, 525, 531,
 576, 582, 583, 721, 739, 746,
 747, 751, 766, 776, 782, 828,
 834, 975, 984, 994, 1002, 1003,
 1007, 1022, 1033, 1083, 1090,
 1130, 1139, 1149, 1157, 1158,
 1161, 1162, 1177, 1187, 1188,
 1239, 1245, 1246, 1285, 1294,
 1306, 1314, 1315, 1318, 1319,
 1334, 1344, 1345, 1396, 1402, 1403
`\glsdefmain`
 20, 21, 80, 471, 502, 721, 752,
 975, 1008, 1130, 1163, 1285, 1320
`\glsdefs@newdocentry`
 [91](#), 92, 759, 1015, 1170, 1327
`\GLSdesc` [155](#), 547, 799, 1052, 1208, 1365
`\Glsdesc` 154,
[154](#), 547, 799, 1052, 1208, 1365
`\glsdesc` 154,
[154](#), 547, 799, 1052, 1208, 1365
`\GLSdescplural`
 [156](#), 548, 799, 1053, 1208, 1365
`\Glsdescplural` 155,
[155](#), 547, 799, 1053, 1208, 1365
`\glsdescplural` 155, [155](#),
 547, 799, 1052, 1053, 1208, 1365
`\glsdescriptionaccessdisplay` . .
 [435](#), 447,
 451, 634, 637–639, 646, 648,
 649, 888, 891, 892, 899, 901, 902
`\glsdescriptionpluralaccessdisplay`
 [435](#), 447,
 634, 636, 637, 646, 888–890, 899

<code>\glsdescwidth</code>	306, 306 , 307, 308, 311, 312, 314, 315, 317, 317 , 318–322, 330 , 331–334, 336–338, 338 , 339–344, 688– 691, 693–697, 701–708, 942, 943, 945–951, 954–956, 958–962	<code>\glsdisplaynumberlist</code>	182 , 205, 559, 574, 811, 826, 1066, 1082, 1221, 1237, 1378, 1394
<code>\glsdetoklabel</code>	38, 71 , 71 , 72–76, 85, 92, 97– 101, 107, 112, 113, 115–117, 130, 172–174, 182–184, 203– 205, 214, 217, 224, 235, 236, 241, 242, 245, 246, 248, 253, 254, 256, 289–293, 345, 412, 418, 419, 434–437, 462, 463, 497–500, 505, 508, 512–514, 519, 521–523, 533, 555, 556, 559, 560, 573–575, 579, 583, 585, 586, 589–591, 593, 594, 625–628, 634–636, 661, 662, 680, 712, 731, 748–750, 755, 759, 763–765, 769, 772–774, 784, 807, 811, 812, 825–827, 830, 831, 835, 837, 838, 841– 843, 845, 846, 878–881, 887– 889, 914, 915, 933, 934, 965, 986, 1003–1006, 1010, 1011, 1015, 1019–1021, 1026, 1029– 1031, 1038, 1061, 1062, 1066, 1067, 1081, 1082, 1086, 1087, 1091, 1093, 1094, 1097, 1098, 1100, 1101, 1119, 1120, 1141, 1158–1161, 1165, 1170, 1173– 1176, 1180, 1184–1186, 1193, 1216, 1217, 1221, 1222, 1236– 1238, 1242, 1246–1249, 1252– 1254, 1256, 1257, 1274–1276, 1296, 1315–1318, 1322, 1327, 1330–1333, 1337, 1341–1343, 1350, 1373, 1374, 1378, 1379, 1393–1395, 1398, 1399, 1403, 1410, 1411, 1413–1415, 1417– 1419, 1421, 1422, 1440, 1441, 1449	<code>\glsdohyperlink</code>	142 , 142 , 143, 541, 542, 793, 794, 1047, 1202, 1203, 1359, 1360
<code>\glsdisablehyper</code> 142 , 542, 793, 1047, 1203, 1360	<code>\glsdohyperlinkhook</code> 142 , 142 , 1202, 1359
<code>\Glsdisp</code>	149, 149 , 1050, 1205, 1362	<code>\glsdohypertarget</code>	140 , 142 , 143, 541, 542, 793, 794, 1047, 1201, 1203, 1358, 1360
<code>\glsdisp</code>	148 , 149, 545, 796, 1050, 1205, 1362	<code>\glsdohypertargethook</code>	141, 141 , 1201, 1202, 1358, 1359
<code>\glsdisplay</code> 125 , 525, 531, 532, 776, 782		<code>\glsdoifexists</code>	71 , 73, 75, 97–100, 111, 112, 143–149, 164–172, 178, 182, 184, 205, 227, 420–424, 443–445, 450, 451, 498–500, 512–514, 520, 521, 542–545, 551–555, 557, 559, 560, 574, 575, 643, 644, 648, 681–683, 748–750, 763– 765, 771, 772, 794–797, 802– 806, 809, 811, 812, 826, 827, 896, 897, 901, 902, 934–937, 1003–1006, 1019–1021, 1028, 1029, 1047–1050, 1056–1061, 1064–1067, 1081, 1082, 1158– 1160, 1173–1175, 1183, 1184, 1203–1206, 1212–1216, 1219, 1221, 1222, 1237, 1238, 1315– 1317, 1330–1332, 1340, 1341, 1360–1363, 1369–1373, 1376, 1378, 1379, 1394, 1395, 1405
<code>\glsdisplayfirst</code> 125 , 525, 531, 776, 782	<code>\glsdoifexistsordo</code>	72 , 128, 173, 498, 533, 555, 748, 783, 807, 1004, 1037, 1061, 1159, 1192, 1217, 1315, 1349, 1374
		<code>\glsdoifexistsorwarn</code>	72 , 248, 255–257, 498, 591, 594, 595, 748, 843, 846, 847, 1004, 1098, 1101, 1102, 1158, 1254, 1257, 1258, 1315, 1418, 1422, 1423
		<code>\glsdoifnoexists</code>	71 , 89, 100, 498, 507, 514, 748, 757, 765, 1004, 1013, 1021, 1158, 1168, 1176, 1315, 1325, 1333
		<code>\glsdonohyperlink</code> 130, 142 , 142 , 534, 541, 542, 785, 793, 1038, 1047, 1193, 1202, 1203, 1350, 1359, 1360

<code>\glsdoparenifnotempty</code>	548, 557, 646, 799, 808, 899, 1053, 1063, 1208, 1218, 1365, 1375
. 287 , 620 , 872 , 1118 , 1274 , 1439	
<code>\glsdosanitizesort</code>	155, 156, 176 , 447, 525, 526, 547, 548, 556, 636, 637, 646, 776, 777, 799, 808, 889, 890, 899, 1053, 1063, 1208, 1218, 1365, 1375
. 16, 17, 469, 470, 719, 973, 1128, 1282, 1283	
<code>\glsdoshowtarget</code>	
. 8 , 140, 142, 970, 1047, 1124, 1201, 1202, 1279, 1358, 1359	
<code>\glsenableentrycount</code>	
. 114 , 522, 523, 773, 774, 1030, 1031, 1185, 1186, 1342, 1343	
<code>\glsenablehyper</code>	
. 143 , 542, 793, 1047, 1203, 1360	
<code>\glsencapwrcontent</code>	
. 130 , 131 , 1038, 1193, 1350	
<code>\glsentryaccess</code>	
. 430 , 434, 632, 634, 885, 887	
<code>\glsentrycounter</code> 131 ,	
260, 264, 534, 596, 598, 785, 848, 850, 1038, 1103, 1105, 1193, 1259, 1260, 1350, 1424, 1426	
<code>\glsentrycounterfalse</code>	
. 15 , 469, 719, 973, 1127, 1282	
<code>\glsentrycounterlabel</code>	
. 254 , 254 , 593, 594, 845, 846, 1101, 1256, 1257, 1422	
<code>\GlsEntryCounterLabelPrefix</code> 251 ,	
253, 254, 592, 593, 844, 845, 1099–1101, 1255, 1256, 1420–1422	
<code>\glsentrycountertrue</code>	
. 15 , 469, 719, 973, 1127, 1282	
<code>\glsentrycurrcount</code>	
. 115 – 117 , 522 – 524 , 773 – 775 , 1030–1032, 1185, 1186, 1342, 1343	
<code>\Glsentrydesc</code> 155 , 175 ,	
176, 256, 447, 451, 547, 556, 595, 646, 648, 799, 808, 847, 899, 901, 1052, 1062, 1102, 1208, 1218, 1258, 1365, 1375, 1423	
<code>\glsentrydesc</code> 141 , 154 ,	
155 , 175 , 256, 257, 447, 451, 526–528, 547, 556, 595, 637– 639, 646, 648, 777–779, 799, 808, 846, 891, 892, 899, 901, 1052, 1062, 1102, 1202, 1208, 1218, 1257, 1359, 1365, 1375, 1423	
<code>\glsentrydescaccess</code>	
. 431 , 435, 632, 634, 885, 888	
<code>\Glsentrydescplural</code>	
. 156 , 176 , 176 , 447,	
548, 557, 646, 799, 808, 899, 1053, 1063, 1208, 1218, 1365, 1375	
<code>\glsentrydescplural</code>	
155, 156, 176 , 447, 525, 526, 547, 548, 556, 636, 637, 646, 776, 777, 799, 808, 889, 890, 899, 1053, 1063, 1208, 1218, 1365, 1375	
<code>\glsentrydescpluralaccess</code>	
. 431 , 435, 632, 635, 885, 888	
<code>\Glsentryfirst</code>	
118 , 122, 151, 177, 177 , 439, 446, 524, 527, 529, 546, 557, 638, 640, 645, 775, 778, 780, 797, 809, 891, 894, 898, 1032, 1034, 1051, 1063, 1187, 1189, 1206, 1219, 1344, 1346, 1363, 1376	
<code>\glsentryfirst</code> 117 , 122, 150, 151,	
177 , 439, 440, 446, 524, 526, 527, 529, 546, 557, 638, 640, 641, 645, 775, 777, 778, 780, 797, 809, 891–894, 898, 1032, 1034, 1051, 1063, 1187–1189, 1206, 1219, 1344–1346, 1363, 1376	
<code>\glsentryfirstaccess</code>	
. 430 , 435, 632, 634, 885, 887	
<code>\Glsentryfirstplural</code>	
. 119 , 121, 153, 178, 178 , 438, 446, 525, 526, 528, 546, 557, 637, 640, 645, 776, 777, 779, 798, 809, 890, 893, 898, 1033, 1052, 1063, 1187, 1188, 1207, 1219, 1344, 1345, 1364, 1376	
<code>\glsentryfirstplural</code> 119 ,	
121, 152, 153, 177 , 438, 446, 524–526, 528, 546, 547, 557, 636, 637, 639, 640, 645, 775– 777, 779, 798, 809, 889, 890, 892, 893, 898, 899, 1032, 1033, 1051, 1052, 1063, 1187, 1188, 1207, 1219, 1344, 1345, 1364, 1376	
<code>\glsentryfirstpluralaccess</code>	
. 431 , 435, 632, 634, 885, 887	
<code>\glsentryfmt</code> 79 , 81 , 120 , 502, 525, 752, 753, 776, 1008, 1033, 1162, 1163, 1188, 1319, 1320, 1345	
<code>\Glsentryfull</code> 181 , 181 ,	
272, 280, 282, 457, 459, 559, 602, 608, 609, 653, 655, 811, 854, 860, 861, 906, 908, 1065, 1108, 1115, 1116, 1221, 1264,	

1270, 1272, 1378, 1430, 1436, 1437
 \glsentryfull 181,
 272, 280, 282, 457, 459, 559,
 602, 608, 609, 653, 655, 810,
 854, 860, 861, 906, 908, 1065,
 1108, 1115, 1116, 1221, 1264,
 1270, 1272, 1378, 1430, 1436, 1437
 \Glsentryfullpl 182, 182,
 272, 280, 282, 457, 459, 559,
 602, 608, 610, 653, 655, 811,
 854, 860, 861, 907, 908, 1065,
 1109, 1115, 1116, 1221, 1264,
 1271, 1272, 1378, 1430, 1436, 1437
 \glsentryfullpl 181,
 272, 280, 282, 457, 459, 559,
 602, 608, 610, 653, 655, 811,
 854, 860, 861, 906, 908, 1065,
 1109, 1115, 1116, 1221, 1264,
 1270, 1272, 1378, 1430, 1436, 1437
 \glsentryitem 254, 300,
 302–306, 308, 309, 318, 319,
 321, 331, 333, 335, 338, 340,
 342, 402, 407, 409, 410, 413,
 594, 669–674, 676–678, 685,
 687–690, 694–696, 701–703,
 705–707, 709–712, 846, 922–
 931, 938, 940–944, 948–950,
 955–957, 959–961, 963, 964,
 966, 1101, 1257, 1422, 1446–1449
 \Glsentrylong 118, 169,
 175, 181, 181, 274, 279, 280,
 442, 445, 452, 456, 457, 524,
 553, 556, 559, 603, 607, 608,
 643, 644, 649, 652, 653, 775,
 805, 807, 810, 811, 855, 859,
 860, 896, 897, 902, 905, 906,
 1032, 1059, 1062, 1065, 1110,
 1114, 1115, 1187, 1215, 1218,
 1221, 1265, 1266, 1270, 1344,
 1372, 1375, 1378, 1431, 1435, 1436
 \glsentrylong 117, 124,
 168, 170, 175, 181, 181, 274,
 275, 277–283, 442, 444, 445,
 452–461, 524, 531, 553, 554,
 556, 559, 603–611, 618, 642,
 644, 649–656, 775, 781, 805,
 807, 810, 811, 855–863, 871,
 896–898, 902–910, 1032, 1036,
 1059, 1062, 1065, 1110–1117,
 1187, 1190, 1214, 1215, 1217,
 1221, 1265–1273, 1344, 1347,
 1371, 1372, 1374, 1378, 1431–1438
 \glsentrylongaccess
 431, 436, 632, 635, 885, 888
 \Glsentrylongpl 119, 171, 181,
181, 182, 274, 279, 280, 443,
 452, 455, 457, 525, 555, 559,
 603, 604, 607, 608, 643, 649,
 652, 653, 776, 806, 810, 811,
 855, 856, 859, 860, 896, 902,
 905–907, 1033, 1060, 1065,
 1110, 1114, 1115, 1187, 1216,
 1221, 1265, 1266, 1269–1271,
 1344, 1373, 1378, 1431, 1435, 1436
 \glsentrylongpl 119, 125,
 170, 172, 181, 181, 274, 275,
 279–282, 443, 452, 453, 455,
 457–459, 524, 531, 554, 555,
 559, 603, 604, 607–610, 618,
 624, 643, 649, 650, 652–655,
 775, 782, 806, 807, 810, 811,
 855, 856, 859–861, 871, 877,
 896, 902, 903, 905–908, 1032,
 1036, 1060, 1061, 1065, 1110,
 1111, 1114–1116, 1187, 1191,
 1215, 1216, 1221, 1265–1267,
 1269–1272, 1344, 1348, 1372,
 1373, 1378, 1431, 1432, 1435–1437
 \glsentrylongpluralaccess
 432, 436, 632, 635, 886, 888
 \Glsentryname 154, 174, 174,
 256, 447, 451, 547, 556, 594,
 646, 648, 798, 807, 846, 899,
 901, 1052, 1061, 1062, 1102,
 1208, 1217, 1257, 1365, 1374, 1423
 \glsentryname 141, 153, 154,
173, 220, 256, 345, 412, 447,
 450, 547, 556, 645, 646, 648,
 712, 798, 799, 807, 899, 901,
 965, 1052, 1061, 1062, 1089,
 1202, 1207, 1208, 1217, 1245,
 1359, 1364, 1365, 1374, 1402, 1449
 \glsentrynumberlist 182, 182,
 204, 559, 574, 811, 826, 1065,
 1066, 1081, 1221, 1237, 1378, 1394
 \glsentryparent
 178, 558, 809, 1064, 1219, 1376
 \Glsentryplural
 121, 152, 176, 176, 438, 446,
 526, 528, 546, 557, 636, 639,

645, 776, 779, 798, 808, 890,
 893, 898, 1033, 1051, 1063,
 1188, 1207, 1218, 1345, 1364, 1375
 \glsentryplural
 121, 151, 152, 176, 438, 446,
 525, 526, 528, 546, 557, 636,
 637, 639, 640, 645, 776, 777,
 779, 798, 808, 889, 890, 892,
 893, 898, 1033, 1051, 1063,
 1188, 1207, 1218, 1345, 1364, 1375
 \glsentrypluralaccess
 431, 435, 632, 634, 885, 887
 \Glsentryprefix
 419, 419, 421, 680, 682, 933, 935
 \glsentryprefix .. 418, 419, 420,
 423, 680, 681, 683, 933, 934, 936
 \Glsentryprefixfirst
 418, 418, 422, 680, 682, 933, 935
 \glsentryprefixfirst 418, 418, 420,
 423, 680, 681, 683, 933, 934, 936
 \Glsentryprefixfirstplural
 418, 418, 422, 680, 683, 933, 936
 \glsentryprefixfirstplural
 418, 418, 421,
 424, 680, 681, 684, 933, 935, 937
 \Glsentryprefixplural
 419, 419, 422, 680, 682, 933, 936
 \glsentryprefixplural
 418, 419, 421,
 424, 680, 681, 683, 933, 935, 937
 \glsentryprevcount
 115, 116, 522, 523, 773, 774,
 1030, 1031, 1185, 1186, 1342, 1343
 \Glsentryshort 123, 165, 175, 180,
 180, 275, 281, 282, 441–443,
 453, 458, 459, 530, 551, 556,
 558, 604, 609, 642, 643, 650,
 654, 655, 781, 803, 808, 810,
 856, 861, 895, 896, 903, 907,
 908, 1035, 1057, 1062, 1065,
 1111, 1115, 1116, 1190, 1212,
 1218, 1220, 1267, 1271, 1272,
 1347, 1369, 1375, 1377, 1432, 1437
 \glsentryshort 123,
 125, 164, 166, 175, 180, 181,
 272, 274, 275, 277, 278, 280–
 283, 441–444, 452–461, 530,
 531, 551, 552, 556, 558, 559,
 602–611, 641–644, 649–656,
 781, 802, 803, 808, 810, 811,
 854–863, 895–897, 902–910,
 1035, 1036, 1056, 1057, 1062,
 1065, 1109–1117, 1189, 1190,
 1212, 1213, 1218, 1220, 1221,
 1264–1273, 1346, 1347, 1369,
 1370, 1375, 1377, 1378, 1430–1438
 \glsentryshortaccess
 431, 436, 632, 635, 885, 888
 \Glsentryshortpl 123, 167,
 181, 181, 275, 281, 282, 440,
 453, 458, 459, 529, 552, 559,
 604, 609, 610, 641, 650, 654,
 655, 780, 804, 810, 856, 861,
 894, 903, 907, 908, 1034, 1058,
 1065, 1111, 1116, 1189, 1214,
 1220, 1221, 1267, 1271, 1272,
 1346, 1371, 1377, 1378, 1432, 1437
 \glsentryshortpl 123, 125, 166, 168,
 180, 181, 182, 274, 275, 280–
 282, 440, 441, 443, 452, 453,
 457–459, 529, 531, 552, 553,
 559, 603, 604, 608–610, 624,
 641, 643, 649, 650, 653–655,
 780, 782, 804, 810, 811, 855,
 856, 860, 861, 877, 894, 896,
 902, 903, 906–908, 1034, 1036,
 1058, 1065, 1110, 1111, 1115,
 1116, 1189, 1191, 1213, 1214,
 1220, 1221, 1265–1267, 1270–
 1272, 1346, 1348, 1370, 1371,
 1377, 1378, 1431, 1432, 1436, 1437
 \glsentryshortpluralaccess
 431, 436, 632, 635, 885, 888
 \glsentrysort
 . 178, 558, 809, 1064, 1219, 1376
 \Glsentrysymbol 157, 177,
 177, 257, 448, 451, 548, 557,
 595, 646, 649, 800, 808, 847,
 899, 902, 1053, 1063, 1102,
 1209, 1219, 1258, 1366, 1376, 1423
 \glsentrysymbol .. 141, 156, 157,
 177, 257, 345, 448, 451, 526–
 528, 548, 557, 595, 613, 616,
 620, 637–639, 646, 648, 777–
 779, 799, 800, 808, 847, 865,
 869, 873, 891, 892, 899, 902,
 1053, 1063, 1102, 1202, 1209,
 1218, 1258, 1359, 1366, 1375, 1423
 \glsentrysymbolaccess
 431, 435, 632, 634, 885, 887

