

Documented Code For glossaries v4.47

Nicola L.C. Talbot

Dickimaw Books

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

2021-09-20

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 v4.47: L^AT_EX2_ε Package to Assist Generating Glossaries”.

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

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

INSTALL Installation instructions.

CHANGES Change log.

README.md Package summary.

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

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

Contents

1	Main Package Code	4
1.1	Package Definition	4
1.2	Package Options	5
1.3	Predefined Text	40
1.4	Xindy	50
1.5	Loops and conditionals	59
1.6	Defining new glossaries	65
1.7	Defining new entries	70
1.8	Resetting and unsetting entry flags	97
1.9	Keeping Track of How Many Times an Entry Has Been Unset	100
1.10	Loading files containing glossary entries	105
1.11	Using glossary entries in the text	105
1.12	Adding an entry to the glossary without generating text	166
1.13	Creating associated files	167
1.14	Writing information to associated files	188
1.15	Glossary Entry Cross-References	197
1.16	Displaying the glossary	199
1.17	Acronyms	228
1.18	Predefined acronym styles	232
1.19	Predefined Glossary Styles	266
1.20	Debugging Commands	266
1.21	Compatibility with version 2.07 and below	271
2	Prefix Support (glossaries-prefix Code)	273
3	Glossary Styles	280
3.1	Glossary hyper-navigation definitions (glossary-hypernav package)	280
3.2	In-line Style (glossary-inline.sty)	282
3.3	List Style (glossary-list.sty)	285
3.4	Glossary Styles using longtable (the glossary-long package)	288
3.5	Glossary Styles using longtable and booktabs (the glossary-longbooktabs) package	294
3.6	Glossary Styles using longtable (the glossary-longragged package)	299
3.7	Glossary Styles using multicol (glossary-mcols.sty)	304
3.8	Glossary Styles using supertabular environment (glossary-super package)	310
3.9	Glossary Styles using supertabular environment (glossary-superragged package)	317
3.10	Tree Styles (glossary-tree.sty)	323

4 Backwards Compatibility	333
4.1 glossaries-compatible-207	333
4.2 glossaries-compatible-307	339
5 Accessibility Support (glossaries-accsupp Code)	353
5.1 Defining Replacement Text	355
5.2 Accessing Replacement Text	359
5.3 Displaying the Glossary	384
5.4 Acronyms	385
5.5 Debugging Commands	399
6 Multi-Lingual Support	402
6.1 Polyglossia Captions	402
6.2 Rollback v4.46 (glossaries-2020-03-19.sty)	403
Glossary	571
Change History	572
Index	599

1 Main Package Code

1.1 Package Definition

This package requires $\text{\LaTeX}2_{\epsilon}$.

```
1 \NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

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

```
3 \DeclareCurrentRelease{v4.47}{2021-09-20}
```

Declare package:

```
4 \ProvidesPackage{glossaries}[2021/09/20 v4.47 (NLCT)]
```

Required packages:

```
5 \RequirePackage{ifthen}
```

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

```
7 \RequirePackage{mfirstuc}
```

The textcase package has much better case changing handling, so use `\MakeTextUppercase` instead of `\MakeUppercase`

```
8 \RequirePackage{textcase}
```

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

```
10 \RequirePackage{xfor}
```

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

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

```
12 \RequirePackage{amsgen}
```

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

```
13 \RequirePackage{etoolbox}
```

Check if doc has been loaded.

```
f@gls@docloaded
```

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

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

```
16 {%
```

```
17   \@gls@docloadedtrue
```

```
18 }%
```

```
19 {%
```

```

20 \ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
21 }
22 \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.

```
org@theglossary
```

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

```
@endtheglossary
```

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

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

```

25 \let\glsorg@PrintChanges\PrintChanges
26 \renewcommand{\PrintChanges}{%
27   \begingroup
28     \let\theglossary\glsorg@theglossary
29     \let\endtheglossary\glsorg@endtheglossary
30     \glsorg@PrintChanges
31   \endgroup
32 }