<code>\Glsentrysymbolplural</code>	<code>\glsentryuseriiaccess</code>
. 158, 177, <u>177</u> , 448, <u>432</u> , 436, 632, 635, 886, 888
548, 557, 646, 800, 809, 900,	<code>\Glsentryuseriii</code> 161, 179, <u>179</u> , 449,
1054, 1063, 1209, 1219, 1366, 1376	549, 558, 647, 801, 810, 900,
<code>\glsentrysymbolplural</code> 157,	1055, 1064, 1210, 1220, 1367, 1377
158, <u>177</u> , 448, 525, 526, 548,	<code>\glsentryuseriii</code> 160, 161, <u>179</u> , 449,
557, 613, 616, 620, 636, 637,	549, 558, 647, 801, 810, 900,
646, 776, 777, 800, 809, 865,	1055, 1064, 1210, 1220, 1367, 1377
869, 873, 889, 890, 900, 1053,	<code>\glsentryuseriiiaccess</code>
1054, 1063, 1209, 1219, 1366, 1376 <u>432</u> , 436, 633, 635, 886, 888
<code>\glsentrysymbolpluralaccess</code>	<code>\Glsentryuseriv</code> 162, 180, <u>180</u> , 449,
. <u>431</u> , 435, 632, 634, 885, 888	550, 558, 647, 801, 810, 900,
<code>\Glsentrytext</code>	1055, 1064, 1211, 1220, 1368, 1377
122, 150, 176, <u>176</u> , 439, 445,	<code>\glsentryuseriv</code>
527, 529, 545, 557, 638, 640, 161, 162, <u>179</u> , 449, 550,
645, 778, 780, 797, 808, 891,	558, 647, 801, 810, 900, 901,
893, 898, 1034, 1050, 1063,	1055, 1064, 1211, 1220, 1368, 1377
1189, 1206, 1218, 1346, 1363, 1375	<code>\glsentryuserivaccess</code>
<code>\glsentrytext</code> 121, 122, <u>432</u> , 437, 633, 635, 886, 889
149, 150, <u>176</u> , 183, 220, 439,	<code>\Glsentryuseriv</code> 163, 180, <u>180</u> , 450,
445, 526, 527, 529, 545, 557,	550, 558, 647, 802, 810, 901,
560, 582, 637, 638, 640, 641,	1056, 1065, 1211, 1220, 1368, 1377
645, 777–780, 797, 808, 812,	<code>\glsentryuseriv</code> 162,
833, 891, 893, 894, 898, 1034,	163, <u>180</u> , 450, 550, 558, 647,
1050, 1063, 1066, 1089, 1188,	648, 802, 810, 901, 1055, 1056,
1189, 1206, 1218, 1222, 1245,	1064, 1065, 1211, 1220, 1368, 1377
1345, 1346, 1363, 1375, 1379, 1402	<code>\glsentryuserivaccess</code>
<code>\glsentrytextaccess</code> <u>432</u> , 437, 633, 635, 886, 889
. <u>430</u> , 434, 632, 634, 885, 887	<code>\Glsentryuserivi</code> 164, 180, <u>180</u> , 450,
<code>\glsentrytitlecase</code> <u>178</u> , 256, 257,	550, 558, 648, 802, 810, 901,
557, 558, 809, 1064, 1219, 1376	1056, 1065, 1212, 1220, 1369, 1377
<code>\glsentrytype</code> 102,	<code>\glsentryuserivi</code> 163,
<u>178</u> , 250, 515, 558, 766, 809,	164, <u>180</u> , 450, 550, 551, 558,
1022, 1064, 1177, 1219, 1334, 1376	648, 802, 810, 901, 1056, 1065,
<code>\Glsentryuseri</code> 159, 179, <u>179</u> , 448,	1211, 1212, 1220, 1368, 1369, 1377
549, 558, 647, 800, 810, 900,	<code>\glsentryuseriviaccess</code>
1054, 1064, 1210, 1220, 1367, 1377 <u>432</u> , 437, 633, 636, 886, 889
<code>\glsentryuseri</code> 158,	<code>\glsesclocationsfalse</code>
159, <u>179</u> , 448, 548, 549, 558, 13, 203, 573, 825,
647, 800, 809, 900, 1054, 1064,	972, 1080, 1126, 1236, 1281, 1393
1209, 1210, 1219, 1366, 1367, 1376	<code>\glsesclocationstrue</code> 468, 717
<code>\glsentryuseriaccess</code>	<code>\glsexpandfields</code>
. <u>432</u> , 436, 632, 635, 886, 888 <u>89</u> , 507, 757, 1013, 1168, 1325
<code>\Glsentryuserii</code> 160, 179, <u>179</u> , 449,	<code>\glsfieldaccsupp</code>
549, 558, 647, 801, 810, 900, <u>433</u> , <u>433</u> , 434, 633, 634, 886, 887
1054, 1064, 1210, 1220, 1367, 1377	<code>\glsfielddef</code>
<code>\glsentryuserii</code> <u>98</u> , 513, 763, 1019, 1174, 1331
. 159, 160, <u>179</u> , 449, 549,	<code>\glsfieldedef</code>
558, 647, 800, 801, 810, 900, <u>97</u> , 512, 763, 1019, 1174, 1331
1054, 1064, 1210, 1220, 1367, 1377	

<code>\glsfieldfetch</code>	<code>\glsgetgrouplabel</code>
. 98, 178, 513, 557, 764, 809, 1020, 1064, 1174, 1219, 1331, 1376	. 260, 596, 848, 1103, 1258, 1424
<code>\glsfieldgdef</code>	<code>\glsgetgrouptitle</code>
.. 97, 512, 763, 1019, 1174, 1331 259, 299, 302–304, 325–330, 408–411, 415, 596, 685, 687, 688, 698–701, 709– 711, 713, 848, 938, 940, 941, 951–954, 963–965, 967, 1102, 1122, 1258, 1423, 1442–1448, 1450
<code>\glsfieldxdef</code>	<code>\gls glossarymark</code> ... 57, 57, 489, 739, 995, 1149, 1150, 1306, 1307
.. 97, 512, 763, 1019, 1173, 1330	<code>\gls groupheading</code> 191, 243, 246, 259, 299, 303, 304, 306, 308, 309, 317, 319, 321, 325–331, 333, 334, 338, 340, 342, 398, 407– 412, 414, 415, 564, 590, 596, 667, 685, 687–690, 694–696, 698–703, 705–707, 709–713, 816, 842, 848, 920, 938, 940– 944, 948–957, 959–967, 1071, 1097, 1102, 1226, 1253, 1258, 1383, 1415, 1417, 1423, 1442–1450
<code>\glsfindwidesttoplevelname</code>	<code>\gls groupskip</code>
..... 411, 711, 965, 1448 190, 191, 243, 246, 258, 300, 303, 307, 308, 310, 313, 314, 318– 321, 331, 333, 335, 339, 341, 343, 346, 398, 408, 409, 411, 414, 564, 590, 596, 666, 667, 686, 687, 689, 690, 692, 695, 696, 701–703, 705–707, 709– 711, 713, 816, 842, 848, 919, 920, 939, 940, 942–946, 948– 950, 955–957, 959–961, 963– 965, 967, 1070, 1071, 1097, 1102, 1226, 1253, 1258, 1383, 1415, 1417, 1423, 1446–1448, 1450
<code>\GLSfirst</code>	<code>\gls hyperfirstfalse</code> 281, 458, 609, 654, 861, 907, 1115, 1271, 1436
. 151, 546, 797, 1051, 1206, 1363	<code>\gls hyperfirsttrue</code>
<code>\Glsfirst</code> 34, 479, 729, 983, 1138, 1293
150, 546, 797, 1051, 1206, 1363	<code>\gls hyperlink</code>
<code>\glsfirst</code>	183, 220, 560, 582, 812, 833, 1066, 1089, 1222, 1245, 1379, 1402
150, 150, 546, 797, 1050, 1051, 1206, 1363	<code>\gls hypernavsep</code>
<code>\glsfirstaccessdisplay</code> 298, 298, 299, 685, 938, 1122
435, 439, 440, 446, 634, 638, 640, 641, 645, 887, 891–894, 898	<code>\gls hypernumber</code>
<code>\GLSfirstplural</code>	55, 262, 264, 265, 488, 597–599, 739, 849–851, 994, 1104, 1105, 1149, 1259, 1261, 1306, 1425, 1426
. 153, 546, 798, 1052, 1207, 1364	<code>\gls ifhyper</code>
<code>\Glsfirstplural</code>	127, 532, 783
152, 152, 546, 798, 1051, 1207, 1364	
<code>\glsfirstplural</code>	
152, 152, 546, 798, 1051, 1207, 1364	
<code>\glsfirstpluralaccessdisplay</code> ..	
..... 435, 438, 446, 634, 636, 637, 639, 640, 645, 887, 889, 890, 892, 893, 898, 899	
<code>\glsfirstpluralacesdisplay</code> ...	
..... 438, 640, 893	
<code>\gls genacfmt</code>	
. 122, 273–275, 281, 440, 452, 458, 529, 603, 604, 609, 641, 649, 650, 654, 780, 855, 856, 860, 894, 902, 903, 907, 1034, 1109–1111, 1115, 1189, 1265, 1266, 1271, 1346, 1431, 1432, 1436	
<code>\gls genentryfmt</code>	
120, 120, 273–275, 279, 281, 285, 437, 452, 456, 458, 528, 603, 604, 607, 609, 611, 613, 614, 616, 618, 620, 622, 624, 639, 649, 650, 652, 654, 779, 855, 856, 859, 860, 863, 865, 867, 868, 871, 873, 875, 877, 892, 902, 903, 906, 907, 1033, 1109– 1111, 1114, 1115, 1118, 1188, 1265, 1266, 1270, 1271, 1274, 1345, 1431, 1432, 1435, 1436, 1439	

<code>\glsifhyperon</code>	129 , 532 , 533 , 783 , 784 , 1038 , 1192 , 1349	<code>\glsinlinesubdescformat</code>	300 , 301 , 669 , 686 , 922 , 939 , 940
<code>\glsIfListOfAcronyms</code> .	22 , 22 , 23 , 472 , 473 , 722 , 976 , 977 , 1131 , 1286	<code>\glsinlinesubnameformat</code>	300 , 301 , 669 , 686 , 922 , 939 , 940
<code>\glsifmeasuring</code>	110 , 110 , 111 , 1027 , 1028 , 1182 , 1339	<code>\glsinlinesubseparator</code> 300 , 301 , 669 , 686 , 922 , 939
<code>\glsifplural</code> 120 , 123 , 143–148 , 164– 172 , 279 , 437 , 440 , 441 , 443– 445 , 455 , 525 , 528–530 , 542– 545 , 551–555 , 607 , 613 , 616 , 618 , 620 , 636 , 639 , 641–644 , 652 , 776 , 779–781 , 794–796 , 802–807 , 859 , 865 , 868 , 871 , 873 , 889 , 892 , 894–898 , 905 , 1033–1035 , 1047–1050 , 1056– 1061 , 1114 , 1188–1190 , 1203– 1205 , 1212–1216 , 1269 , 1345– 1347 , 1360–1362 , 1369–1373 , 1435	<code>\glsinsert</code> 121–124 , 143–148 , 164– 172 , 279 , 438–445 , 455 , 456 , 525–530 , 542–545 , 551–555 , 607 , 613 , 616 , 618 , 620 , 636– 644 , 652 , 776–781 , 794–796 , 802–807 , 859 , 865 , 868–871 , 873 , 889–898 , 905 , 906 , 1033– 1035 , 1048–1050 , 1056–1061 , 1114 , 1188–1190 , 1203–1205 , 1212–1216 , 1269 , 1270 , 1345– 1347 , 1360–1362 , 1369–1373 , 1435	
<code>\glsifusedtranslatordict</code> 32 , 478 , 728 , 982 , 1137 , 1292	<code>\glskeylisttok</code> 270 , 271 , 461 , 601 , 612–615 , 617 , 619 , 621–625 , 657–661 , 853 , 863 , 864 , 866– 872 , 874–876 , 878 , 910–914 , 1107 , 1108 , 1263 , 1264 , 1428 , 1429	
<code>\glsifusetranslator</code> 31–34 , 51 , 52 , 415 , 416 , 478 , 479 , 486 , 487 , 662 , 679 , 727–729 , 736 , 737 , 915 , 932 , 982 , 983 , 992 , 993 , 1136– 1138 , 1147 , 1148 , 1292 , 1293 , 1304	<code>\glslabel</code> 101 , 120– 124 , 129 , 130 , 165–172 , 273– 275 , 279 , 281 , 438–442 , 452 , 455 , 456 , 458 , 515 , 525–530 , 533 , 534 , 551–555 , 603 , 604 , 607 , 609 , 613 , 616 , 618 , 620 , 636–642 , 649 , 650 , 652 , 654 , 765 , 776–781 , 784 , 785 , 803– 807 , 855 , 856 , 859 , 860 , 865 , 868 , 869 , 871 , 873 , 889–895 , 902 , 903 , 905 , 907 , 1021 , 1033– 1035 , 1037 , 1038 , 1057–1061 , 1109–1111 , 1114 , 1115 , 1176 , 1188–1190 , 1192 , 1193 , 1212– 1216 , 1265 , 1266 , 1269–1271 , 1333 , 1345–1347 , 1349 , 1350 , 1369–1373 , 1431 , 1432 , 1435 , 1436	
<code>\glsignore</code> 185 , 561 , 812 , 1067 , 1223 , 1380	<code>\glslabelhypertarget</code> 141 , 1202 , 1359	
<code>\glsindexingsetting</code> 16 , 19 , 20 , 200 , 203 , 973 , 975 , 1079 , 1080 , 1128–1130 , 1234 , 1236 , 1282 , 1284 , 1391 , 1393	<code>\glslabelhypertargetdefs</code> 141 , 141 , 1202 , 1359
<code>\glsindexonlyfirstfalse</code> 34 , 479 , 729 , 983 , 1138 , 1293	<code>\glslabelhypertargetprefix</code> 141 , 141 , 1202 , 1359
<code>\glsinlinedescformat</code> 300 , 301 , 669 , 686 , 922 , 939	<code>\glslabelhypertargetvalue</code> 141 , 141 , 1202 , 1359
<code>\glsinlinedopostchild</code>	299 , 300 , 669 , 685 , 686 , 922 , 938 , 939	<code>\glslabeltok</code>	270 , 271 , 601 , 611–625 , 657–660 , 853 , 863–873 , 875–878 , 910– 913 , 1107 , 1108 , 1263 , 1264 , 1429
<code>\glsinlinenamedescformat</code>	300 , 301 , 669 , 685 , 686 , 922 , 939		
<code>\glsinlineparentchildseparator</code> 300 , 301 , 669 , 686 , 922 , 939		
<code>\glsinlinenepostchild</code> 300 , 301 , 669 , 686 , 922 , 939		
<code>\glsinlineseparator</code> 300 , 301 , 669 , 686 , 922 , 939		

<code>\glsletentryfield</code>	643, 649, 650, 652–655, 657, 888, 896, 902, 903, 905–908, 910
. 172 , 555, 807, 1061, 1216, 1373	
<code>\Glslink</code> .. 128 , 129, 1037, 1192, 1349	<code>\glslongpluralaccesskey</code> 462 , 661, 914
<code>\glslink</code> 128 , 129, 271, 272, 280–282, 286, 456–459, 532, 601, 602, 608, 609, 612, 653–655, 783, 853, 854, 859–861, 865, 906, 908, 1037, 1108, 1114–1116, 1118, 1192, 1264, 1270–1272, 1274, 1349, 1429, 1435–1437, 1439	<code>\glslongpluralkey</code> 267 , 462, 599, 661, 851, 914, 1106, 1262, 1427
<code>\glslinkcheckfirsthyperhook</code> 129 , 129 , 533, 784, 1037, 1038, 1192, 1349	<code>\glslongtok</code> ... 270 , 271, 272, 274, 275, 280, 281, 452, 456, 458, 461, 601–604, 608, 609, 611– 619, 621–625, 649, 650, 653, 654, 657–660, 853–856, 859, 861, 863–872, 874–878, 902, 903, 906, 907, 910–914, 1108– 1111, 1114, 1115, 1263–1266, 1270, 1271, 1429–1432, 1435, 1436
<code>\glslinkpostsetkeys</code>	<code>\glslowercase</code> 4 , 967, 1122, 1277
..... 129 , 130, 533, 534, 784, 1038, 1192, 1193, 1349, 1350	<code>\glsLtpenaltycheck</code>
<code>\glslinkvar</code> 127 , 127 , 128, 532, 783, 1036, 1037, 1191, 1192, 1348, 1349 316 , 316 , 693, 694, 947
<code>\glslistdottedwidth</code>	<code>\glsmakefirstuc</code> . 44, 990, 1144, 1300
.... 305 , 305 , 670, 688, 923, 941	<code>\glsmcols</code>
<code>\glslistexpandedname</code> .. 302 , 302 , 940	330, 697–700, 951–954, 1442–1445
<code>\glslistgroupheaderfmt</code> ... 302 , 302 , 303, 304, 687, 688, 940, 941	<code>\GlsMcolTreeSetup</code>
<code>\glslistinit</code>	323
..... 302 , 303, 940	<code>\glsmeasuredepth</code> 111 , 1028, 1183, 1340
<code>\glslistnavigationitem</code>	<code>\glsmeasureheight</code> 111 , 141, 1028, 1047, 1183, 1201, 1340, 1358
.... 302 , 303, 304, 687, 940, 941	<code>\glsmeasurewidth</code>
<code>\glslocalreset</code> 111 , 362, 363, 365, 412– 414, 1028, 1183, 1340, 1449, 1450
112 , 113, 520, 521, 771, 772, 1028, 1029, 1183, 1184, 1340, 1341	<code>\glsmfuaddmap</code>
<code>\glslocalresetall</code> 5 , 6 , 97, 118, 119, 129, 144, 146, 149–163, 165, 167, 169, 171, 256, 257, 268, 269, 418, 419, 421, 422, 967, 968, 1018, 1019, 1032, 1037, 1048–1056, 1058–1060, 1062–1065, 1102, 1107, 1122, 1173, 1187, 1192, 1203–1214, 1216, 1257, 1258, 1262, 1263, 1277, 1330, 1344, 1349, 1360–1371, 1373, 1423, 1428
. 113 , 521, 772, 1029, 1184, 1341	<code>\glsmfublocker</code>
<code>\glslocalunset</code> 5 , 6 , 6 , 97, 145, 147, 149, 151–165, 167, 169, 171, 967, 968, 1019, 1048–1060, 1122, 1173, 1204–1212, 1214–1216, 1277, 1330, 1361–1369, 1371–1373
112 , 114, 127, 521, 522, 542– 545, 772, 773, 794–796, 1029, 1030, 1036, 1184, 1191, 1341, 1348	<code>\glsmfuexcl</code> ... 5 , 6 , 174, 176–182, 184, 256, 257, 967, 968, 1067, 1122, 1217–1222, 1277, 1374–1379
<code>\glslocalunsetall</code>	<code>\glsmoveentry</code>
. 114 , 522, 773, 1030, 1184, 1341	. 107 , 519, 769, 1026, 1180, 1337
<code>\glslocationcstoencap</code>	<code>\GLSname</code> 154 , 547, 798, 1052, 1208, 1365
..... 55 , 994, 1149, 1306	
<code>\glslongaccessdisplay</code>	
..... 436 , 442, 444, 445, 452–461, 635, 642–644, 649–657, 888, 896–898, 902–910	
<code>\glslongaccesskey</code> 461 , 661, 914	
<code>\glslongkey</code>	
267 , 461, 599, 661, 851, 914, 1106, 1262, 1427	
<code>\glslongpluralaccessdisplay</code> 436 , 443, 452, 453, 455, 457–459, 635,	

`\Glsname` 153, 842, 1081, 1097, 1098, 1237, 1253, 1394, 1414, 1415, 1417, 1418
`\glsname` 153, 153, 547, 798, 1052, 1207, 1364
`\glsnameaccessdisplay` 434, 447, 450, 451, 634, 645, 646, 648, 649, 887, 899, 901, 902
`\glsnamefont` .. 256, 262, 450, 451, 594, 595, 597, 628, 648, 846, 847, 849, 881, 901, 1101, 1102, 1104, 1257, 1259, 1422, 1423, 1425
`\glsnavhypergroupdotarget` . 297, 297
`\glsnavhyperlink` 295, 298, 299, 684, 685, 937, 938, 1121, 1122
`\glsnavhyperlinkname` 295, 296, 297, 684, 937, 1121
`\glsnavhypertarget` 297, 302–304, 326–330, 400, 408, 410, 411, 415, 684, 687, 688, 698–701, 710, 711, 713, 937, 940, 941, 951–954, 963–965, 967, 1121, 1442–1445, 1447, 1448, 1450
`\glsnavigation` 298, 302–304, 326–330, 406, 408, 410, 411, 415, 684, 687, 698–700, 710, 711, 713, 938, 940, 941, 951–954, 963–965, 967, 1121, 1442–1445, 1447, 1448, 1450
`\glsnavigationitem` 298, 298
`\glsnextpages` 13, 84, 223, 253, 468, 504, 583, 593, 717, 754, 835, 845, 972, 1010, 1091, 1100, 1126, 1165, 1246, 1256, 1281, 1322, 1403, 1421
`\glsnoexpandfields` 89, 507, 757, 1013, 1168, 1325
`\glsnogroupskipfalse` 14, 468, 718, 972, 1127, 1281
`\glsnoidxdisplayloc` 205–207, 247, 575, 576, 591, 827, 828, 843, 1082, 1083, 1098, 1238, 1239, 1254, 1395, 1396, 1418
`\glsnoidxdisplayloclisthandler` 205, 247, 575, 590, 827, 842, 1082, 1098, 1237, 1254, 1394, 1418
`\GlsNoIdxDoRerunCheck` 206, 209
`\glsnoidxinithook` 240, 241
`\glsnoidxitemhook` 241, 241
`\glsnoidxloclist` . 204, 242, 243, 246, 246, 574, 590, 826, 841, 842, 1081, 1097, 1098, 1237, 1253, 1394, 1414, 1415, 1417, 1418
`\glsnoidxloclisthandler` .. 247, 247, 590, 842, 1098, 1253, 1418
`\GlsNoIdxMissingAction` 241, 241, 244
`\glsnoidxnumberlistloophandler` 205, 206, 575, 576, 827, 1082, 1083, 1238, 1395
`\glsnoidxprecontenthook` 240, 241, 1413
`\glsnoidxprenumberlist` 85, 246, 1010, 1097, 1165, 1253, 1322, 1417, 1418
`\glsnoidxstripaccents` 28, 28, 475, 476, 725, 726, 979, 980, 1134, 1135, 1289, 1290
`\glsnomakeindexwarning` 60, 193, 491, 566, 741, 818, 997, 1073, 1151, 1228, 1308, 1385
`\glsnonextpages` 84, 223, 253, 504, 583, 593, 754, 835, 845, 1010, 1091, 1100, 1165, 1246, 1256, 1322, 1403, 1421
`\glsnopostdotfalse` 14, 468, 718, 972, 1127, 1281
`\glsnoxindywarning` 54, 60, 61–63, 66–68, 186, 187, 488, 491–493, 495, 496, 561, 562, 738, 741–743, 745–747, 813, 814, 994, 997–999, 1001, 1002, 1068, 1149, 1151–1154, 1156, 1157, 1223, 1224, 1306, 1308–1314, 1380, 1381
`\glsnumberformat` 55, 182, 488, 559, 560, 739, 811, 994, 1066, 1149, 1221, 1306, 1378
`\glsnumberlistloop` 205, 206, 575, 827, 1082, 1083, 1238, 1395
`\glsnumbersgroupname` 43, 50, 52, 260, 484, 486, 487, 596, 734, 736, 738, 848, 989, 992, 993, 1103, 1144, 1146, 1148, 1258, 1299, 1303, 1305, 1424
`\glsnumlistlastsep` 183, 183, 205, 560, 575, 812, 826, 1066, 1082, 1222, 1237, 1379, 1394
`\glsnumlistparser` 183, 202, 560, 573, 812, 824, 1066, 1080, 1222, 1235, 1379, 1392

<code>\glsnumlistsep</code>	183, 183, 205, 560, 575, 812, 826, 1066, 1082, 1222, 1237, 1379, 1394	<code>\glsplural</code>	151, 151, 546, 798, 1051, 1206, 1207, 1363, 1364
<code>\glsopenbrace</code>	65, 185, 190, 191, 495, 561, 564, 666, 667, 745, 812, 816, 919, 920, 1000, 1001, 1067, 1070, 1071, 1155, 1223, 1226, 1312, 1380, 1383	<code>\glspluralaccessdisplay</code>	435, 438, 446, 634, 636, 637, 639, 640, 645, 887, 889, 890, 892, 893, 898
<code>\glsorder</code>	35, 35, 197–199, 201, 238, 480, 570–572, 587, 729, 730, 821–824, 839, 984, 1076– 1079, 1094, 1139, 1232–1235, 1250, 1294, 1389, 1390, 1392, 1411	<code>\glspluralsuffix</code>	50, 50, 104, 105, 271, 274, 275, 452, 453, 457–460, 486, 517, 603, 604, 649, 650, 654, 656, 736, 767, 768, 855, 856, 903, 907– 909, 992, 1024, 1108, 1110, 1111, 1146, 1178, 1264, 1266, 1267, 1303, 1335, 1429, 1431, 1432
<code>\glsorg@endtheglossary</code>	<code>\glspostdescription</code>
<code>\glsorg@PrintChanges</code>	<code>\glspostlinkhook</code>	128, 129, 144– 149, 165–172, 443–445, 533, 542–545, 551–555, 643–645, 784, 794–797, 803–807, 896– 898, 1037, 1048–1050, 1056– 1061, 1192, 1203–1206, 1212– 1216, 1349, 1360–1363, 1369–1373
<code>\glsorg@theglossary</code>	<code>\glsprefixsep</code>	420, 420, 421–424, 681–684, 934–937
<code>\glspagelistwidth</code>	306, 308, 311, 312, 314, 315, 317, 319–322, 331, 333, 334, 336, 337, 338, 340–344, 688– 691, 693–697, 701–708, 942, 943, 945–951, 955, 956, 958–962	<code>\glsprestandardsort</code>
<code>\glspar</code>	53, 488, 738, 993, 1148, 1305	<code>\glsquote</code>	46, 185, 561, 813, 1067, 1223, 1301, 1380
<code>\glspatchLToutput</code>	312–315, 316, 692, 693, 945–947	<code>\glsrefentry</code>
<code>\glspatchtabularx</code>	<code>\glsreset</code>	111, 113, 520, 521, 771, 772, 1028, 1029, 1183, 1184, 1340, 1341
<code>\glspenaltygroupskip</code>	313, 314, 316, 692, 693, 945–947	<code>\glsresetall</code>
<code>\glspercentchar</code> 92, 186, 190, 191, 508, 509, 561, 564, 565, 759, 760, 813, 815, 817, 1015, 1016, 1067, 1070, 1071, 1170, 1223, 1226, 1227, 1327, 1380, 1383, 1384		<code>\glsresetcurrcountfalse</code>
<code>\GLSpl</code>	147, 544, 796, 1049, 1205, 1362	<code>\glsresetentrycounter</code>
<code>\Glspl</code>	119, 146, 146, 285, 525, 544, 611, 775, 795, 863, 1033, 1049, 1118, 1187, 1204, 1273, 1344, 1361, 1439		
<code>\glspl</code>	118, 145, 146, 285, 524, 543, 611, 775, 795, 863, 1032, 1048, 1049, 1118, 1187, 1204, 1273, 1344, 1361, 1439		
<code>\GLSplural</code>		
<code>\Glsplural</code>	151, 151, 546, 798, 1051, 1207, 1364		

<code>\glsresetentrylist</code>	190, 240, 244, 252, 253, <u>253</u> , 564, 589, 592, 593, 666, 816, 840, 844, 845, 919, 1070, 1096, 1100, 1226, 1252, 1256, 1383, 1413, 1416, 1421	726, 727, 978, 980, 981, 1133, 1135, 1136, 1287, 1288, 1290, 1291
<code>\glsresetsubentrycounter</code> <u>253</u> , 254, 300, 593, 594, 669, 686, 845, 846, 922, 939, 1100, 1101, 1256, 1257, 1421, 1422	<code>\GlsSetQuote</code> 101, 191, <u>193</u> , 514, 565, 566, 569, 765, 816, 818, 821, 1021, 1071, 1073, 1075, 1176, 1227, 1228, 1231, 1333, 1384, 1385, 1388
<code>\glsrestoreLTOuput</code>	... <u>316</u> , 693, 947	<code>\glsSetSuffixF</code> <u>55</u> , 488, 738, 994, 1149, 1306
<code>\glssanitizeortfalse</code> 30, 31, 477, 727, 981, 1136, 1291	<code>\glsSetSuffixFF</code> <u>55</u> , 488, 738, 994, 1149, 1306
<code>\glssanitizeorttrue</code>	... 30, 477, 726, 727, 981, 1135, 1136, 1291	<code>\glssettoctitle</code> <u>49</u> , 52, 223, 486, 487, 583, 736, 737, 835, 991, 993, 1090, 1146, 1148, 1246, 1303, 1304, 1403
<code>\glsavenumberlistfalse</code> 12, 468, 717, 971, 1126, 1281	<code>\glssetwidest</code> ... <u>412</u> , 712, 965, 1449
<code>\glsavewritesfalse</code> 40, 482, 732, 988, 1142, 1298	<code>\GlsSetWriteIstHook</code> <u>193</u> , 566, 818, 1073, 1228, 1385
<code>\glssee</code>	<u>219</u> , 581, 833, 1089, 1244, 1401	<code>\GlsSetXdyCodePage</code> <u>68</u> , 496, 746, 747, 1002, 1157, 1314
<code>\glsseeformat</code> 189, 204–206, <u>219</u> , 563, 574, 575, 581, 666, 815, 825, 827, 833, 919, 1069, 1081, 1082, 1089, 1225, 1236, 1238, 1244, 1382, 1393, 1395, 1401	<code>\GlsSetXdyFirstLetterAfterDigits</code> <u>186</u> , 561, 813, 1067, 1068, 1223, 1380
<code>\glsseeitem</code> 220, <u>220</u> , 581, 582, 833, 1089, 1245, 1401, 1402	<code>\GlsSetXdyLanguage</code> <u>67</u> , 496, 746, 1002, 1157, 1314
<code>\glsseeitemformat</code> 220, <u>220</u> , 582, 833, 1089, 1245, 1402	<code>\GlsSetXdyLocationClassOrder</code> <u>66</u> , 495, 745, 1001, 1156, 1313
<code>\glsseelastsep</code> 220, <u>220</u> , 581, 833, 1089, 1245, 1401, 1402	<code>\GlsSetXdyMinRangeLength</code> .. <u>187</u> , 562, 813, 814, 1068, 1224, 1381
<code>\glsseelist</code> 219, <u>219</u> , 581, 833, 1089, 1244, 1401	<code>\GlsSetXdyNumberGroupOrder</code> <u>186</u> , 561, 562, 813, 1068, 1223, 1224, 1380, 1381
<code>\glsseesep</code> 220, <u>220</u> , 581, 833, 1089, 1245, 1401, 1402	<code>\GlsSetXdyStyles</code> <u>67</u> , 496, 746, 1002, 1156, 1157, 1313
<code>\glsentencecase</code>	<u>4</u> , 125, 129, 149, 173–175, 378–381, 967, 1036, 1037, 1050, 1061, 1062, 1122, 1190–1192, 1205, 1217, 1218, 1276, 1347–1349, 1362, 1374, 1375	<code>\glsshortaccessdisplay</code> 175, <u>436</u> , 441– 444, 452–461, 556, 635, 641– 644, 649–657, 808, 888, 895– 897, 902–910, 1062, 1218, 1375
<code>\glsSetAlphaCompositor</code> <u>54</u> , 488, 738, 994, 1149, 1306	<code>\glsshortaccesskey</code> <u>461</u> , 661, 914
<code>\glsSetCompositor</code> <u>54</u> , 488, 738, 994, 1149, 1305	<code>\glsshortaccsupp</code> . <u>433</u> , 434, 633, 886
<code>\glssetexpandfield</code> <u>25</u> , 26, 29–31, 474, 476, 477, 724, 726, 727, 978, 980, 981, 1133, 1135, 1136, 1287, 1288, 1290, 1291	<code>\glsshortkey</code> <u>266</u> , 461, 599, 661, 851, 914, 1106, 1262, 1427
<code>\glssetnoexpandfield</code> <u>25</u> , 26, 29–31, 474, 476, 477, 724,	<code>\glsshortplaccsupp</code> <u>433</u> , 633, 886
		<code>\glsshortpluralaccessdisplay</code> <u>436</u> , 440, 441, 443, 452, 453, 457–459, 635, 641, 643, 649, 650, 653–655, 657, 888, 894, 896, 902, 903, 906–908, 910

<code>\glsshortpluralaccesskey</code>	<code>\Glssymbol</code>
. 461 , 661 , 914	156 , 548 , 799 , 1053 , 1209 , 1366
<code>\glsshortpluralkey</code> 267 , 461 , 599 ,	<code>\glssymbol</code> 156 , 156 , 548 ,
661 , 851 , 914 , 1106 , 1262 , 1427	799 , 1053 , 1208 , 1209 , 1365 , 1366
<code>\glsshorttok</code> 270 , 271 ,	<code>\glssymbolaccessdisplay</code> 435 ,
461 , 601 , 611–619 , 621–625 ,	448 , 451 , 634 , 637–639 , 646 ,
657–661 , 853 , 863–878 , 910–	648 , 649 , 887 , 891 , 892 , 899 , 902
914 , 1107 , 1108 , 1263 , 1264 , 1429	<code>\glssymbolnav</code> 299 , 685 , 938 , 1122
<code>\glsshowaccsupp</code> 8 , 9 , 465 , 466 , 715 ,	<code>\GLSsymbolplural</code>
716 , 969 , 970 , 1124 , 1278 , 1279 158 , 548 , 800 , 1054 , 1209 , 1366
<code>\glsshowtarget</code> 8 , 9 , 465 , 466 ,	<code>\Glssymbolplural</code> 157 ,
715 , 969 , 970 , 1124 , 1278 , 1279	157 , 548 , 800 , 1053 , 1209 , 1366
<code>\glsshowtargetfont</code> 8 ,	<code>\glssymbolplural</code> 157 ,
8 , 465 , 466 , 715 , 969 , 1124 , 1279	157 , 548 , 800 , 1053 , 1209 , 1366
<code>\glsshowtargetfonttext</code>	<code>\glssymbolpluralaccessdisplay</code> .
. 8 , 8 , 969 , 1124 , 1279 435 , 448 , 634 ,
<code>\glsshowtargetinner</code>	636 , 637 , 646 , 887 , 889 , 890 , 900
. 8 , 8 , 969 , 1124 , 1278 , 1279	<code>\glssymbolsgroupname</code>
<code>\glsshowtargetouter</code> 8 ,	43 , 50 , 52 , 260 , 483 , 486 , 487 ,
8 , 466 , 715 , 969 , 1124 , 1278 , 1279	596 , 734 , 736 , 738 , 848 , 989 ,
<code>\glsshowtargetsymbol</code>	992 , 993 , 1103 , 1143 , 1146 ,
. 8 , 8 , 466 , 715 , 969 , 1124 , 1279	1148 , 1258 , 1299 , 1303 , 1305 , 1424
<code>\glssortnumberfmt</code>	<code>\glstarget</code> 110 ,
17 , 18 , 19 , 470 , 471 , 719 , 720 ,	111 , 255 , 301–306 , 308–310 ,
974 , 975 , 1128 , 1129 , 1283 , 1284	318 , 319 , 321 , 331 , 333 , 335 ,
<code>\glsspace</code> 267 , 267 ,	338–340 , 342 , 359 , 385–389 ,
600 , 851 , 864 , 1106 , 1262 , 1427	407–411 , 413 , 414 , 594–596 ,
<code>\glstepentry</code>	669–678 , 686–690 , 694–696 ,
253 , 254 , 593 , 594 , 845 , 846 ,	701–703 , 705–707 , 709–713 ,
1100 , 1101 , 1256 , 1257 , 1421 , 1422	846–848 , 922–931 , 939–944 ,
<code>\glstepsubentry</code>	948–950 , 955–957 , 959–961 ,
253 , 254 , 593 , 594 , 845 , 846 ,	963–966 , 1027 , 1028 , 1101 ,
1100 , 1101 , 1256 , 1257 , 1421 , 1422	1182 , 1257 , 1339 , 1422 , 1446–1450
<code>\glssubentrycounterfalse</code>	<code>\glstexorpdfstring</code> 173 , 173 , 174 ,
. 16 , 469 , 719 , 973 , 1128 , 1282	178 , 213 , 1061 , 1064 , 1086 ,
<code>\glssubentrycounterlabel</code>	1217 , 1219 , 1241 , 1374 , 1376 , 1398
. 254 , 254 , 593 ,	<code>\GLStext</code> 149 , 545 , 797 , 1050 , 1206 , 1363
594 , 845 , 846 , 1101 , 1257 , 1422	<code>\Glstext</code> 150 ,
<code>\glssubentryitem</code> 254 , 300 , 303–306 ,	150 , 545 , 797 , 1050 , 1206 , 1363
308 , 310 , 318 , 319 , 321 , 331 ,	<code>\glstext</code> 149 ,
333 , 335 , 339 , 340 , 342 , 404 ,	150 , 545 , 797 , 1050 , 1206 , 1363
408 , 409 , 411 , 413 , 594 , 669–	<code>\glstextaccessdisplay</code> 434 ,
678 , 686–690 , 694–696 , 701–	439 , 445 , 634 , 637 , 638 , 640 ,
703 , 705–707 , 709–712 , 846 ,	641 , 645 , 887 , 891 , 893 , 894 , 898
922–931 , 939–944 , 948–950 ,	<code>\glstextformat</code> 120 ,
955–957 , 959–961 , 963 , 964 ,	128 , 130 , 525 , 533 , 534 , 776 ,
966 , 1101 , 1257 , 1422 , 1446–1449	784 , 785 , 1033 , 1037 , 1038 ,
<code>\glssubgroupheading</code> 398	1188 , 1192 , 1193 , 1345 , 1349 , 1350
<code>\GLSsymbol</code>	<code>\glstextup</code> 50 , 266 , 459 ,
. 157 , 548 , 800 , 1053 , 1209 , 1366	486 , 599 , 656 , 736 , 851 , 909 ,