```

End of doc stuff.

```
33 \fi
```

1.2 Package Options

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

```

34 \newif\if@gls@debug
35 \define@choicekey{glossaries.sty}{debug}{\@gls@debug@val\@gls@debug@nr}%
36 {true,false,showtargets,showaccsupp}{true}{%
37   \ifcase\@gls@debug@nr\relax
38     % debug=true
39     \@gls@debugtrue
40     \renewcommand*\GlossariesWarning}[1]{%
41       \PackageWarning{glossaries}{##1}%
42     }%
43     \renewcommand*\GlossariesWarningNoLine}[1]{%
44       \PackageWarningNoLine{glossaries}{##1}%
45     }%
46     \let\@glsshowtarget\@gobble
47     \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
48   \or
49     % debug=false
50     \@gls@debugfalse

```

```

51 \let\@glsshowtarget\@gobble
52 \let\@glsshowaccsupp\@gobblethree
53 \PackageInfo{glossaries}{debug mode OFF}%
54 \or
55 % debug=showtargets
56 \@gls@debugtrue
57 \renewcommand*{\GlossariesWarning}[1]{%
58 \PackageWarning{glossaries}{##1}%
59 }%
60 \renewcommand*{\GlossariesWarningNoLine}[1]{%
61 \PackageWarningNoLine{glossaries}{##1}%
62 }%
63 \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
64 \renewcommand{\@glsshowtarget}{\@glsshowtarget}%
65 \or
66 % debug=showaccsupp
67 \@gls@debugtrue
68 \renewcommand*{\GlossariesWarning}[1]{%
69 \PackageWarning{glossaries}{##1}%
70 }%
71 \renewcommand*{\GlossariesWarningNoLine}[1]{%
72 \PackageWarningNoLine{glossaries}{##1}%
73 }%
74 \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
75 \renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
76 \fi
77 }

```

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

```

78 \newcommand*{\glsshowtarget}[1]{%
79 \ifmode
80 \nfss@text{\glsshowtargetfont [#1]}%
81 \else
82 \ifinner

```

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

```

83 {\glsshowtargetfont [#1]}%
84 \else
85 \glsshowtargetouter{#1}%
86 \fi
87 \fi
88 }

```

`showtargetouter`

```

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

```

`howtargetsymbol`

```

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

```

sshowtargetfont

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

\@glsshowtarget

debug=showtargets will redefine this.

```
93 \newcommand*{\@glsshowtarget}[1]{}
```

@@glsshowtarget

Need to detokenize the label in the event that it contains awkward characters like under-scores.

```
94 \newrobustcmd*{\@@glsshowtarget}[1]{%
```

```
95   \begingroup
```

```
96   \protected@edef\@gls@tmp{#1}%
```

```
97   \@onelevel@sanitize\@gls@tmp
```

```
98   \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
```

```
99   \endgroup
```

```
100 }
```

@glsshowaccsupp

debug=showaccsupp will redefine this.

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

\glsshowaccsupp

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

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

```
103   \ifstrempy{#1}%
```

```
104   {\@glsshowtarget{/#2 (#3)}}%
```

```
105   {\@glsshowtarget{/#2 (#3) [#1]}}%
```

```
106 }
```

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

gls@see@noindex

```
107 \newcommand*{\@gls@see@noindex}{%
```

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

```
109   {'\gls@xr@key' key may only be used after \string\makeglossaries\space
```

```
110   or \string\makenoidxglossaries\space (or move
```

```
111   \string\newglossaryentry\space
```

```
112   definitions into the preamble))%
```

```
113   {You must use \string\makeglossaries\space
```

```
114   or \string\makenoidxglossaries\space before defining
```

```
115   any entries that have a '\gls@xr@key' key. It may
```

```
116   be that the 'see' key has been written to the .glsdefs
```

```
117   file from the previous run, in which case you need to
```

```
118   move your definitions
```

```
119   to the preamble if you don't want to use
```

```
120   \string\makeglossaries\space
```

```
121   or \string\makenoidxglossaries}%
```

```
122 }
```