992, 1106, 1146, 1262, 1303, 1427
 \glstildechar . . . 61, 186, 190, 191,
 492, 561, 564, 742, 813, 815,
 816, 998, 1067, 1070, 1071,
 1153, 1223, 1226, 1309, 1380, 1383
 \glstranslatefalse
 32, 33, 478, 479, 728,
 982, 983, 1137, 1138, 1292, 1293
 \glstranslatetrue
 . . . 32, 33, 478, 479, 728, 729,
 982, 983, 1137, 1138, 1292, 1293
 \glstreechildpredesc . . . 407, 408,
 409, 709, 710, 962–964, 1446, 1447
 \glstreegroupheaderfmt . . . 325–
 330, 406, 408–411, 415, 698–
 701, 708–711, 713, 951–954,
 962–965, 967, 1442–1448, 1450
 \glstreeindent 409, 410,
 411, 413, 414, 673–675, 710–
 713, 927, 928, 964, 966, 1447–1450
 \glstreeitem 325, 326, 406,
406, 407, 697, 698, 708, 709,
 951, 952, 962, 1442, 1443, 1446
 \glstreenamebox 412, 413,
 414, 712, 713, 965, 966, 1449, 1450
 \glstreenamefmt
 406, 406, 407–410, 412–
 414, 708–713, 962–966, 1445–1450
 \glstreenavigationfmt
 326–330, 406, 408, 410,
 411, 415, 698–700, 708, 710,
 711, 713, 951–954, 962–965,
 967, 1442–1445, 1447, 1448, 1450
 \glstreepredesc . . . 407, 407, 409,
 410, 709–711, 962–964, 1446–1448
 \GlsTreeSetup 344
 \GlsTreeStarBox 345, 346
 \GlsTreeStarItemCounterBox
 346, 346, 347
 \GlsTreeStarNameBox 346, 347
 \GlsTreeStarOuterBox 346, 347
 \GlsTreeStarSubItemCounterBox
 346, 347, 348
 \GlsTreeStarSymbolBox 346, 347
 \glstreesubitem 325, 406,
 407, 697, 709, 951, 962, 1442, 1446
 \glstreesubsubitem 325, 406,
 407, 697, 709, 951, 962, 1442, 1446
 \GlsTreeUpdateWidestNameAndSymbol
 345

\GlsTreeUpdateWidestNameOrSymbol
 345
 \glstype 129, 130, 144–148, 165–172,
 443–445, 533, 542–545, 551–
 555, 643, 644, 784, 794–796,
 802–807, 896–898, 1037, 1038,
 1048–1050, 1056–1061, 1192,
 1193, 1203–1205, 1212–1216,
 1349, 1350, 1360–1362, 1369–1373
 \glsucmarkfalse
 . . . 15, 468, 718, 972, 1127, 1282
 \glsucmarktrue
 . . . 14, 468, 718, 972, 1127, 1281
 \glsunexpandedfieldvalue . . . 99,
 302, 764, 940, 1020, 1175, 1332
 \glsunset 112, 114–116, 127, 520–
 523, 542–545, 771–774, 794–
 797, 1028, 1029, 1031, 1036,
 1183–1186, 1191, 1340–1343, 1348
 \glsunsetall
 . . 114, 522, 772, 1029, 1184, 1341
 \glsupacrpluralsuffix
 50, 276, 282, 486, 605,
 610, 614, 617, 619, 622, 736,
 857, 862, 867, 870, 872, 875,
 992, 1111, 1112, 1117, 1146,
 1267, 1272, 1303, 1432, 1433, 1438
 \glsupercase . . . 5, 57, 96, 121–124,
 150–164, 166, 168, 170, 172,
 256, 257, 271, 272, 279–282,
 379–381, 423, 424, 438–442,
 444–450, 455–457, 459, 967,
 995, 1018, 1033–1035, 1050–
 1059, 1061, 1108, 1114–1116,
 1122, 1150, 1173, 1188–1190,
 1206–1216, 1264, 1269, 1270,
 1272, 1277, 1307, 1330, 1345–
 1347, 1363–1373, 1429, 1435–1437
 \GlsUseAcrEntryDispStyle
 273, 273,
 276–278, 280, 282, 283, 453–
 455, 457, 459, 460, 602–608,
 610, 650, 651, 654–656, 854–
 858, 860, 862, 903–905, 907,
 909, 1109, 1111–1113, 1115–
 1117, 1265, 1267–1269, 1271–
 1273, 1430, 1432–1434, 1436–1438
 \GlsUseAcrStyleDefs
 272, 273, 276–278,
 281–283, 453–455, 457, 459–

461, 602, 603, 605–608, 610, 611, 650, 651, 654–656, 854– 860, 862, 903–905, 907, 909, 910, 1109, 1111–1113, 1115– 1117, 1265, 1267–1269, 1271– 1273, 1430, 1432–1434, 1436–1438	<code>\glsuservi</code> 163, 163, 550, 802, 1056, 1211, 1368
<code>\GLSuseri</code> 159, 549, 800, 1054, 1210, 1367	<code>\glsuserviaccessdisplay</code> 437, 450, 635, 648, 889, 901
<code>\Glsuseri</code> 158, 158, 549, 800, 1054, 1209, 1366	<code>\GlsWarnAddProtectedPageFmt</code> 213, 213, 1085, 1241, 1398
<code>\glsuseri</code> 158, 158, 548, 800, 1054, 1209, 1366	<code>\glswrallowprimitivemodsfalse</code> 214, 1086, 1242, 1399
<code>\glsuseriaccessdisplay</code> 436, 448, 635, 647, 888, 900	<code>\glswrallowprimitivemodstrue</code> 579, 831
<code>\GLSuserii</code> 160, 549, 801, 1054, 1210, 1367	<code>\glswrglossdisableanchorcmds</code> 213, 217, 264, 1086, 1087, 1105, 1241, 1243, 1260, 1398, 1400, 1426
<code>\Glsuserii</code> 159, 159, 549, 801, 1054, 1210, 1367	<code>\glswrglossdisablelocationcmds</code> 207, 212, 212, 1083, 1085, 1239, 1240, 1395, 1397
<code>\glsuserii</code> 159, 159, 549, 800, 1054, 1210, 1367	<code>\glswrglosslocationtarget</code> 213, 264, 1086, 1105, 1241, 1260, 1398, 1426
<code>\glsuseriiaccessdisplay</code> 436, 449, 635, 647, 888, 900	<code>\glswrglosslocationtextfmt</code> 213, 264, 1086, 1105, 1241, 1260, 1398, 1426
<code>\GLSuseriii</code> 161, 549, 801, 1055, 1210, 1367	<code>\glswrite</code> 187–193, 201, 203, 209, 562–566, 572, 576, 577, 664–668, 814–818, 824, 828, 829, 918–921, 1068– 1072, 1079, 1084, 1224–1228, 1235, 1239, 1381–1385, 1391, 1396
<code>\Glsuseriii</code> 160, 160, 549, 801, 1055, 1210, 1367	<code>\glswritedefhook</code> 92, 100, 509, 514, 759, 765, 1015, 1021, 1170, 1176, 1327, 1333
<code>\glsuseriii</code> 160, 160, 549, 801, 1054, 1055, 1210, 1367	<code>\glswriteentry</code> 211, 211, 577, 829, 1084, 1085, 1240, 1397
<code>\glsuseriiiaccessdisplay</code> 436, 449, 635, 647, 888, 900	<code>\glswritefiles</code> 40, 209, 482, 576, 732, 828, 987, 988, 1083, 1142, 1239, 1298, 1396
<code>\GLSuseriv</code> 162, 550, 801, 1055, 1211, 1368	<code>\glsxindyfalse</code> 35, 480, 730, 984, 1139, 1294
<code>\Glsuseriv</code> 161, 161, 550, 801, 1055, 1211, 1368	<code>\glsxindytrue</code> 36, 480, 730, 984, 985, 1139, 1295
<code>\glsuseriv</code> 161, 161, 549, 801, 1055, 1210, 1211, 1367, 1368	<code>\glsxtrgetgrouptitle</code> 398, 399
<code>\glsuserivaccessdisplay</code> 437, 449, 635, 647, 888, 900, 901	<code>\glsxtrifemptyglossary</code> 1416
<code>\GLSuserv</code> 163, 550, 802, 1056, 1211, 1368	<code>\group</code> 229, 232, 399, 400, 405, 406, 1406, 1408
<code>\Glsuserv</code> 162, 162, 550, 802, 1055, 1211, 1368	
<code>\glsuserv</code> 162, 162, 550, 801, 1055, 1211, 1368	
<code>\glsuservaccessdisplay</code> 437, 450, 635, 647, 648, 889, 901	
<code>\GLSuservi</code> 164, 550, 802, 1056, 1212, 1369	
<code>\Glsuservi</code> 163, 163, 550, 802, 1056, 1211, 1368	
	H
	<code>\H</code> 28, 476, 725, 980, 1134, 1289
	<code>\hangindent</code> 329, 330, 399, 402, 404, 406, 409–

411, 413–415, 673–675, 700, 701, 708, 710–713, 926–928, 954, 962–964, 966, 967, 1444–1450	<code>\if@gl@docloaded</code> 6, 6, 20, 464, 471, 483, 714, 721, 733, 968, 975, 1123, 1130, 1277, 1285
<code>\hbox</code> 111, 305, 520, 670, 688, 771, 923, 941, 1028, 1183, 1340	<code>\if@gl@sisacronymlist</code> 23, 129, 473, 533, 722, 784, 977, 1037, 1131, 1192, 1286, 1349
<code>\hfill</code> 305, 670, 688, 923, 941	<code>\if@openright</code> 59, 490, 740, 996, 1151, 1308
<code>\hline</code> 307–311, 318–320, 322, 332–334, 336, 337, 339– 344, 689–691, 695–697, 701– 708, 942–944, 948–951, 955–962	<code>\if@org@KV@gl@slink@local</code> 126, 127, 1036, 1191, 1348
<code>\hsize</code> 305, 306, 317, 331, 338, 688, 694, 701, 705, 941, 942, 948, 954, 955, 958, 959	<code>\ifbool</code> 41, 71, 472, 479, 482, 498, 525–527, 721, 729, 732, 748, 776–778, 988, 1003, 1142, 1158, 1298, 1315
<code>\hspace</code> 406, 709, 962, 1446	<code>\ifboolexpr</code> 51, 78, 486, 501, 596, 736, 751, 848, 992, 1007, 1147, 1162, 1304, 1318
<code>\hss</code> 305, 670, 688, 923, 941	<code>\ifcase</code> 7, 9, 11, 32, 39, 42, 44, 84, 251, 407, 465–467, 478, 483, 504, 592, 673, 709, 714, 716, 717, 728, 733, 754, 843, 926, 963, 968, 970, 971, 982, 986, 988, 990, 1010, 1099, 1123, 1125, 1137, 1141, 1143, 1144, 1165, 1255, 1277, 1279, 1280, 1292, 1297, 1298, 1300, 1322, 1420, 1446
<code>\ht</code> 316, 693, 947	<code>\ifcsdef</code> 32, 52, 59, 87, 88, 95–100, 110, 211, 234, 235, 239, 244, 273, 433, 478, 487, 490, 506, 507, 511–514, 531, 577, 585, 587, 588, 597, 602, 633, 728, 737, 740, 756, 757, 761–765, 782, 829, 837, 839, 840, 849, 854, 886, 982, 993, 996, 1012, 1013, 1017–1021, 1027, 1084, 1092, 1094–1096, 1109, 1121, 1137, 1148, 1151, 1167, 1172–1175, 1182, 1240, 1248, 1250, 1251, 1265, 1292, 1304, 1308, 1324, 1329–1332, 1339, 1397, 1409, 1412, 1416, 1430
<code>\hyperbf</code> 264 , 599, 851, 1105, 1261, 1426	<code>\ifcsempty</code> 73, 419, 499, 680, 749, 933, 934, 1005, 1159, 1316
<code>\hyperdef</code> 48, 485, 735, 991, 1146, 1303	<code>\ifcsequal</code> 74, 499, 749, 1005, 1159, 1316
<code>\hyperemph</code> 265, 599, 851, 1105, 1261, 1426	<code>\ifcsname</code> . . . 99, 764, 1020, 1175, 1332
<code>hyperfirst</code> (option) 34	<code>\ifcsstrequal</code> 100, 514, 765, 1021, 1175, 1332
<code>\hyperit</code> 264 , 599, 851, 1105, 1261, 1426	
<code>\hyperlink</code> 126, 142, 264, 532, 541, 598, 783, 793, 850, 1036, 1047, 1105, 1191, 1202, 1260, 1348, 1359, 1426	
<code>\hypermd</code> 264 , 599, 851, 1105, 1261, 1426	
<code>\hypermm</code> 264 , 599, 850, 1105, 1261, 1426	
<code>\hypersc</code> 265 , 599, 851, 1105, 1261, 1426	
<code>\hypersf</code> 264 , 599, 851, 1105, 1261, 1426	
<code>\hypersl</code> 264 , 599, 851, 1105, 1261, 1426	
<code>\hypertarget</code> 141, 541, 793, 1047, 1201, 1358	
<code>\hypertt</code> 264 , 599, 851, 1105, 1261, 1426	
<code>\hyperup</code> 264 , 599, 851, 1105, 1261, 1426	
I	
<code>\IeC</code> 28, 475, 725, 979, 1134, 1289	
<code>\if</code> 132, 133, 217, 535, 580, 664, 786, 832, 917, 1039, 1040, 1087, 1194, 1243, 1351, 1400	
<code>\if@endfor</code> 684, 937, 1121	
<code>\if@endpe</code> . . . 224, 400, 406, 1247, 1404	
<code>\if@filesw</code> 36, 985, 1139, 1295	
<code>\if@gl@debug</code> 7, 24, 210, 465, 473, 577, 714, 723, 829, 968, 977, 1084, 1123, 1132, 1240, 1277, 1287, 1396	

<code>\ifcsstring</code>	1394, 1395, 1398, 1427, 1445, 1446
.. 99, 513, 764, 1020, 1175, 1332	<code>\ifdefempty</code> 23, 37, 38, 58, 69, 74–76,
<code>\ifcsundef</code>	80, 81, 120, 122, 200, 202, 244–
10, 35, 48, 51,	246, 271, 272, 278, 287, 437,
55–57, 59, 70, 71, 79, 81, 83,	440, 455, 473, 481, 490, 497,
102, 105, 115, 126, 131, 132,	499, 500, 502, 525, 528, 529,
142, 209, 221, 235, 248, 250,	572, 573, 589, 590, 601, 602,
255, 259–262, 265, 272, 434–	607, 612, 616, 618, 620, 636,
437, 467, 470, 480, 485, 486,	639, 641, 652, 723, 731, 740,
488–490, 497, 498, 501–503,	747, 749, 750, 752, 753, 776,
515, 517, 522, 532, 534, 541,	779, 780, 824, 825, 841, 842,
542, 570, 571, 576, 582, 584,	853, 854, 859, 865, 868, 870,
585, 591, 594, 596, 597, 599,	873, 889, 892, 894, 905, 977,
602, 634–636, 668, 684, 716,	986, 996, 1003, 1005, 1006,
720, 730, 735, 737, 739, 740,	1008, 1009, 1033, 1034, 1078–
748, 752, 754, 766, 768, 773,	1080, 1096, 1097, 1108, 1109,
783, 785, 793, 822, 828, 834,	1113, 1118, 1132, 1140, 1150,
836, 837, 843, 846, 848, 849,	1158, 1160, 1161, 1163, 1188,
851, 854, 887–889, 922, 938,	1189, 1234, 1235, 1252, 1253,
971, 974, 984, 991, 992, 994–	1263, 1264, 1269, 1274, 1286,
996, 1003, 1008, 1009, 1022,	1296, 1307, 1315, 1317, 1318,
1024, 1030, 1036, 1038, 1039,	1320, 1345, 1346, 1391, 1392,
1047, 1083, 1089, 1093, 1098,	1416, 1417, 1429, 1430, 1435, 1439
1099, 1101–1105, 1109, 1121,	<code>\ifdefequal</code>
1125, 1129, 1139, 1146, 1147,	36, 37, 73–76,
1149–1151, 1158, 1162–1164,	87–89, 93, 103, 107, 246, 434,
1177, 1179, 1185, 1191, 1193,	480, 481, 498–500, 506, 507,
1194, 1202, 1203, 1239, 1245,	509, 516, 519, 590, 634, 730,
1248, 1254, 1257–1259, 1261,	731, 749, 750, 756, 757, 760,
1264, 1280, 1294, 1302–1304,	767, 770, 842, 887, 985, 1004–
1306, 1308, 1315, 1319–1321,	1006, 1012, 1013, 1016, 1023,
1334, 1336, 1342, 1348, 1350,	1026, 1097, 1139, 1140, 1159–
1351, 1359, 1360, 1396, 1402,	1161, 1167, 1168, 1171, 1177,
1410, 1418–1420, 1422–1426, 1430	1180, 1253, 1295, 1316–1318,
<code>\ifcsvoid</code>	1324, 1325, 1328, 1334, 1337, 1417
. 75, 259, 1005, 1160, 1317, 1423	<code>\ifdefstrequal</code>
<code>\ifdef</code> 99, 513, 764, 1020, 1175, 1332
5, 45, 75, 76,	<code>\ifdefstring</code>
85, 91, 92, 111, 126, 173, 204,	13,
205, 213, 266, 302, 406, 420,	51, 78, 197–199, 202, 237, 238,
433, 500, 505, 508, 520, 532,	245, 247, 425, 468, 486, 501,
555, 557, 574, 575, 599, 633,	570–572, 587–591, 628, 718,
681, 708, 709, 750, 755, 759,	736, 751, 821–823, 825, 838,
771, 783, 807, 809, 826, 827,	839, 841, 842, 881, 972, 992,
851, 886, 934, 940, 962, 967,	1007, 1076–1078, 1080, 1094–
990, 1006, 1011, 1015, 1028,	1096, 1098, 1127, 1147, 1162,
1036, 1061, 1081, 1082, 1086,	1232–1234, 1236, 1250–1254,
1106, 1122, 1145, 1160, 1161,	1281, 1304, 1318, 1319, 1389,
1165, 1169, 1170, 1183, 1191,	1390, 1392, 1411, 1412, 1417, 1418
1217, 1237, 1238, 1241, 1262,	<code>\ifdefvoid</code>
1277, 1300, 1317, 1318, 1322,	26, 27,
1326, 1327, 1340, 1348, 1374,	36, 51, 106, 198, 199, 245, 246,
	475, 518, 589, 590, 724, 725,

769, 841, 842, 979, 984, 992, 1025, 1076, 1077, 1097, 1133, 1134, 1139, 1147, 1180, 1232, 1233, 1252, 1253, 1288, 1289, 1294, 1303, 1337, 1389, 1390, 1417	987, 990, 1080, 1141, 1142, 1145, 1235, 1296, 1297, 1300, 1392
\ifdim 275, 316, 412, 604, 693, 712, 856, 947, 965, 1111, 1266, 1432, 1449	\ifglsdescsuppressed <u>73</u> , 300, 499, 685, 749, 939, 1005, 1159, 1316
\iffalse 106, 113, 518, 521, 769, 772, 1025, 1029, 1180, 1184, 1337, 1341	\ifglsentrycounter 15, 16, 47, 253, 254, 469, 485, 593, 594, 718, 719, 735, 845, 846, 972, 973, 990, 1100, 1101, 1127, 1128, 1145, 1256, 1257, 1282, 1302, 1421, 1422
\IfFileExists 13, 31, 32, 199, 200, 225, 468, 478, 570, 571, 584, 718, 727, 728, 822, 823, 835, 972, 982, 1077–1079, 1091, 1126, 1136, 1137, 1233, 1234, 1247, 1281, 1292, 1390, 1391, 1404	\ifglsentryexists 71, <u>71</u> , 72, 90, 91, 101, 103, 497, 498, 508, 514, 516, 748, 758, 759, 765, 767, 1003, 1004, 1014, 1015, 1021, 1023, 1158, 1159, 1168–1170, 1176, 1178, 1315, 1325–1327, 1333, 1335
\ifglossaryexists 56, 67, <u>69</u> , 72, 197–199, 489, 496–498, 569–571, 739, 746–748, 821– 823, 994, 1002–1004, 1076– 1078, 1149, 1157–1159, 1231– 1233, 1306, 1314–1316, 1388–1390	\ifglshesclocations 133, 212, 213, 578, 830, 1040, 1085, 1086, 1195, 1240, 1241, 1352, 1397, 1398
\ifgl@sanitize@description 29, 476, 726, 980, 1135, 1290	\ifglshescseq <u>100</u> , 514, 765, 1021, 1175, 1332
\ifgl@sanitize@name 29, 476, 726, 980, 1135, 1290	\ifglshescddefeq <u>99</u> , 513, 764, 1020, 1175, 1332
\ifgl@sanitize@symbol 30, 476, 726, 981, 1135, 1290	\ifglshescfieldeq <u>99</u> , 513, 764, 1020, 1175, 1332
\ifgl@xindy@glnumbers 68, 497, 747, 1002, 1157, 1314	\ifglshescfieldvoid <u>75</u> , 1005, 1160, 1317
\ifgl@acr@description 34, 623, 877, 984, 1138, 1294	\ifglshescchildren <u>73</u> , 300, 498, 669, 686, 748, 922, 939, 1004, 1159, 1316
\ifgl@acr@dua 34, 614, 619, 622–624, 867, 872, 875, 877, 984, 1139, 1294	\ifglshescdesc <u>73</u> , 300, 499, 686, 749, 939, 1005, 1159, 1316
\ifgl@acr@footnote 34, 129, 533, 623, 624, 784, 877, 983, 1037, 1138, 1192, 1294, 1349	\ifglshescfield <u>75</u> , 500, 750, 1005, 1160, 1317
\ifgl@acr@synonym 21, 471, 472, 721, 976, 1130, 1131, 1285	\ifglshescfieldlong <u>74</u> , 117– 119, 174, 273–275, 279, 281, 452, 455, 458, 499, 524, 525, 556, 603, 604, 607, 609, 618, 649, 650, 652, 654, 749, 775, 776, 807, 855, 856, 859, 860, 871, 902, 903, 905, 907, 1005, 1032, 1033, 1062, 1109–1111, 1114, 1115, 1160, 1187, 1217, 1265, 1266, 1269, 1271, 1317, 1344, 1374, 1431, 1432, 1435, 1436
\ifgl@acr@shortcuts 47, 285, 484, 611, 735, 863, 990, 1118, 1145, 1273, 1302, 1439	\ifglshescparent ... <u>73</u> , 227, 235, 241, 245, 412, 499, 585, 589, 711, 749, 837, 841, 965, 1005, 1093, 1097, 1159, 1248, 1252, 1316, 1405, 1410, 1414, 1417, 1449
\ifgl@acr@smallcaps 34, 614, 615, 617, 619, 622, 866–868, 870, 872, 874, 875, 984, 1138, 1294	
\ifgl@acr@smaller 34, 614, 615, 618, 619, 866–868, 870, 872, 984, 1138, 1294	
\ifgl@automake <u>38</u> , 39, 45, 202, 481, 482, 573, 731, 732, 824, 986,	

<code>\ifglshasprefix</code>	419 , 420 , 421 , 423 , 680–683 , 933–936	594 , 719 , 735 , 845 , 846 , 973 , 990 , 1100 , 1101 , 1128 , 1145 , 1256 , 1257 , 1282 , 1302 , 1421 , 1422
<code>\ifglshasprefixfirst</code>	419 , 420 , 421 , 423 , 680–683 , 934–936	<code>\ifglstoc</code> 59 , 491 , 741 , 996 , 1151 , 1308
<code>\ifglshasprefixfirstplural</code>	419 , 421 , 422 , 424 , 680 , 681 , 683 , 684 , 934–937	<code>\ifglstranslate</code> 31 , 50 , 478 , 486 , 727 , 736 , 982 , 992 , 1136 , 1146 , 1292 , 1303
<code>\ifglshasprefixplural</code>	419 , 421 , 422 , 424 , 680–683 , 933 , 935–937	<code>\ifglsucmark</code> 57 , 489 , 739 , 740 , 995 , 1150 , 1307
<code>\ifglshasshort</code>	75 , 220 , 499 , 750 , 1005 , 1089 , 1160 , 1245 , 1317 , 1402	<code>\ifglused</code> 71 , 117 , 120–122 , 129 , 185 , 211 , 420–424 , 438– 440 , 498 , 524–529 , 533 , 561 , 577 , 613 , 616 , 618 , 620 , 636– 641 , 681–683 , 748 , 774 , 776– 780 , 784 , 812 , 829 , 865 , 868 , 871 , 873 , 889–894 , 934–937 , 1003 , 1031 , 1033 , 1034 , 1037 , 1067 , 1085 , 1158 , 1186 , 1188 , 1189 , 1192 , 1223 , 1240 , 1315 , 1343 , 1345 , 1346 , 1349 , 1380 , 1397
<code>\ifglshassymbol</code>	74 , 402 , 404 , 407–410 , 413 , 414 , 499 , 613 , 616 , 620 , 709–713 , 749 , 865 , 868 , 873 , 963 , 964 , 966 , 1005 , 1160 , 1317 , 1446–1450	<code>\ifglswrallowprimitivemods</code> 214 , 216 , 579 , 831 , 1086 , 1242 , 1399
<code>\ifglshyperfirst</code>	129 , 533 , 784 , 1037 , 1192 , 1349	<code>\ifglxindy</code> 16 , 35 , 53 , 54 , 60–63 , 65–68 , 80 , 107 , 108 , 134 , 186 , 187 , 193 , 197 , 198 , 200 , 217 , 218 , 225 , 480 , 488 , 491–497 , 502 , 519 , 536 , 561 , 562 , 566 , 569 , 570 , 580 , 581 , 584 , 663 , 664 , 730 , 738 , 741–747 , 752 , 770 , 787 , 813 , 814 , 818 , 821 , 822 , 831 , 833 , 836 , 916–918 , 973 , 984 , 994 , 997–1002 , 1008 , 1026 , 1040 , 1067 , 1068 , 1073 , 1076 , 1077 , 1079 , 1087 , 1088 , 1091 , 1128 , 1139 , 1148 , 1149 , 1151– 1157 , 1163 , 1180 , 1181 , 1195 , 1223 , 1224 , 1228 , 1232–1234 , 1243 , 1244 , 1247 , 1282 , 1294 , 1305 , 1306 , 1308–1314 , 1320 , 1337 , 1338 , 1352 , 1380 , 1381 , 1385 , 1389–1391 , 1399 , 1401 , 1404
<code>\ifglsindeonlyfirst</code>	211 , 577 , 829 , 1085 , 1240 , 1397	<code>\ifignoredglossary</code> 81 , 103 , 106 , 211 , 221 , 503 , 515 , 518 , 577 , 582 , 753 , 766 , 769 , 829 , 834 , 1009 , 1022 , 1025 , 1084 , 1090 , 1163 , 1177 , 1180 , 1240 , 1245 , 1320 , 1334 , 1337 , 1397 , 1402
<code>\ifglsnogroupskip</code>	303 , 307 , 308 , 310 , 313 , 314 , 318 , 319 , 321 , 331 , 333 , 335 , 339 , 341 , 343 , 408 , 409 , 411 , 414 , 687 , 689 , 690 , 692 , 695 , 696 , 701–703 , 705–707 , 709–711 , 713 , 940 , 942–946 , 948–950 , 955–957 , 959–961 , 963–965 , 967 , 1446–1448 , 1450	<code>\ifin@</code> 44 , 484 , 734 , 990 , 1144 , 1300
<code>\ifglsnonumberlist</code>	252 , 592 , 844 , 1100 , 1255 , 1420	<code>\ifinlistcs</code> 239 ,
<code>\ifglsnopostdot</code>	14 , 468 , 718 , 972 , 1127 , 1281	
<code>\ifglsnumberline</code>	59 , 491 , 741 , 996 , 1151 , 1308	
<code>\ifglsglresetcurrcount</code>	114 , 115 , 1030 , 1185 , 1342	
<code>\ifglsgl sanitizesort</code>	26 , 30 , 31 , 474 , 477 , 724 , 726 , 727 , 978 , 981 , 1133 , 1135 , 1136 , 1288 , 1291	
<code>\ifglsgl savenumberlist</code>	87 , 202 , 220 , 506 , 573 , 582 , 756 , 824 , 833 , 1012 , 1079 , 1089 , 1166 , 1235 , 1245 , 1323 , 1392 , 1402	
<code>\ifglsgl savewrites</code>	40 , 197 , 211 , 482 , 569 , 577 , 732 , 821 , 829 , 987 , 1076 , 1084 , 1142 , 1231 , 1240 , 1297 , 1388 , 1397	
<code>\ifglsgl subentrycounter</code>	16 , 48 , 253 , 254 , 469 , 485 , 593 ,	

248, 250, 588, 591, 840, 843,
 1095, 1098, 1251, 1254, 1412, 1419
`\ifinner` . 8, 465, 715, 969, 1124, 1278
`\ifKV@glslink@hyper`
 129–131, 533, 534, 784,
 785, 1038, 1192, 1193, 1349, 1350
`\ifKV@glslink@local` 126, 127, 542–
 545, 794–796, 1036, 1191, 1348
`\ifmeasuring@`
 . 109, 520, 771, 1027, 1181, 1338
`\ifmmode` 8,
 465, 715, 969, 1124, 1278, 1279
`\ifnum` 115, 116, 200, 245, 316, 409,
 411, 413, 470, 481, 523, 572,
 589, 673–675, 693, 710–712,
 719, 720, 732, 773, 774, 823,
 841, 927, 928, 947, 964, 966,
 974, 1030, 1031, 1078, 1096,
 1128, 1185, 1186, 1234, 1252,
 1342, 1343, 1391, 1417, 1447–1450
`\ifstrempty` 9, 84, 466, 669,
 716, 922, 970, 1124, 1279, 1321
`\ifstrequal` 596, 848
`\ifthenelse` 31,
 48, 59, 131, 201, 477, 485, 490,
 534, 572, 596, 624, 684, 727,
 735, 740, 785, 824, 848, 877,
 937, 981, 991, 996, 1038, 1079,
 1121, 1136, 1145, 1151, 1193,
 1235, 1291, 1302, 1308, 1350, 1392
`\IfTrackedLanguage`
 . 193, 566, 818, 1073, 1228, 1385
`\IfTrackedLanguageFileExists` . .
 52, 416, 487, 662,
 679, 737, 915, 932, 993, 1147, 1304
`\iftrue` 106, 112, 518, 521, 769, 772,
 1025, 1029, 1180, 1184, 1337, 1341
`\ifundef` 15, 78, 90, 102, 187, 191,
 201, 469, 501, 508, 515, 562,
 565, 572, 718, 719, 751, 758,
 766, 814, 817, 824, 972, 973,
 1007, 1014, 1022, 1068, 1071,
 1079, 1127, 1128, 1162, 1169,
 1177, 1224, 1227, 1235, 1282,
 1319, 1326, 1334, 1381, 1384, 1391
`\IfValueTF` 345
`\ifvmode` 184, 560, 812, 1067, 1222, 1379
`\ifvoid` 316, 694, 947
`\ifx` . . 15, 17, 22, 38, 43, 44, 48, 60,
 62, 64, 65, 68, 69, 77, 103–106,
 108–110, 131, 132, 135–140,
 175, 187, 190, 192–196, 209,
 213, 214, 216, 218–220, 223,
 260–263, 287, 289, 424, 434,
 469–473, 484, 485, 491, 492,
 494, 496, 497, 515–519, 534,
 536–541, 556, 562, 564, 566–
 568, 576, 578–581, 583, 596–
 598, 612, 614, 615, 617, 619–
 621, 623, 625, 627, 633, 664–
 669, 673–675, 718–720, 722,
 731, 734, 735, 741, 742, 744,
 745, 747, 766–770, 785, 787–
 793, 807, 808, 814, 816, 818–
 820, 828, 830–833, 835, 848–
 850, 864, 866, 868, 870, 872–
 874, 876, 878, 881, 887, 918,
 920–922, 926–928, 972–977,
 986, 989–991, 997, 998, 1000,
 1002, 1003, 1007, 1022–1027,
 1038, 1039, 1041–1046, 1062,
 1068, 1071–1073, 1075, 1083,
 1085–1091, 1103–1105, 1118,
 1127–1129, 1131, 1140, 1144,
 1146, 1152–1155, 1157, 1158,
 1161, 1177–1179, 1181, 1182,
 1193, 1194, 1196–1201, 1217,
 1218, 1224, 1226, 1228–1231,
 1239, 1241–1244, 1246, 1259,
 1260, 1274, 1282, 1283, 1286,
 1296, 1299, 1300, 1303, 1309–
 1312, 1314, 1315, 1318, 1334–
 1336, 1338, 1339, 1350, 1351,
 1353–1358, 1374, 1375, 1381,
 1383, 1385–1388, 1396, 1398–
 1401, 1403, 1424–1426, 1439, 1440
`\immediate` 37,
 38, 90, 92, 117, 197, 209, 225,
 481, 508, 509, 524, 569, 571,
 577, 584, 731, 758, 759, 774,
 775, 821, 823, 828, 829, 836,
 986, 1014, 1015, 1031, 1032,
 1076–1079, 1084, 1091, 1092,
 1140, 1141, 1169, 1170, 1186,
 1232–1234, 1239, 1247, 1296,
 1326, 1327, 1343, 1389, 1396, 1404
`\in@` 44, 484, 734, 990, 1144, 1300
`index` (option) [43](#)
`index` (style) [407](#)
`indexgroup` (style) [408](#)

<code>indexhypergroup</code> (style)	408	1016, 1017, 1171, 1172, 1328, 1329
<code>\indexname</code>	43, 484, 734, 989, 1144, 1299	<code>\keys</code> 12, 323, 344, 374
<code>indexonlyfirst</code> (option)	34	keys:
<code>\indexspace</code>	302, 303, 325, 325, 326–330, 406, 408–411, 414, 415, 686, 687, 697–701, 708–711, 713, 940, 951–954, 962–965, 967, 1442–1448, 1450	<code>access</code> 426
<code>inline</code> (style)	299	<code>counter</code> 83
<code>\input</code>	50, 120, 486, 525, 736, 776, 992, 1033, 1146, 1188, 1303, 1345	<code>description</code> 81
<code>\inputencodingname</code>	36, 480, 730, 984, 1139, 1294	<code>descriptionaccess</code> 426
<code>\InputIfFileExists</code>	91, 508, 759, 1015, 1169, 1326	<code>descriptionplural</code> 82
<code>\int</code>	17–19, 47, 228, 241, 242, 248, 249, 323, 348–354, 360, 361, 398–404, 1283, 1284, 1301, 1414, 1419	<code>descriptionpluralaccess</code> 427
<code>\istfile</code>	209, 576, 828	<code>first</code> 82
<code>\istfilename</code>	53, 53, 187, 191, 198, 199, 201, 488, 562, 565, 570–572, 664, 667, 738, 814, 817, 822–824, 918, 920, 994, 1068, 1071, 1077–1079, 1148, 1224, 1227, 1233–1235, 1305, 1381, 1384, 1389, 1391, 1392	<code>firstaccess</code> 426
<code>\item</code>	302–305, 325, 326, 407, 408, 669, 670, 672, 673, 687, 688, 697, 698, 709, 710, 922, 923, 926, 940–942, 951, 952, 962, 963, 1442, 1443, 1446, 1447	<code>firstplural</code> 82
J		
<code>\jobname</code>	37, 53, 90–92, 187, 191, 197–200, 225, 481, 488, 508, 562, 565, 569–571, 584, 664, 667, 731, 738, 758, 759, 814, 817, 821–823, 835, 918, 920, 986, 994, 1014, 1015, 1068, 1071, 1076–1079, 1091, 1140, 1148, 1169, 1170, 1224, 1227, 1232–1234, 1247, 1296, 1305, 1326, 1327, 1381, 1384, 1389–1391, 1404	<code>firstpluralaccess</code> 426
K		
<code>kernelglossredefs</code> (option)	42	<code>long</code> 86
<code>\key@ifundefined</code>	94, 95, 510, 760, 761,	<code>longaccess</code> 427
		<code>longplural</code> 86
		<code>longpluralaccess</code> 427
		<code>name</code> 81
		<code>nonumberlist</code> 84
		<code>parent</code> 84
		<code>plural</code> 82
		<code>pluralaccess</code> 426
		<code>see</code> 83
		<code>short</code> 86
		<code>shortaccess</code> 427
		<code>shortplural</code> 86
		<code>shortpluralaccess</code> 427
		<code>sort</code> 82
		<code>symbol</code> 83
		<code>symbolaccess</code> 426
		<code>symbolplural</code> 83
		<code>symbolpluralaccess</code> 426
		<code>text</code> 82
		<code>textaccess</code> 426
		<code>type</code> 83
		<code>user1</code> 85
		<code>user1access</code> 427
		<code>user2</code> 85
		<code>user2access</code> 427
		<code>user3</code> 85
		<code>user3access</code> 427
		<code>user4</code> 86
		<code>user4access</code> 427
		<code>user5</code> 86
		<code>user5access</code> 428
		<code>user6</code> 86
		<code>user6access</code> 428
		<code>\KV@glslink@hyperfalse</code> 126, 129, 142, 532, 533, 542, 783,

784, 793, 1036–1038, 1047,
1191, 1192, 1203, 1348, 1349, 1360
\KV@glslink@hypertrue
126, 143, 532, 542, 783, 793,
1036, 1047, 1191, 1203, 1348, 1360