seenoinindex

```
123 \define@choicekey{glossaries.sty}{seenoinindex}%
124 [\gls@seenoinindex@val\gls@seenoinindex@nr]{error,warn,ignore}{%
125   \ifcase\gls@seenoinindex@nr
126     \renewcommand*{\@gls@see@noindex}{%
127       \PackageError{glossaries}%
128       {'\gls@xr@key' key may only be used after \string\makeglossaries\space
129       or \string\makenoidxglossaries}%
130       {You must use \string\makeglossaries\space
131       or \string\makenoidxglossaries\space before defining
132       any entries that have a '\gls@xr@key' key}%
133     }%
134   \or
135     \renewcommand*{\@gls@see@noindex}{%
136       \GlossariesWarning{'\gls@xr@key' key ignored}%
137     }%
138   \or
139     \renewcommand*{\@gls@see@noindex}{}%
140   \fi
141 }
```

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.

```
142 \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.

```
143 \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.

```
144 \ifcsundef{chapter}%
145   {\newcommand*{\@@glossarysec}{section}}%
146   {\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.

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

Determine whether or not to use numbered sections.

glossarysecstar

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


glossaryseclabel

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

\glsautoprefix Prefix to add before label if automatically generated:

```
152 \newcommand*{\glsautoprefix}{}
```

numberedsection

```
153 \define@choicekey{glossaries.sty}{numberedsection}%
154 [\gls@numberedsection@val\gls@numberedsection@nr]{%
155 false,nolabel,autolabel,nameref}[nolabel]{%
156 \ifcase\gls@numberedsection@nr\relax
157 \renewcommand*{\@@glossarysecstar}{*}%
158 \renewcommand*{\@@glossaryseclabel}{}%
159 \or
160 \renewcommand*{\@@glossarysecstar}{}%
161 \renewcommand*{\@@glossaryseclabel}{}%
162 \or
163 \renewcommand*{\@@glossarysecstar}{}%
164 \renewcommand*{\@@glossaryseclabel}{%
165 \label{\glsautoprefix\@glo@type}}%
166 \or
167 \renewcommand*{\@@glossarysecstar}{*}%
168 \renewcommand*{\@@glossaryseclabel}{%
169 \protected@edef\@currentlabelname{\glossarytoctitle}%
170 \label{\glsautoprefix\@glo@type}}%
171 \fi
172 }
```

The default glossary style is stored in `\@glossary@default@style`. This is initialised to list. (The list style is defined in the accompanying package described in [section 1.19](#).) Note that the list style is incompatible with classicthesis so change the default to index if that package has been loaded.

y@default@style

```
173 \ifpackageloaded{classicthesis}
174 {\newcommand*{\@glossary@default@style}{index}}
175 {\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 [section 1.19](#). This now uses `\def` instead of `\renewcommand` as `\@glossary@default@style` may have been set to `\relax`.

```
176 \define@key{glossaries.sty}{style}{%
177 \def\@glossary@default@style{#1}%
178 }
```

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.

s@declareoption

```
179 \newcommand*{\@gls@declareoption}[2]{%
180   \DeclareOptionX{#1}{#2}%
181   \DeclareOption{#1}{#2}%
182 }
```

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”:

aryentrynumbers

```
183 \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).

```
184 \@gls@declareoption{nonumberlist}{%
185   \renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
186 }
```

savenumberlist Provide means to store the number list for entries.

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

eautionumberlist

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

eautionumberlist Automatically activates number list for entries containing the see key.

```
190 \@gls@declareoption{seeautonumberlist}{%
191   \renewcommand*{\@glo@seeautonumberlist}{%
192     \def\@glo@prefix{\glsnextpages}%
193   }%
194 }
```

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) so `esclocations=false` will switch off this mechanism allowing for a faster and more stable approach.

```
195 \define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
196 \glsclocationstrue
```

\@gls@loadlong

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

`nolong` This option prevents 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.

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

`\@gls@loadsuper` The package isn't loaded if isn't installed.

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

`nosuper` This option prevents 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.

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

`\@gls@loadlist`

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

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

```
204 \@gls@declareoption{nolist}{%
205   \renewcommand*{\@gls@loadlist}{%
206     \ifdefstring{\@glossary@default@style}{list}%
207     {\let\@glossary@default@style\relax}%
208     }%
209   }%
210 }
```

`\@gls@loadtree`

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

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

```
212 \@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).

```
213 \@gls@declareoption{nostyles}{%
214   \renewcommand*{\@gls@loadlong}{}%
215   \renewcommand*{\@gls@loadsuper}{}%
216   \renewcommand*{\@gls@loadlist}{}%
217   \renewcommand*{\@gls@loadtree}{}%
218   \let\@glossary@default@style\relax
219 }
```

postdescription 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.)

```

220 \newcommand*{\glspostdescription}{%
221   \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
222 }

```

nopostdot Boolean option to suppress post description dot

```

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

```

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

```

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

```

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

```

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

228 \@ifclassloaded{memoir}
229 {%
230   \glsucmarktrue
231 }%
232 {%
233   \glsucmarkfalse
234 }

```

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.

aryentrycounter

```

235 \newcommand*{\@gls@define@glossaryentrycounter}{%
236   \ifgl sentrycounter

```

Define the glossaryentry counter if it doesn't already exist.

```

237   \ifundef\c@glossaryentry
238   {%
239     \ifx\@gls@counterwithin\@empty
240       \newcounter{glossaryentry}%
241     \else
242       \newcounter{glossaryentry}[\@gls@counterwithin]%
243     \fi
244     \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
245   }%
246   {}%
247 \fi
248 }

```

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.

```

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

```

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

```

251 \define@key{glossaries.sty}{counterwithin}{%
252   \renewcommand*{\@gls@counterwithin}{#1}%
253   \glentrycountertrue
254   \@gls@define@glossaryentrycounter
255 }

```

s@counterwithin The default value is no parent counter:

```

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

```

lossarysubentry 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.

subentrycounter

```

257 \newcommand{\@gls@define@glossarysubentrycounter}{%
  Check if counter already defined.
258   \ifundef\c@glossarysubentry
259   {%
260     \ifglssubentrycounter
261     \ifglentrycounter
262     \newcounter{glossarysubentry}[glossaryentry]%
263     \else
264     \newcounter{glossarysubentry}%
265     \fi
  }%
  As with \theHglossaryentry, this starts with \currentglossary. to help avoid duplicate
  hyper targets.
266   \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
267   \fi
268 }%
269 {}%
270 }

```

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.

```

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

```

efault@sorttype Initialise default sort for \printnoidxglossary

```

273 \newcommand*{\@gls@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`.

```
274 \define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
275   \renewcommand*{\@glo@default@sorttype}{#1}%
276   \csname @gls@setupsort@#1\endcsname
277 }
```

`prestandardsort`

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

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.

```
278 \newcommand*{\glsprestandardsort}[3]{%
279   \glsdosanitizesort
280 }
```

`check@sortallowed`

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

`upsort@standard`

Set up the macros for default sorting.

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

Store entry information when it's defined.

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

No count register required for standard sort.

```
284   \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.)

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

```
289     \let\glsdosanitizesort\@gls@sanitizesort
290     \glsprestandardsort{\@glo@sort}{##1}{##2}%
291     \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
292   }%
```

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

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

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

```
294   \let\@glo@check@sortallowed\@gobble
295 }
```

Set standard sort as the default:

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

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

```
297 \newcommand*{\glssortnumberfmt[1]}{%
298   \ifnum#1<100000 0\fi
299   \ifnum#1<10000 0\fi
300   \ifnum#1<1000 0\fi
301   \ifnum#1<100 0\fi
302   \ifnum#1<10 0\fi
303   \number#1%
304 }
```

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

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

Store entry information when it's defined.

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

Defined count register associated with the glossary.

```
307 \def\@gls@defsortcount##1{%
308   \expandafter\global
309   \expandafter\newcount\csname glossary@##1@sortcount\endcsname
310 }%
```

Increment count register associated with the glossary and use as the sort key.

```
311 \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.