L

\L 28, 476, 725, 980, 1134, 1290
\l 5, 12, 27, 29,
30, 46, 110, 226–232, 240–243,
248, 323–325, 348–357, 359–
362, 364–378, 385–389, 392–
406, 476, 725, 968, 980, 1122,
1134, 1182, 1277, 1290, 1301,
1339, 1404–1408, 1413–1415, 1419
\label 11, 109–
111, 251, 253, 254, 467, 592,
593, 717, 844, 845, 971, 1027,
1028, 1099, 1100, 1126, 1182,
1255, 1256, 1280, 1339, 1420, 1421
\language
. . . 35, 480, 730, 984, 1139, 1294
languages (option) 33
\leaders 305, 670, 688, 923, 941
\leavevmode
100, 130, 514, 533, 765, 784,
1021, 1038, 1176, 1193, 1333, 1350
\legacy 47, 398, 401, 1301
\let 6, 7, 14, 17–
19, 21, 28, 29, 31–34, 36–38,
40–44, 47, 48, 51, 53, 62, 63,
76, 77, 87, 89, 100–106, 109–
114, 116, 120, 126–128, 130,
131, 133, 141–149, 164–172,
175, 182, 183, 191, 193, 196,
198, 199, 201–206, 211–213,
215, 216, 219–224, 246, 250,
253, 271, 283–285, 298, 302,
316, 325, 326, 406, 407, 415–
418, 428, 429, 443–445, 464,
465, 468, 470–472, 475, 476,
478, 479, 481–486, 488, 492,
493, 500, 501, 506, 507, 514–
518, 520–523, 525, 532–536,
541–545, 551–556, 559, 560,
565, 566, 569–575, 577–579,
581–583, 590, 591, 593, 595,
601, 611–623, 630, 643, 644,
657–662, 668, 679, 680, 684,
685, 693, 697, 698, 709, 714,
715, 718–721, 725–729, 731–
738, 743, 750, 751, 756–758,
765–769, 771–774, 776, 783–
787, 793–797, 802–808, 811,
812, 816, 818, 821–827, 829–
831, 833–835, 842, 843, 845,
847, 853, 863, 864, 866–874,
876, 883, 884, 896–898, 910–
915, 921, 932, 933, 937, 938,
940, 947, 951, 952, 962, 968,
969, 972–976, 979, 980, 982–
993, 998, 999, 1006, 1007,
1012–1014, 1021–1025, 1027–
1031, 1033, 1036–1038, 1040,
1047–1050, 1056–1062, 1066,
1071, 1072, 1076–1086, 1089–
1091, 1097–1100, 1108, 1117,
1118, 1121–1123, 1127–1130,
1134–1148, 1153, 1154, 1161,
1167, 1168, 1176–1180, 1182–
1188, 1191–1193, 1195, 1202–
1206, 1212–1216, 1218, 1221,
1222, 1227, 1228, 1231–1238,
1240–1242, 1244–1247, 1253,
1254, 1256, 1263, 1273, 1277,
1278, 1281, 1283–1285, 1289,
1290, 1292–1300, 1302, 1304,
1305, 1310, 1311, 1317, 1318,
1324, 1325, 1333–1337, 1339–
1345, 1348–1350, 1352, 1359–
1363, 1369–1373, 1375, 1378,
1379, 1384, 1385, 1388–1390,
1392–1395, 1397–1399, 1401–
1404, 1417, 1419, 1421, 1429,
1438, 1439, 1442, 1443, 1446
\letcs 36, 38,
73–76, 92, 94, 95, 98, 104, 105,
172–174, 204, 205, 234, 236,
242, 245, 246, 256, 259, 260,
412, 498–500, 508, 510, 513,
517, 555, 556, 570, 571, 574,
575, 585, 586, 589, 590, 594,
596, 712, 731, 749, 750, 759,
761, 764, 767, 768, 807, 822,
823, 826, 827, 837, 838, 841,
842, 846, 848, 965, 984, 986,
1004–1006, 1015, 1017, 1020,
1023, 1024, 1061, 1062, 1081,
1082, 1092–1094, 1097, 1101–
1103, 1121, 1139, 1141, 1159–

`\makenoidxglossaries` . . . 9, 10, 36,
37, 84, 202, [203](#), 206, 207, 252,
466, 480, 481, 504, 573–576,
716, 730, 731, 754, 824–828,
970, 985, 1010, 1080–1083,
1125, 1139, 1140, 1165, 1235–
1238, 1279, 1295, 1322, 1392–1395
`\MakeTextUppercase`
. 5, 464, 714, 968, 1122, 1277
`\MakeUppercase`
637, 638, 644, 890–892, 897, 898
`\marginpar` 8, 466, 715, 969, 1124, 1279
`\markboth` 57, 489, 739, 995, 1150, 1307
`\mbox` 184, 304, 329, 330,
412, 560, 670, 687, 700, 712,
812, 923, 941, 953, 954, 966,
1067, 1222, 1379, 1444, 1445, 1449
`mcolalmtree` (style) [329](#)
`mcolalmtreegroup` (style) [329](#)
`mcolalmtreehypergroup` (style) . . . [329](#)
`mcolalmtreeespannav` (style) [330](#)
`mcolindex` (style) [325](#)
`mcolindexgroup` (style) [325](#)
`mcolindexhypergroup` (style) [325](#)
`mcolindexspannav` (style) [326](#)
`mcoltree` (style) [326](#)
`mcoltree*` (style) [323](#)
`mcoltreegroup` (style) [326](#)
`mcoltreehypergroup` (style) [327](#)
`mcoltreenoname` (style) [327](#)
`mcoltreenonamegroup` (style) [328](#)
`mcoltreenonamehypergroup` (style) [328](#)
`mcoltreenonamespannav` (style) . . . [328](#)
`mcoltreespannav` (style) [327](#)
`\memUchead` 57, 489, 739, 995, 1150, 1307
`\MessageBreak` 24, 41, 78,
223, 252, 416, 417, 424, 425,
474, 482, 483, 501, 583, 627,
662, 679, 723, 732, 733, 751,
835, 881, 915, 916, 932, 978,
988, 1007, 1090, 1132, 1143,
1162, 1246, 1287, 1298, 1319, 1403
`mfirstuc` (option) [44](#)
`\mfirstucMakeUppercase` [5](#),
464, 489, 512, 526–530, 545–
555, 602, 607–609, 616, 620,
640–642, 645–648, 652, 653,
655, 683, 684, 714, 739, 740,
762, 777–781, 797–805, 807,
853, 854, 859–861, 869, 873,
893–895, 898–901, 905, 906,
908, 936, 937, 968, 1122, 1277
`\MFUaddmap` 5, 967, 1122, 1277
`\MFUblocker` 5, 967, 1122, 1277
`\MFUexcl` 5, 967, 1122, 1277
`\MFUsentencecase` 5,
5, 44, 173, 174, 178, 967, 968,
990, 1061, 1064, 1122, 1144,
1217, 1219, 1277, 1300, 1374, 1376
`\midrule` 313, 314, 692, 945, 946
`\month` 187, 191, 562, 565,
664, 667, 814, 817, 918, 920,
1068, 1071, 1224, 1227, 1381, 1384
`\msg` 226, 227, 1405

N

`\n` 192, 565, 667, 668, 817,
920, 921, 1071, 1072, 1227, 1384
`name` (key) [81](#)
`\NeedsTeXFormat` . . . 4, 295, 299, 301,
305, 312, 317, 322, 330, 337,
344, 415–417, 424, 463, 464,
627, 662, 663, 668, 679, 714,
880, 915, 916, 922, 932, 940,
967, 1121, 1122, 1276, 1442, 1445
`\new@glossaryentry` 89, [90](#), 204, 507,
508, 574, 758, 826, 1014, 1081,
1168, 1169, 1237, 1325, 1326, 1394
`\new@ifnextchar` . . . 78, 95, 96, 117–
119, 143–147, 149–171, 268–
270, 420–423, 501, 511, 512,
524, 542–555, 599, 600, 681–
683, 751, 762, 775, 794–806,
851, 852, 934–936, 1007, 1018,
1032, 1047–1060, 1106, 1107,
1162, 1172, 1173, 1187, 1203–
1216, 1262, 1263, 1319, 1329,
1330, 1344, 1360–1373, 1427, 1428
`\newacronym` 265, 266,
[266](#), 271, 599, 601, 612, 614,
615, 617, 619, 621, 623, 625,
851, 853, 864, 866, 868, 870,
872, 874, 876, 878, 1105, 1106,
1108, 1261, 1263, 1426, 1427, 1429
`\newacronymhook` . . . 271, [271](#), 461,
601, 612, 614, 616, 617, 619,
622, 623, 625, 657, 853, 864,
866, 868, 870, 872, 875, 876,
878, 910, 1108, 1263, 1264, 1429

`\newacronymstyle` 273,
273, 274–278, 280–283, 602–
608, 610, 854–860, 862, 1109–
1113, 1115–1117, 1265–1269,
1271–1273, 1430–1434, 1436–1438
`\newcommand` . 4–6, 8–35, 37–45, 47–
50, 53–63, 65–75, 77–82, 84–
96, 99–101, 106–120, 122, 124–
129, 131, 133, 140–187, 193,
196–200, 202, 203, 206, 207,
209–213, 215, 217–222, 224,
234–239, 241, 244, 245, 247,
248, 251–262, 264–273, 275,
283, 285, 286, 288–302, 316,
325, 345, 346, 406, 407, 412,
418–423, 425, 430–434, 451,
461–463, 465–475, 477–499,
501–514, 518–525, 528, 529,
531–536, 541–562, 566, 569–
573, 575–582, 584–597, 599–
604, 611–628, 632–634, 649,
657, 661, 662, 668, 680–687,
693, 697, 708, 709, 712, 715–
725, 727–729, 731–736, 738–
743, 745–749, 751–765, 769–
776, 779–787, 793–813, 818,
821–825, 827–856, 863–881,
885–887, 902, 910, 914, 915,
922, 933–940, 947, 951, 962,
965, 967–1018, 1020, 1021,
1025–1034, 1036–1038, 1040,
1047–1068, 1073, 1075–1080,
1083–1109, 1111, 1117–1122,
1124–1146, 1148–1173, 1175,
1176, 1180–1193, 1195, 1201–
1224, 1228, 1231–1234, 1236,
1238–1259, 1261–1266, 1273–
1294, 1296–1300, 1302, 1303,
1305–1330, 1332, 1333, 1337–
1350, 1352, 1358–1381, 1385,
1388–1393, 1395–1404, 1409–
1413, 1416–1418, 1420–1430,
1432, 1438–1442, 1445, 1446, 1449
`\newcount` 87, 470, 506, 720,
756, 974, 1012, 1129, 1167, 1324
`\newcounter` 15, 16, 469, 718,
719, 972, 973, 1127, 1128, 1282
`\NewDocumentCommand` 141,
209, 323, 344, 345, 412, 1202, 1359
`\newenvironment`
. 255, 594, 846, 1101, 1257, 1422
`\newglossary` 20–22, 43,
78, 79, 80, 201, 471, 472, 483,
484, 501, 502, 572, 721, 722,
734, 751, 752, 824, 975, 976,
989, 1007, 1008, 1079, 1130,
1131, 1143, 1144, 1162, 1163,
1235, 1285, 1299, 1319, 1320, 1392
`\newglossaryentry` 9, 43, 86, 89, 89,
115, 271, 466, 484, 505, 507,
522, 601, 611, 613, 614, 616,
618, 620, 622, 624, 657–660,
716, 734, 755–758, 773, 853,
863, 865, 867, 869, 871, 873,
875, 877, 910–913, 970, 989,
1011–1014, 1030, 1108, 1125,
1144, 1166, 1168, 1185, 1264,
1279, 1299, 1323, 1325, 1342, 1429
`\newglossarystyle`
. 261, 299, 302–315, 317–
322, 324–343, 398, 407–412,
414, 415, 597, 685, 687–713,
849, 938, 940–946, 948–965,
967, 1103, 1259, 1424, 1442–1450
`\newif` 6, 7, 23, 31, 35, 38, 114, 214,
464, 465, 473, 478, 480, 481,
579, 714, 722, 727, 730, 731,
831, 968, 977, 982, 984, 986,
1030, 1086, 1123, 1131, 1136,
1139, 1141, 1185, 1242, 1277,
1286, 1292, 1294, 1296, 1342, 1399
`\newignoredglossary`
. 80, 502, 752, 1008, 1163, 1320
`\newlabel` 141, 1202, 1359
`\newlength` 140, 305, 306,
317, 330, 331, 338, 410, 541,
688, 694, 701, 705, 710, 793,
941, 942, 948, 954, 955, 958,
959, 964, 1047, 1201, 1358, 1448
`\newrobustcmd` 9, 73–75, 89,
90, 96–99, 117–119, 128, 143–
171, 173, 174, 178, 184, 185,
219, 220, 259, 262, 266–270,
287, 411, 420–423, 434–437,
466, 498–500, 507, 508, 511,
512, 524, 532, 542–561, 581,
582, 599, 600, 634–636, 680–
683, 711, 715, 716, 748–750,
757, 758, 762, 775, 783, 794–
812, 833, 851, 852, 887–889,

933–936, 965, 970, 1004, 1005,	1179, 1180, 1182, 1193–1196,
1013, 1014, 1018–1020, 1032,	1201, 1222, 1228–1235, 1241,
1037, 1047–1062, 1064, 1067,	1243, 1245, 1247, 1249, 1260,
1089, 1102, 1104, 1106, 1107,	1261, 1264, 1286, 1303, 1309,
1118, 1124, 1159, 1160, 1168,	1310, 1326, 1336, 1337, 1339,
1169, 1172–1175, 1186, 1187,	1350–1353, 1358, 1379, 1385–
1192, 1203–1217, 1219, 1222,	1388, 1392, 1398, 1400, 1402,
1223, 1244, 1245, 1258, 1259,	1404, 1410, 1411, 1426, 1427, 1429
1262, 1263, 1274, 1279, 1316,	<code>noglossaryindex</code> (option) 43
1317, 1325, 1326, 1329–1332,	<code>nogroupskip</code> (option) 14
1343, 1344, 1349, 1360–1374,	<code>\nohyperpage</code> 262, 598,
1376, 1379, 1380, 1401, 1402,	849, 850, 1104, 1259, 1260, 1425
1423, 1425, 1427, 1428, 1439, 1448	<code>nohypertypes</code> (option) 24
<code>\newterm</code> 43, 484, 734, 989, 1144, 1299	<code>\noindent</code> 326–
<code>\newtoks</code> 134, 197, 270,	328, 330, 409–411, 595, 698–
536, 569, 601, 787, 821, 853,	700, 710, 711, 847, 952–954,
1041, 1076, 1107, 1108, 1196,	964, 965, 1442–1445, 1447, 1448
1231, 1263, 1353, 1388, 1428, 1429	<code>\noist</code> 191, 193, 196,
<code>\newwrite</code> 37, 90,	196, 569, 667, 668, 821, 920,
187, 191, 197, 201, 481, 508,	921, 1071, 1072, 1075, 1076,
562, 565, 569, 572, 731, 758,	1227, 1228, 1231, 1384, 1385, 1388
814, 817, 821, 824, 986, 1014,	<code>nolangwarn</code> (option) 25
1068, 1071, 1076, 1079, 1140,	<code>nolist</code> (option) 13
1169, 1224, 1227, 1231, 1235,	<code>nolong</code> (option) 13
1296, 1326, 1381, 1384, 1388, 1391	<code>nomain</code> (option) 21
<code>\nfss@text</code> 8, 465, 715, 969, 1124, 1279	<code>nonumberlist</code> (key) 84
<code>\noalign</code> 316, 693, 947	<code>nonumberlist</code> (option) 12
<code>\nobreak</code> 304,	<code>\nopagebreak</code> 400
316, 670, 687, 694, 923, 941, 947	<code>\nopostdesc</code> . . 44, 53, 53 , 100, 223,
<code>\noexpand</code> 22, 49, 61,	484, 487, 488, 514, 583, 669,
62, 91, 105–107, 110, 131–134,	734, 738, 765, 835, 922, 989,
140, 182, 193–196, 202, 212,	993, 1021, 1091, 1144, 1148,
218, 221, 225, 236, 264, 265,	1176, 1246, 1299, 1305, 1333, 1403
271, 472, 485, 491, 492, 508,	<code>nopostdot</code> (option) 14
517, 518, 531, 532, 534–536,	<code>noredefwarn</code> (option) 25
541, 560, 566–571, 573, 578,	<code>\normalbaselineskip</code> . . . 316, 693, 947
579, 582, 584, 586, 594, 595,	<code>nostyles</code> (option) 14
599, 601, 611, 613–618, 620–	<code>nosuper</code> (option) 13
622, 624, 628, 657–661, 663,	<code>notranslate</code> (option) 32
722, 736, 742, 759, 768, 769,	<code>notree</code> (option) 14
782, 785–787, 792, 793, 811,	<code>nowarn</code> (option) 24
818–824, 830, 831, 834, 836,	<code>\ns@ACRfull</code>
838, 846, 847, 851, 853, 863,	. 269, 600, 852, 1107, 1262, 1428
865–867, 869, 871, 873–875,	<code>\ns@Acrfull</code> 268,
877, 878, 881, 910–914, 916,	269, 600, 851, 1107, 1262, 1428
976, 991, 997, 998, 1015, 1024,	<code>\ns@acrfull</code>
1025, 1027, 1038–1041, 1046,	. 268, 599, 851, 1106, 1262, 1427
1066, 1073–1080, 1085, 1088,	<code>\ns@ACRfullpl</code>
1089, 1091–1094, 1105, 1108,	. 270, 600, 852, 1107, 1263, 1428
1131, 1146, 1152, 1153, 1169,	