```
312 \ifcsundef{glossary@##1@sortcount}%
313 {\@gls@defsortcount{##1}}%
314 {}%
315 \expandafter\global\expandafter
316 \advance\csname glossary@##1@sortcount\endcsname by 1\relax
317 \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
318   \expandafter\glssortnumberfmt
319   {\csname glossary@##1@sortcount\endcsname}}%
320 }%
```

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

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

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

```
322 \let\@glo@check@sortallowed\@gobble
323 }
```

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

```
324 \newcommand*{\@gls@setupsort@use}{%
```

Don't store entry information when it's defined.

```
325 \let\do@glo@storeentry\@gobble
```

Defined count register associated with the glossary.

```
326 \def\@gls@defsortcount##1{%  
327   \expandafter\global  
328   \expandafter\newcount\csname glossary@##1@sortcount\endcsname  
329   }%
```

Initialise the sort key to empty.

```
330 \def\@gls@defsort##1##2{%  
331   \expandafter\gdef\csname glo@##2@sort\endcsname{%}  
332   }%
```

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

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

Get the parent, if one exists

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

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

```
335   \ifx\@glo@parent\@empty  
336   \else  
337     \expandafter\@gls@setsort\expandafter{\@glo@parent}%  
338   \fi
```

Set index information for this entry

```
339   \protected@edef\@glo@type{\csname glo@##1@type\endcsname}%  
340   \protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%  
341   \ifx\@gls@tmp\@empty  
342     \expandafter\global\expandafter  
343     \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax  
344     \expandafter\protected@xdef\csname glo@##1@sort\endcsname{%  
345       \expandafter\glssortnumberfmt  
346       {\csname glossary@\@glo@type @sortcount\endcsname}}%  
347     \@glo@storeentry{##1}%  
348   \fi  
349   }%
```

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

```
350 \let\@glo@check@sortallowed\@gobble  
351 }
```

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

```
352 \newcommand*{\@gls@setupsort@none}{%
```

Don't store entry index information.

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

No count register required for standard sort.

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

Don't modify sort value.

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



```

356 \expandafter\global\expandafter\let\csname glo###2@sort\endcsname\@glo@sort
357 }%

```

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

```

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

```

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

```

359 \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
360 {Option sort=none not allowed with \string##1}%
361 {(Use sort=def instead)}}%
362 }

```

`\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.)

```

363 \newcommand*\glsdefmain{%
364 \if@gls@docloaded
365 \newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
366 \else
367 \newglossary{main}{gls}{glo}{\glossaryname}%
368 \fi

```

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

```

369 \newcommand*\gls@tr@set@main@toctitle{%
370 \translatelet{\glossarytoctitle}{Glossary}%
371 }%
372 }

```

Keep track of the default glossary. This is initialised to the main glossary, but can be changed if for some reason you want to make a secondary glossary the main glossary. This affects any commands that can optionally take a glossary name as an argument (or as the value of the type key in a key-value list). This was mainly done so that `\loadglsentries` can temporarily change `\glsdefaulttype` while it loads a file containing new glossary entries (see [section 1.10](#)).

`\glsdefaulttype`

```

373 \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`

```

374 \newcommand*\acronymtype{\glsdefaulttype}

```

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

```

375 \@gls@declareoption{nomain}{%
376 \let\glsdefaulttype\relax
377 \renewcommand*\glsdefmain}{%
378 }

```

acronym The acronym option sets an associated conditional which is used in [section 1.17](#) to determine whether or not to define a separate glossary for acronyms.

```

379 \define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
380   \ifglssacronym
381     \renewcommand{\@gls@do@acronymsdef}{%
382       \DeclareAcronymList{acronym}%
383       \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
384       \renewcommand*{\acronymtype}{acronym}%
385       \newcommand*{\@gls@tr@set@acronym@toctitle}{%
386         \translatelet{\glossarytoctitle}{Acronyms}%
387       }%
388     }%
389   \else
390     \let\@gls@do@acronymsdef\relax
391   \fi
392 }

```

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

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

```

393 \AtBeginDocument{%
394   \ifglssacronym
395     \ifbool{glscpatible-3.07}%
396     {}%
397     {%
398       \providecommand*{\printacronyms}[1][]{%
399         \printglossary[type=\acronymtype,#1]}%
400     }%
401   \fi
402 }

```

@do@acronymsdef Set default value

```

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

```

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

```

404 \@gls@declareoption{acronyms}{%
405   \glssacronymtrue
406   \def\@gls@do@acronymsdef{%
407     \DeclareAcronymList{acronym}%
408     \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
409     \renewcommand*{\acronymtype}{acronym}%

```

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

```

410     \newcommand*{\@gls@tr@set@acronym@toctitle}{%
411       \translatelet{\glossarytoctitle}{Acronyms}%
412     }%
413   }%
414 }

```

`glsacronymlists` Comma-separated list of glossary labels indicating which glossaries contain acronyms. Note that `\SetAcronymStyle` must be used after adding labels to this macro.

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

`addtoacronymlists`

```
416 \newcommand*{\@addtoacronymlists}[1]{%
417   \ifx\@glsacronymlists\@empty
418     \protected@xdef\@glsacronymlists{#1}%
419   \else
420     \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
421   \fi
422 }
```

`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. Use `\SetAcronymStyle` after identifying all the acronym lists.)

```
423 \newcommand*{\DeclareAcronymList}[1]{%
424   \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
425 }
```

`glsIfListOfAcronyms`

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

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

```
426 \newcommand{\glsIfListOfAcronyms}[1]{%
427   \protected@edef\@do@gls@islistofacronyms{%
428     \noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
429   \@do@gls@islistofacronyms
430 }
```

Internal command requires label and list to be expanded:

```
431 \newcommand{\@gls@islistofacronyms}[4]{%
432   \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
433     \def\@gls@before{##1}\def\@gls@after{##2}}%
434   \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
435   \ifx\@gls@after\@nnil
```

Not found

```
436   #4%
437   \else
```

Found

```
438   #3%
439   \fi
440 }
```

`glsisacronymlist` Convenient boolean.

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

`ckisacronymlist` Sets the above boolean if argument is a label representing a list of acronyms.

```
442 \newcommand*{\gls@ckisacronymlist}[1]{%
443   \glsIfListOfAcronyms{#1}%
444   {\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
445 }
```

`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.)

```
446 \newcommand*{\SetAcronymLists}[1]{%
447   \renewcommand*{\@glsacronymlists}{#1}%
448 }
```

`acronymlists`

```
449 \define@key{glossaries.sty}{acronymlists}{%
450   \DeclareAcronymList{#1}%
451 }
```

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 [section 1.6](#)).

`\glscounter`

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

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

```
453 \define@key{glossaries.sty}{counter}{%
454   \renewcommand*{\glscounter}{#1}%
455 }
```

`gls@nohyperlist`

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

`lareNoHyperList`

```
457 \newcommand*{\GlsDeclareNoHyperList}[1]{%
458   \ifdefempty\@gls@nohyperlist
459   {%
460     \renewcommand*{\@gls@nohyperlist}{#1}%
461   }%
462   {%
463     \appto\@gls@nohyperlist{,#1}%
464   }%
465 }
```

`nohypertypes`

```
466 \define@key{glossaries.sty}{nohypertypes}{%
467   \GlsDeclareNoHyperList{#1}%
468 }
```

`glossariesWarning` Prints a warning message.

```

469 \newcommand*{\GlossariesWarning}[1]{%
470   \PackageWarning{glossaries}{#1}%
471 }

```

`glossariesWarningNoLine` Prints a warning message without the line number.

```

472 \newcommand*{\GlossariesWarningNoLine}[1]{%
473   \PackageWarningNoLine{glossaries}{#1}%
474 }

```

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

```

475 \newcommand{\glosortentrieswarning}{%
476   \typeout{Using TeX to sort glossary entries---this may
477   take a while}%
478 }

```

`glossariesNowarn` Define package option to suppress warnings

```

479 \@gls@declareoption{nowarn}{%
480   \ifgls@debug
481     \GlossariesWarning{Warnings can't be suppressed in debug mode}%
482   \else
483     \renewcommand*{\GlossariesWarning}[1]{}%
484     \renewcommand*{\GlossariesWarningNoLine}[1]{}%
485     \renewcommand*{\glosortentrieswarning}{}%
486     \renewcommand*{\@gls@missinglang@warn}[2]{}%
487   \fi
488 }