1395, 1409, 1411–1413, 1416,
 1420, 1421, 1424, 1425, 1427, 1430
 \PackageInfo . . . 7, 8, 37, 197, 210,
 465, 481, 569, 577, 714, 715,
 730, 731, 821, 829, 969, 985,
 1076, 1084, 1123, 1124, 1140,
 1232, 1240, 1278, 1295, 1389, 1396
 \PackageWarning 7,
 8, 24, 44, 47, 465, 473, 714,
 715, 723, 968, 969, 977, 1123,
 1124, 1132, 1278, 1287, 1300, 1301
 \PackageWarningNoLine
 7, 8, 24, 416, 465,
 473, 474, 662, 679, 714, 715,
 723, 915, 932, 968, 969, 977,
 978, 1123, 1124, 1132, 1278, 1287
 \pagegoal 316, 693, 947
 \pagelistname 50, 52, 309–
 311, 313, 314, 320, 322, 334–
 337, 341–344, 486, 487, 690–
 692, 696, 697, 703–705, 707,
 708, 736, 737, 943, 944, 946,
 949–951, 956–958, 960–962,
 991, 993, 1146, 1148, 1303, 1305
 \par 53, 302,
 304, 325, 327–330, 376–378,
 398–400, 402, 404, 406, 407,
 409–415, 488, 595, 596, 670,
 673–675, 687, 697–701, 708–
 713, 738, 847, 848, 923, 927,
 928, 940, 941, 951–954, 962–
 967, 993, 1148, 1305, 1442–1450
 parent (key) 84
 \parindent 325–330,
 399, 401–403, 405, 407, 409–
 411, 413–415, 673–675, 697–
 701, 709–713, 926–928, 951–
 954, 962–964, 966, 967, 1442–1450
 \parskip 325–
 328, 399, 401, 407, 409, 410,
 697–699, 709–711, 951–953,
 962–964, 1442–1444, 1446–1448
 \PassOptionsToPackage
 417, 424, 627, 679, 880, 932
 \pdfbookmark 346
 \pdfstringdefPreHook
 213, 1086, 1241, 1398
 \penalty 316, 693, 947
 \PGLS 423, 683, 936
 \PglS 421, 421, 682, 935
 \pgls 420, 421, 681, 934
 \PGLSpl 423, 683, 936
 \PglSpl 422, 422, 682, 935
 \pglSpl 420, 422, 681, 934
 \phantomsection
 58, 490, 740, 996, 1150, 1307
 plural (key) 82
 pluralaccess (key) 426
 preprocess-sort (option) 30
 \prg 70, 74, 249, 250
 \printacronyms
 21, 472, 721, 976, 1131, 1285
 \PrintChanges
 6, 464, 714, 968, 1123, 1277
 \printglossaries
 202, 222, 573, 582, 825, 834,
 1080, 1090, 1236, 1245, 1393, 1402
 \printglossary
 21, 25, 43, 202, 221, 222,
 252, 472, 474, 483, 484, 573,
 582, 592, 722, 724, 734, 825,
 834, 844, 976, 978, 989, 1080,
 1089, 1090, 1100, 1131, 1133,
 1143, 1144, 1236, 1245, 1255,
 1285, 1287, 1299, 1393, 1402, 1421
 \printindex
 43, 484, 734, 989, 1144, 1299
 \printnoidxglossaries
 204, 222, 574, 582, 826, 834,
 1081, 1090, 1236, 1246, 1393, 1403
 \printnoidxglossary
 203, 204, 207, 208,
 222, 222, 237, 238, 252, 573,
 574, 576, 582, 587, 588, 592,
 825, 826, 828, 834, 839, 840,
 844, 1080, 1081, 1083, 1090,
 1094, 1095, 1100, 1236, 1238,
 1246, 1250, 1251, 1255, 1393,
 1395, 1402, 1403, 1411, 1412, 1421
 \printnumbers
 43, 484, 734, 989, 1144, 1299
 \printsymbols
 43, 483, 734, 989, 1143, 1299
 \ProcessOptions
 417, 424, 627, 679, 880, 932
 \ProcessOptionsX
 44, 484, 734, 990, 1145, 1300
 \prop 226–229, 1405, 1406
 \protect 59, 125, 174,
 274, 275, 281, 297, 442, 443,

452, 453, 458, 491, 531, 603,
604, 609, 613, 616, 618, 643,
649, 650, 654, 741, 781, 782,
855, 856, 861, 865, 869, 871,
896, 902, 903, 907, 997, 1036,
1061, 1110, 1111, 1115, 1116,
1121, 1151, 1190, 1191, 1217,
1265–1267, 1271, 1308, 1347,
1348, 1374, 1431, 1432, 1436, 1437
\protected@csedef
. . . 97, 512, 763, 1019, 1174, 1331
\protected@csxappto 1121
\protected@csxdef 97,
512, 763, 1019, 1121, 1173, 1330
\protected@eappto
. 37, 81, 107, 212, 731,
752, 753, 770, 830, 986, 1008,
1009, 1026, 1085, 1140, 1163,
1180, 1241, 1296, 1320, 1337, 1398
\protected@edef
. . . . 9, 11, 22, 37, 62, 64, 66,
68, 69, 73, 77, 79–81, 97–103,
107, 108, 125, 130–132, 174,
182, 184, 200, 202, 203, 211,
214, 216, 217, 221, 234, 235,
251, 254, 261, 264, 271, 287,
295, 433, 434, 461, 466, 467,
492, 493, 495, 497, 516, 518,
519, 525–527, 531, 534, 535,
560, 577–579, 592, 594, 595,
597, 601, 620, 624, 628, 633,
634, 664, 665, 680, 684, 715,
717, 720, 722, 731, 742, 744,
746, 747, 749, 751–753, 763–
767, 769, 770, 776–778, 782,
784–786, 811, 812, 825, 829–
831, 834, 836, 837, 844–847,
849, 853, 873, 877, 881, 886,
887, 910, 917, 918, 933, 937,
970, 971, 974–976, 986, 998,
999, 1001, 1003, 1004, 1007–
1009, 1019–1023, 1025, 1026,
1036, 1038, 1039, 1062, 1066,
1067, 1078, 1080, 1081, 1084,
1086, 1087, 1089, 1092, 1093,
1099, 1100, 1103, 1105, 1108,
1118, 1121, 1124, 1126, 1129,
1131, 1140, 1153, 1154, 1156–
1159, 1161–1163, 1173–1177,
1180, 1181, 1190, 1191, 1193,
1194, 1217, 1221–1223, 1234–
1236, 1240, 1242, 1243, 1245,
1248, 1255, 1256, 1259, 1260,
1264, 1274, 1279, 1280, 1286,
1296, 1310, 1311, 1313–1316,
1318–1320, 1330–1334, 1337,
1338, 1347, 1348, 1350, 1351,
1374, 1378–1380, 1391–1393,
1397–1400, 1402, 1409, 1410,
1420, 1421, 1424, 1426, 1429, 1439
\protected@write
77, 79, 141, 187–189, 201, 203,
207, 211, 221, 297, 501, 562,
563, 572, 573, 576, 577, 582,
665, 684, 751, 752, 814, 815,
824, 825, 828, 829, 834, 918,
937, 1007, 1008, 1069, 1070,
1079, 1081, 1083, 1084, 1089,
1121, 1161, 1162, 1202, 1224,
1225, 1235, 1236, 1239, 1240,
1245, 1318, 1319, 1359, 1381,
1382, 1391–1393, 1395, 1397, 1402
\protected@xdef
22, 28, 88, 103, 106, 108, 109,
216, 224, 429, 430, 470–472,
475, 506, 519, 520, 579, 630–
632, 719, 720, 722, 725, 757,
767, 769–771, 831, 835, 884,
885, 974–976, 979, 1013, 1023,
1025–1027, 1087, 1091, 1128,
1129, 1131, 1134, 1167, 1177–
1181, 1242, 1246, 1247, 1286,
1289, 1324, 1334–1338, 1399, 1403
\providecommand 5, 21, 36, 49, 50,
57, 77, 117, 143, 189, 201, 204,
225, 302, 325, 406, 419, 425,
472, 480, 486, 489, 501, 524,
542, 563, 572, 574, 576, 584,
594, 595, 628, 681, 686, 697,
708, 721, 730, 736, 740, 751,
774, 794, 815, 824, 826, 828,
836, 846, 847, 881, 934, 940,
951, 962, 968, 976, 985, 991,
992, 995, 1007, 1031, 1047,
1070, 1079, 1081, 1091, 1092,
1122, 1131, 1139, 1146, 1150,
1161, 1186, 1203, 1225, 1235,
1236, 1247, 1277, 1285, 1295,
1303, 1307, 1318, 1343, 1360,
1382, 1391–1393, 1404, 1442, 1445

<code>\provideglossaryentry</code>	630, 643, 644, 657–660, 664, 666–668, 673–675, 684, 687, 693, 697, 700, 701, 708–714, 717, 718, 720, 721, 728, 731–735, 743, 751, 753, 754, 757, 759, 767, 769, 771, 773, 774, 783, 785–792, 807, 808, 816–818, 820, 821, 823–827, 830–833, 835, 841, 843, 848, 849, 878, 883, 884, 896, 897, 910–912, 914, 917, 919–921, 926–928, 937, 940, 947, 951, 954, 962–968, 971, 972, 974–976, 982, 983, 986, 988–991, 999, 1007, 1009, 1010, 1013, 1015, 1023, 1025, 1028, 1030, 1031, 1037, 1039–1046, 1062, 1070–1073, 1075, 1076, 1078, 1079, 1081, 1082, 1085, 1087–1091, 1096–1099, 1103, 1118, 1121, 1123, 1125, 1127, 1129, 1130, 1137, 1138, 1140, 1141, 1143–1145, 1154, 1161, 1164, 1165, 1168, 1169, 1177–1179, 1183, 1185, 1186, 1191–1201, 1217, 1218, 1226–1231, 1234–1238, 1240, 1242–1244, 1246, 1247, 1252, 1254, 1255, 1258, 1259, 1274, 1277, 1280, 1281, 1285, 1292, 1293, 1296–1299, 1301, 1302, 1311, 1318, 1321, 1322, 1325, 1326, 1334–1336, 1340, 1342, 1343, 1348–1358, 1374, 1375, 1383–1388, 1391–1393, 1395, 1397, 1399–1401, 1403, 1404, 1416, 1417, 1419, 1420, 1423, 1424, 1440, 1442, 1444–1450
. . . 89 , 508, 758, 1014, 1168, 1325	
<code>\ProvidesFile</code>	
. . . 50 , 486, 736, 992, 1146, 1303	
<code>\ProvidesGlossariesLang</code>	
. . . 50 , 486, 736, 992, 1146, 1303	
<code>\ProvidesPackage</code> 4 , 295, 299, 301, 306, 312, 317, 323, 330, 338, 344, 415–417, 424, 463, 464, 627, 662, 663, 668, 679, 684–686, 688, 692, 694, 697, 701, 705, 708, 714, 880, 915, 916, 922, 932, 937, 938, 940, 942, 945, 948, 951, 954, 958, 962, 967, 1121, 1122, 1276, 1442, 1445	
Q	
<code>\quark</code>	227, 1405
R	
<code>\r</code>	28 , 475, 725, 979, 1134, 1289
<code>\raggedleft</code>	376–378
<code>\raggedright</code>	315 , 317–322, 338–344, 356, 375, 377, 378, 693–697, 705–708, 946–951, 959–962
<code>\raisebox</code>	141 , 541, 793, 1047, 1201, 1358
<code>\ref</code>	254 , 593, 845, 1101, 1256, 1421, 1422
<code>\refstepcounter</code>	109 –111, 253, 254, 593, 845, 1027, 1028, 1100, 1182, 1256, 1339, 1421
<code>\regex</code>	259 , 1102, 1258, 1423
<code>\relax</code>	7 , 11, 14, 21, 32, 33, 37, 39, 42–44, 47, 48, 63, 77, 82, 84, 88, 91, 103, 106, 111, 115, 116, 127, 128, 131–133, 135–140, 175, 190, 191, 193, 195, 196, 200–205, 208, 209, 212, 216, 217, 219, 223, 224, 245, 250, 251, 260, 261, 289, 298, 302, 316, 324, 325, 329, 330, 406, 407, 409–411, 413, 415, 428, 429, 443–445, 465, 467, 468, 470–472, 478, 479, 481, 483–485, 493, 501, 503, 504, 507, 508, 516, 518, 520, 523, 532–541, 556, 564–569, 572–575, 578–581, 583, 589, 591, 592, 596, 597, 625,
<code>\renewacronymstyle</code>	273 , 452–455, 457–460, 602, 649–652, 654–656, 854, 902–905, 907, 909, 1109, 1265, 1430
<code>\renewcommand</code>	5 –16, 19–25, 30, 32, 33, 37, 39, 40, 42, 44, 47, 51–55, 68, 81, 84, 85, 100, 111, 115, 116, 182, 184, 186, 187, 193, 194, 200–205, 224, 225, 231, 251, 266, 271, 272, 274–278, 280–283, 299, 300, 303–311, 313, 314, 316–322, 324–331, 333–335, 338–343, 407–

415, 425, 428, 429, 437, 440,
 442, 443, 450–461, 464–469,
 471–474, 477–484, 486–488,
 496, 502, 504, 514, 522, 523,
 560–562, 566, 567, 572–575,
 583, 584, 592, 601–612, 614,
 615, 617–619, 621–623, 625,
 628, 630, 636, 639, 641–643,
 648–660, 663, 664, 668–678,
 685–703, 705–707, 709–719,
 721–724, 726–730, 732, 733,
 735, 737, 738, 746, 753–755,
 765, 773, 774, 811–813, 818,
 819, 824–827, 835, 836, 843,
 844, 853–868, 870, 872, 874–
 876, 878, 881, 883, 884, 889,
 892, 894, 896, 901–913, 916,
 917, 921–931, 938–957, 959–
 973, 975–978, 981–983, 985,
 987, 989, 990, 992–994, 1002,
 1009–1011, 1021, 1028, 1030,
 1031, 1066–1068, 1073, 1074,
 1079–1082, 1091, 1092, 1099,
 1106, 1108–1117, 1122–1128,
 1130–1133, 1135–1138, 1140–
 1145, 1147–1149, 1157, 1163–
 1165, 1176, 1182, 1185, 1186,
 1222–1224, 1228, 1229, 1234–
 1238, 1246, 1247, 1255, 1261,
 1263–1273, 1277–1282, 1284–
 1287, 1291–1293, 1295, 1297–
 1300, 1302, 1304–1306, 1314,
 1320–1322, 1333, 1339, 1342,
 1343, 1379–1381, 1385, 1386,
 1391–1395, 1403, 1404, 1407,
 1420, 1427, 1429–1438, 1442–1450
\renewenvironment 255, 299, 303,
 306–312, 314, 315, 317–322,
 324–344, 398, 407, 409, 410,
 412, 594, 685, 687–691, 693–
 712, 846, 938, 940, 942–964,
 966, 1101, 1257, 1422, 1442–1449
\renewglossarystyle
 . 261, 597, 849, 1103, 1259, 1424
\RequireGlossariesLang
 50, 52, 416, 486, 487,
 662, 679, 736, 737, 915, 932,
 992, 993, 1146, 1147, 1303, 1304
\RequirePackage 4–6, 13, 14, 32, 45,
 51, 52, 288, 306, 312, 317, 323,
 330, 338, 415–417, 425, 464,
 468, 478, 484, 486, 625, 627,
 628, 662, 679, 688, 692, 694,
 697, 701, 705, 714, 717, 718,
 727, 728, 734, 736, 878, 880,
 881, 915, 932, 942, 945, 948,
 951, 954, 958, 967, 968, 972,
 982, 983, 990, 992, 1118, 1122,
 1123, 1126, 1127, 1137, 1145,
 1147, 1274, 1276, 1277, 1281,
 1292, 1300, 1303, 1305, 1439, 1442
\restorecounters@
 . 131, 534, 785, 1039, 1194, 1351
restoremakegloss (option) 37
\romannumeral . 212, 215, 216, 348–
 354, 361, 404, 405, 413, 414,
 578, 579, 675, 712, 713, 830,
 831, 928, 965, 966, 1085, 1086,
 1241, 1242, 1398, 1399, 1449, 1450

S
\s@glshyp@opt 127,
 127, 532, 783, 1037, 1191, 1348
\s@GlsSetXdyFirstLetterAfterDigits
 186,
 561, 813, 1067, 1068, 1223, 1380
\s@GlsSetXdyNumberGroupOrder . .
 186, 187, 561, 562,
 813, 1068, 1223, 1224, 1380, 1381
\s@ifglossaryexists 70, 70,
 221, 497, 582, 747, 748, 834,
 1003, 1090, 1158, 1245, 1315, 1402
\s@newglossary
 78, 78, 501, 751, 1007, 1162, 1319
sanitize (option) 31
sanitizesort (option) 30
\savecounters@
 . 131, 534, 785, 1039, 1193, 1350
savenumberlist (option) 12
savewrites (option) 40
sc-short-long (acrstyle) 276
sc-short-long-desc (acrstyle) 278, 454
section (option) 10
see (key) 83
seeautonumberlist (option) 13
\seename 50, 219, 486, 581, 736, 833,
 992, 1089, 1146, 1244, 1303, 1401
seenoindex (option) 9
\seq 207, 208, 226–228,
 232, 249, 250, 296, 298, 1404–1407

<code>\setabbreviationstyle</code>	<code>\setglossarysection</code>
..... 288, 876, 1118, 1274, 1439	... 58, 490, 740, 995, 1150, 1307
<code>\SetAcronymLists</code>	<code>\setglossarystyle</code>
... 23, 473, 722, 977, 1131, 1286	. 223, 250, 261, 289, 303–305,
<code>\SetAcronymStyle</code>	307–315, 318–322, 324–330,
288,	332–337, 339–341, 343, 408–
479, 480, 623, 624, 729, 876, 877	411, 414, 415, 583, 596, 597,
<code>\setacronymstyle</code> 266, 272, 288, 602,	625, 687–693, 695–711, 713,
854, 876, 1106, 1109, 1118,	835, 848, 849, 878, 940–965,
1261, 1264, 1274, 1427, 1430, 1439	967, 1090, 1099, 1103, 1119,
<code>\setbool</code> 31, 477, 727, 981, 1136, 1291	1246, 1254, 1259, 1274, 1403,
<code>\setbox</code>	1420, 1424, 1440, 1442–1448, 1450
316, 693, 694, 947	<code>\setglossentrycompatibility</code> ...
<code>\setcounter</code> ... 110, 253, 593, 845,	258, 591, 595, 597, 843, 847, 849
1027, 1100, 1182, 1256, 1339, 1421	<code>\setkeys</code>
<code>\SetCustomDisplayStyle</code>	31, 36, 48, 58, 102,
..... 288, 624, 625, 877, 878	130, 184, 223, 271, 477, 480,
<code>\SetCustomStyle</code>	485, 490, 515, 533, 560, 561,
288, 625, 878	583, 601, 612, 614, 615, 617,
<code>\SetDefaultAcronymDisplayStyle</code>	619, 621, 623, 625, 727, 730,
..... 285, 611,	735, 740, 766, 784, 812, 835,
612, 863, 864, 1118, 1274, 1439	853, 864, 866, 868, 870, 872,
<code>\SetDefaultAcronymStyle</code>	874, 876, 878, 982, 985, 991,
..... 285, 612, 623, 864, 876	995, 1022, 1038, 1067, 1090,
<code>\SetDescriptionAcronymDisplayStyle</code>	1108, 1136, 1139, 1145, 1150,
..... 286, 616, 617, 868, 870	1177, 1193, 1222, 1223, 1246,
<code>\SetDescriptionAcronymStyle</code> ...	1263, 1291, 1295, 1302, 1307,
..... 286, 617, 623, 870, 877	1334, 1350, 1379, 1380, 1403, 1429
<code>\SetDescriptionDUAAcronymDisplayStyle</code>	<code>\setlength</code>
..... 286, 614–616, 867, 868	. 305, 306, 317, 325–328, 331,
<code>\SetDescriptionDUAAcronymStyle</code>	338, 407, 409, 410, 414, 675,
..... 286, 615, 623, 868, 877	688, 694, 697–699, 701, 705,
<code>\SetDescriptionFootnoteAcronymDisplayStyle</code>	709–713, 928, 941, 942, 948,
..... 286, 612, 614, 865–867	951–955, 958, 959, 962–964,
<code>\SetDescriptionFootnoteAcronymStyle</code>	966, 1442–1444, 1446–1448, 1450
..... 286, 614, 623, 866, 877	<code>\SetSmallAcronymDisplayStyle</code> ..
<code>\SetDUADisplayStyle</code> 287, 620–622, 873–875
..... 287, 622, 623, 875, 876	<code>\SetSmallAcronymStyle</code>
<code>\SetDUASStyle</code> .. 287, 623, 624, 876, 877 287, 621, 624, 874, 877
<code>\setentrycounter</code> .. 62, 189, 247,	<code>\setstylefile</code>
260, 492, 563, 591, 596, 663,	53, 488, 738, 993, 994, 1148, 1305
742, 815, 843, 848, 916, 998,	<code>\settodepth</code> 111, 1028, 1183, 1340
1070, 1098, 1103, 1153, 1225,	<code>\settoheight</code>
1254, 1259, 1310, 1382, 1418, 1424	. 111, 541, 793, 1028, 1183, 1340
<code>\SetFootnoteAcronymDisplayStyle</code>	<code>\settowidth</code>
..... 287, 618, 619, 870, 872	111,
<code>\SetFootnoteAcronymStyle</code>	275, 369–373, 604, 674, 675,
..... 287, 619, 624, 872, 877	712, 713, 856, 928, 965, 966,
<code>\SetGenericNewAcronym</code>	1028, 1111, 1183, 1266, 1340, 1432
271, 272, 601, 602, 853, 854,	<code>\setupglossaries</code>
1108, 1109, 1263, 1265, 1429, 1430	... 47, 484, 735, 990, 1145, 1302
<code>\setglossarypreamble</code>	<code>\sfcode</code> . 14, 468, 718, 972, 1127, 1281
... 56, 489, 739, 994, 1149, 1306	

<code>\SmallNewAcronymDef</code>	287 ,	<code>\string</code>	9 , 10 , 20 , 25 , 29 ,
	461 , 620 , 622 , 660 , 873 , 875 , 913		36 , 37 , 39 , 41 , 42 , 49 , 60 – 62 ,
<code>sort</code> (key)	82		64 – 66 , 68 , 77 , 79 , 84 , 86 , 92 ,
<code>sort</code> (option)	16		95 , 96 , 107 , 108 , 115 , 117 – 119 ,
<code>\space</code> 9 , 10 , 36 , 37 , 39 , 41 , 42 , 49 ,			132 , 133 , 135 – 137 , 139 , 141 ,
61 , 65 , 66 , 68 , 77 , 84 , 86 , 115 ,			182 , 185 – 193 , 195 – 197 , 201 –
117 – 119 , 124 , 125 , 182 , 188 –			204 , 206 , 207 , 213 , 217 – 219 ,
191 , 196 , 198 – 204 , 206 , 207 ,			223 , 225 , 237 , 238 , 252 , 266 ,
213 , 220 , 223 , 225 , 252 , 254 ,			288 , 297 , 463 , 464 , 466 , 471 ,
267 , 274 , 275 , 277 , 278 , 280 –			474 , 480 – 483 , 486 , 491 – 497 ,
283 , 287 , 298 , 301 , 303 , 304 ,			501 , 504 , 505 , 508 , 509 , 511 ,
306 , 318 , 331 , 338 , 339 , 407 –			519 , 522 , 524 , 525 , 531 , 532 ,
411 , 413 , 414 , 442 , 443 , 452 –			535 , 537 , 538 , 540 , 559 , 561 –
461 , 466 , 480 – 483 , 486 , 492 ,			566 , 568 , 569 , 572 – 576 , 580 ,
495 , 497 , 504 , 505 , 522 , 524 ,			581 , 583 , 584 , 587 , 588 , 592 ,
525 , 531 , 532 , 559 , 562 – 565 ,			595 – 597 , 663 – 668 , 684 , 716 ,
569 – 576 , 581 , 583 , 584 , 593 ,			721 , 724 , 730 – 733 , 736 , 741 ,
594 , 597 , 600 , 603 – 611 , 616 ,			742 , 744 – 747 , 751 , 752 , 754 –
620 , 642 , 643 , 649 – 657 , 663 –			756 , 759 – 762 , 770 , 773 – 775 ,
667 , 669 – 677 , 685 – 689 , 694 ,			782 , 783 , 786 , 788 – 790 , 792 ,
701 , 705 , 709 – 713 , 716 , 730 –			811 – 818 , 820 , 821 , 824 – 828 ,
733 , 736 , 742 , 745 , 747 , 754 –			832 , 833 , 835 , 836 , 839 , 840 ,
756 , 773 , 775 , 781 – 783 , 811 ,			844 , 847 , 849 , 876 , 916 – 921 ,
814 – 817 , 821 – 828 , 833 , 835 ,			937 , 970 , 975 , 978 , 980 , 985 ,
836 , 845 , 849 , 851 , 855 – 863 ,			986 , 988 , 991 , 997 – 1003 , 1007 ,
869 , 873 , 896 , 902 – 910 , 916 –			1008 , 1010 – 1012 , 1015 – 1018 ,
920 , 923 , 925 – 931 , 938 – 942 ,			1026 , 1030 – 1033 , 1039 , 1040 ,
948 , 955 , 959 , 962 – 966 , 970 ,			1042 , 1043 , 1045 , 1046 , 1066 –
985 , 986 , 988 , 991 , 998 , 1000 –			1073 , 1075 , 1076 , 1079 – 1081 ,
1003 , 1007 , 1010 – 1012 , 1030 ,			1083 , 1085 , 1087 , 1088 , 1090 –
1032 , 1033 , 1036 , 1066 , 1069 –			1092 , 1094 , 1095 , 1100 , 1106 ,
1071 , 1075 – 1081 , 1083 , 1085 ,			1118 , 1121 , 1125 , 1130 , 1133 ,
1089 , 1090 , 1092 , 1101 , 1106 ,			1135 , 1139 – 1141 , 1143 , 1146 ,
1110 – 1118 , 1122 , 1125 , 1139 –			1151 – 1157 , 1161 , 1162 , 1165 ,
1141 , 1143 , 1146 , 1152 , 1155 –			1166 , 1170 , 1172 , 1173 , 1180 ,
1157 , 1161 , 1165 , 1166 , 1185 ,			1181 , 1185 – 1187 , 1194 – 1198 ,
1187 , 1190 , 1191 , 1221 , 1224 –			1200 , 1202 , 1221 , 1223 – 1228 ,
1227 , 1231 – 1238 , 1241 , 1245 –			1230 – 1232 , 1235 – 1239 , 1241 ,
1247 , 1256 , 1257 , 1262 , 1265 –			1243 , 1244 , 1246 , 1247 , 1250 ,
1274 , 1279 , 1295 , 1296 , 1298 ,			1251 , 1255 , 1261 , 1274 , 1279 ,
1303 , 1309 , 1312 – 1314 , 1318 ,			1284 , 1287 , 1290 , 1295 , 1296 ,
1322 , 1323 , 1342 , 1344 , 1347 ,			1298 , 1303 , 1308 – 1314 , 1318 ,
1348 , 1378 , 1381 – 1384 , 1388 –			1319 , 1322 , 1323 , 1327 , 1329 ,
1393 , 1395 , 1398 , 1402 – 1404 ,			1330 , 1337 , 1338 , 1342 – 1344 ,
1422 , 1427 , 1431 – 1439 , 1446 – 1450			1351 – 1355 , 1357 , 1359 , 1378 ,
<code>\spacefactor</code>			1380 – 1385 , 1387 – 1389 , 1391 –
. . . 14 , 468 , 718 , 972 , 1127 , 1281			1396 , 1398 , 1400 , 1401 , 1403 ,
<code>\SS</code>	29 , 476 , 726 , 980 , 1135 , 1290		1404 , 1411 , 1412 , 1421 , 1427 , 1439
<code>\ss</code>	29 , 476 , 726 , 980 , 1135 , 1290	<code>\strut</code>	241 , 303 – 306 , 308 , 310 ,
<code>\str</code>	248 , 260 , 1103 , 1258 , 1419 , 1424		318 , 319 , 321 , 331 , 333 , 335 ,

339, 340, 342, 388, 402, 411, 596, 669–672, 674, 676–678, 687–690, 694–696, 701–703, 705–707, 711, 848, 923–925, 927, 929–931, 940–944, 948– 950, 955–957, 959–961, 965, 1448	
style (option)	11
style-options (option)	11
styles:	
altlist	304
altlistgroup	304
altlisthypergroup	304
altlong4col	311
altlong4col-booktabs	314
altlong4colborder	311
altlong4colheader	311
altlong4colheaderborder	312
altlongragged4col	320
altlongragged4col-booktabs	315
altlongragged4colborder	322
altlongragged4colheader	321
altlongragged4colheaderborder	322
altsuper4col	336
altsuper4colborder	337
altsuper4colheader	336
altsuper4colheaderborder	337
altsuperragged4col	342
altsuperragged4colborder	343
altsuperragged4colheader	343
altsuperragged4colheaderborder	343
almtree	412
almtreegroup	414
almtreehypergroup	415
index	407
indexgroup	408
indexhypergroup	408
inline	299
list	302
listdotted	305
listgroup	303
listhypergroup	303
long	306
long-booktabs	312
long3col	307
long3col-booktabs	313
long3colborder	308
long3colheader	309
long3colheaderborder	309
long4col	309
long4col-booktabs	313
long4colborder	310
long4colheader	310
long4colheaderborder	310
longborder	307
longheader	307
longheaderborder	307
longragged	317
longragged-booktabs	314
longragged3col	319
longragged3col-booktabs	315
longragged3colborder	320
longragged3colheader	320
longragged3colheaderborder	320
longraggedborder	318
longraggedheader	318
longraggedheaderborder	319
mcolalmtree	329
mcolalmtreegroup	329
mcolalmtreehypergroup	329
mcolalmtreespannav	330
mcolindex	325
mcolindexgroup	325
mcolindexhypergroup	325
mcolindexspannav	326
mcoltree	326
mcoltree*	323
mcoltreegroup	326
mcoltreehypergroup	327
mcoltreenoname	327
mcoltreenonamegroup	328
mcoltreenonamehypergroup	328
mcoltreenonamespannav	328
mcoltreespannav	327
sublistdotted	305
super	331
super3col	332
super3colborder	333
super3colheader	334
super3colheaderborder	334
super4col	334
super4colborder	335
super4colheader	335
super4colheaderborder	336
superborder	332
superheader	332
superheaderborder	332
superragged	338
superragged3col	340

	870, 872, 875, 909, 1111, 1112,		
	1117, 1267, 1268, 1272, 1433, 1438		
<code>\texttt</code>	264, 599, 851, 1105, 1261, 1426	<code>\theHglentrycounter</code>	131,
<code>\textulc</code>	266, 599, 851, 1106, 1262, 1427		132, 214, 216, 534, 578, 579,
<code>\textup</code>	264, 266, 599, 851,		664, 785, 830, 831, 917, 1038,
	1105, 1106, 1261, 1262, 1426, 1427		1039, 1086, 1087, 1193, 1194,
<code>\TH</code>	29, 476, 726, 980, 1135, 1290	<code>\theH</code>	218, 580, 832, 1088, 1244, 1400
<code>\th</code>	29, 476, 726, 980, 1135, 1290	<code>\theHglossaryentry</code>	15, 469, 718, 973, 1127, 1282
<code>\the</code>	49, 51, 61, 66,	<code>\theHglossarysubentry</code>	16, 469, 719, 973, 1128, 1282
	68, 79, 135–140, 187, 191, 193–	<code>\theHglentrycounter</code>	131, 132, 214,
	195, 209, 211, 215, 216, 221,		216, 534, 535, 578, 579, 785,
	245, 263, 264, 271, 272, 274,		830, 831, 1039, 1086, 1087,
	275, 280, 281, 452, 456, 458,		1194, 1241, 1242, 1351, 1398, 1399
	461, 485, 487, 492, 495, 497,	<code>\thepage</code>	141, 1202, 1359
	501, 536–541, 562, 565, 567,	<code>\thesection</code>	48, 485, 735, 991, 1146, 1303
	568, 576, 577, 579, 582, 589,	<code>\this@dialect</code>	52, 416, 487, 662,
	594, 595, 598, 601–604, 608,		679, 737, 915, 932, 993, 1147, 1304
	609, 611–622, 624, 628, 649,	<code>\tiny</code>	8, 466, 715, 969, 1124, 1279
	650, 653, 654, 657–661, 663,	<code>\tl</code>	5, 17–19, 30,
	664, 667, 736, 737, 742, 746,		46, 70, 74, 110, 141, 207, 226–
	747, 751, 787–793, 814, 817–		229, 231, 232, 240–243, 249,
	820, 828, 829, 831, 834, 841,		260, 323, 324, 348–356, 358–
	846, 847, 850, 853–856, 859,		376, 384–386, 388–398, 400,
	861, 863–867, 869, 871, 873–		401, 968, 1122, 1182, 1202,
	875, 877, 878, 881, 902, 903,		1277, 1283, 1284, 1301, 1339,
	906, 907, 910–914, 916, 918,		1359, 1404–1408, 1413–1415, 1424
	920, 991, 992, 998, 1001, 1003,	<code>toc (option)</code>	10
	1007, 1041–1046, 1068, 1071,	<code>\token</code>	208
	1073, 1075, 1083, 1084, 1086,	<code>\toks@</code>	49, 51, 61, 66,
	1089, 1097, 1105, 1108–1111,		68, 79, 134–140, 193–195, 220,
	1114, 1115, 1146, 1147, 1152,		221, 263, 264, 485–487, 492,
	1156, 1157, 1162, 1196–1201,		495, 497, 501, 536–541, 566–
	1224, 1227, 1229–1231, 1239,		568, 582, 594, 595, 598, 628,
	1240, 1242, 1245, 1252, 1260,		663, 736, 737, 742, 746, 747,
	1264–1266, 1270, 1271, 1303,		751, 787–793, 818–820, 834,
	1304, 1309, 1313, 1314, 1319,		846, 847, 850, 881, 916, 991,
	1353–1358, 1381, 1384, 1386–		992, 998, 1001, 1003, 1007,
	1388, 1396, 1397, 1399, 1402,		1041–1046, 1073, 1075, 1089,
	1417, 1426, 1429–1432, 1435, 1436		1104, 1105, 1146, 1147, 1152,
<code>\the@numberlist</code>	182, 183,		1156, 1157, 1162, 1196–1201,
	560, 811, 812, 1066, 1222, 1379		1229–1231, 1245, 1260, 1303,
<code>\theglossary</code>	6, 7, 464, 714, 968, 1123, 1277	<code>\toprule</code>	313, 314, 692, 945, 946
<code>theglossary (env.)</code>	255	<code>\TrackIfKnownLanguage</code>	51, 992, 1147, 1304
<code>\theglossaryentry</code>	15, 254, 469, 593, 718, 845,	<code>\trans@languages</code>	51, 486, 736, 992, 1147, 1304
	973, 1101, 1127, 1256, 1282, 1422		
<code>\theglossarysubentry</code>	16, 254, 469, 594, 719, 845,		
	973, 1101, 1128, 1257, 1282, 1422		

`\translate` . 51, 52, 487, 737, 738,
 992, 993, 1147, 1148, 1304, 1305
`translate` (option) 32
`\translatelet`
 ... 20–22, 43, 471, 472, 483,
 484, 721, 722, 734, 975, 976,
 989, 1130, 1131, 1144, 1285, 1299
`tree` (style) 408
`tree*` (style) 344
`treegroup` (style) 409
`treehypergroup` (style) 410
`treenoname` (style) 410
`treenonamegroup` (style) 411
`treenonamehypergroup` (style) 411
`\triangleright`
 8, 466, 715, 969, 1124, 1279
`\ttfamily` 8, 466, 715, 969, 1124, 1279
`\TX@trial`
 . 111, 520, 771, 1028, 1183, 1340
`type` (key) 83
`\typeout` 24, 473, 723, 977, 1132, 1287

U

`\u` 28, 475, 725, 979, 1134, 1289
`\uccode` 245, 589, 841, 1096, 1252, 1416
`ucmark` (option) 14
`\undef` 85, 91, 221, 504, 508, 582, 755,
 759, 834, 1011, 1015, 1089,
 1165, 1169, 1245, 1322, 1326, 1402
`\unexpanded` 44, 990, 1144, 1300
`\unskip` .. 100, 305, 514, 670, 688,
 765, 923, 941, 1021, 1176, 1333
`\unvbox` 316, 693, 694, 947
`\use` 227, 228, 233, 240–242,
 248, 370–372, 378–381, 383–
 389, 395, 1409, 1413, 1414, 1419
`\usedictionary`
 ... 51, 486, 737, 992, 1147, 1304
`\usepackage` 29, 237, 238, 266, 463,
 464, 587, 588, 839, 840, 980,
 1094, 1095, 1106, 1135, 1250,
 1251, 1261, 1290, 1411, 1412, 1427
`user1` (key) 85
`user1access` (key) 427
`user2` (key) 85
`user2access` (key) 427
`user3` (key) 85
`user3access` (key) 427
`user4` (key) 86
`user4access` (key) 427

`user5` (key) 86
`user5access` (key) 428
`user6` (key) 86
`user6access` (key) 428
`\UTFviii@two@octets`
 ... 29, 476, 726, 980, 1135, 1290
`\UTFviii@two@octets@combine` ...
 ... 29, 476, 726, 980, 1135, 1290

V

`\v` 28, 476, 725, 980, 1134, 1289
`\value` 110, 1027, 1182, 1339
`\vbox` 316, 693, 694, 947
`\vsize` 316, 694, 947
`\vskip` 302,
 316, 325, 406, 687, 693, 697,
 708, 940, 947, 951, 962, 1442, 1445
`\vss` 316, 693, 694, 947

W

`\warn@nomakeglossaries` ... 200,
 202–204, 572–574, 823–826,
 1078–1081, 1234–1237, 1391–1393
`\warn@noprintglossary` 202–
 204, 221, 224, 572–574, 582,
 583, 824–826, 834, 835, 1079–
 1081, 1089, 1091, 1235, 1236,
 1245, 1247, 1392, 1393, 1402, 1404
`\write` 38, 47, 92, 117, 187–
 192, 204, 209, 225, 481, 508,
 509, 524, 562–566, 570, 571,
 574, 577, 584, 664–668, 731,
 759, 774, 775, 814–818, 822,
 823, 826, 828, 836, 918–921,
 986, 1015, 1031, 1032, 1068–
 1072, 1076–1079, 1081, 1084,
 1091, 1092, 1141, 1170, 1186,
 1224–1228, 1232–1234, 1236,
 1239, 1247, 1296, 1301, 1327,
 1343, 1381–1385, 1393, 1396, 1404
`writelnslabelnames` (option) 38
`writelnslabels` (option) 38
`\writeist`
 187, 196, 197, 562, 565, 566,
 569, 664, 667, 668, 814, 816,
 818, 821, 918, 920, 921, 1068,
 1071, 1072, 1076, 1224, 1227,
 1228, 1231, 1381, 1384, 1385, 1388

X

`\x` 598

<code>\xatlevel@</code>	<code>\xmakefirstuc</code>
. 131, 534, 785, 1038, 1193, 1350 525–527, 531, 555, 556,
<code>\xcapitalisewords</code>	680, 776–778, 782, 807, 808, 933
557	<code>\xspace</code>
<code>\xdef</code> 104, 516, 518, 583, 684,	265,
767, 937, 938, 1023, 1178, 1335	599, 851, 1105, 1261, 1426, 1427
<code>\xglsaccsupp</code> .. 434, <u>434</u> , 633, 634, 887	
<code>\xglsfieldaccsupp</code> <u>433</u> , 633, 886	Y
<code>\xifinlist</code> . 90, 758, 1014, 1169, 1326	<code>\year</code>
<code>\xifinlistcs</code> .. 234, 235, 239, 585,	187, 191, 562, 565,
588, 836, 837, 840, 1092, 1093,	664, 667, 814, 817, 918, 920,
1095, 1248, 1251, 1409, 1410, 1412	1068, 1071, 1224, 1227, 1381, 1384
<code>xindy</code> (option)	Z
<u>36</u>	<code>\Z</code>
<code>xindygloss</code> (option)	259, 1102, 1258, 1423
<u>36</u>	<code>\z@</code>
<code>xindynoglsnumbers</code> (option)	316, 693, 694, 947
<u>36</u>	