```

`glossariesMissingLangWarn` Missing language warning.

```

489 \newcommand*{\@gls@missinglang@warn}[2]{%
490   \PackageWarningNoLine{glossaries}%
491   {No language module detected for '#1'.\MessageBreak
492   Language modules need to be installed separately.\MessageBreak
493   Please check on CTAN for a bundle called\MessageBreak
494   'glossaries-#2' or similar}%
495 }

```

`glossariesNoLangWarn` Suppress warning if language support not found.

```

496 \@gls@declareoption{nolangwarn}{%
497   \renewcommand*{\@gls@missinglang@warn}[2]{}%
498 }

```

`glossariesNonglossDefined` Issue a warning if overriding `\printglossary`

```

499 \newcommand*{\@gls@warnonglossdefined}{%
500   \GlossariesWarning{Overriding \string\printglossary}%
501 }

```

theglossdefined Issue a warning if overriding theglossary

```

502 \newcommand*{\@gls@warnontheglossdefined}{%
503   \GlossariesWarning{Overriding 'theglossary' environment}%
504 }
```

noredefwarn Suppress warning on redefinition of \printglossary

```

505 \@gls@declareoption{noredefwarn}{%
506   \renewcommand*{\@gls@warnonglossdefined}{}%
507   \renewcommand*{\@gls@warnontheglossdefined}{}%
508 }
```

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.

ls@sanitizedesc

```

509 \newcommand*{\@gls@sanitizedesc}{%
510 }
```

glsssetexpandfield `\glsssetexpandfield{<field>}`

Sets field to always expand.

```

511 \newcommand*{\glsssetexpandfield}[1]{%
512   \csdef{gls@assign@#1@field}##1##2{%
513     \@gls@expand@field{##1}{#1}{##2}%
514   }%
515 }
```

glsssetnoexpandfield `\glsssetnoexpandfield{<field>}`

Sets field to never expand.

```

516 \newcommand*{\glsssetnoexpandfield}[1]{%
517   \csdef{gls@assign@#1@field}##1##2{%
518     \@gls@noexpand@field{##1}{#1}{##2}%
519   }%
520 }
```

sign@type@field The type must always be expandable.

```

521 \glsssetexpandfield{type}
```

sign@desc@field The description is not expanded by default:

```

522 \glsssetnoexpandfield{desc}
```

descplural@field

```

523 \glsssetnoexpandfield{descplural}
```

ls@sanitizename

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

sign@name@field Don't expand name by default.

```
525 \glssetnoexpandfield{name}
```

@sanitizesymbol

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

gn@symbol@field Don't expand symbol by default.

```
527 \glssetnoexpandfield{symbol}
```

bolplural@field

```
528 \glssetnoexpandfield{symbolplural}
```

Sanitizing stuff:

ls@sanitizesort

```
529 \newcommand*{\@gls@sanitizesort}{%  
530   \ifglssanitizesort  
531     \@gls@sanitizesort  
532   \else  
533     \@gls@nosanitizesort  
534   \fi  
535 }
```

ls@sanitizesort

```
536 \newcommand*\@gls@sanitizesort{%  
537   \@onelevel@sanitize\@glo@sort  
538 }
```

@nosanitizesort

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

dx@sanitizesort Remove braces around first character (if present) before sanitizing.

```
540 \newcommand*\@gls@noidx@sanitizesort{%  
541   \ifdefvoid\@glo@sort  
542   {}%  
543   {%  
544     \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort  
545   }%  
546 }  
547 \def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%  
548   \def\@glo@sort{#1#2}%  
549   \@onelevel@sanitize\@glo@sort  
550 }
```

@nosanitizesort

```
551 \newcommand*{\@@gls@noidx@nosanitizesort}{%
552   \ifdefvoid\@glo@sort
553   {}%
554   {%
555     \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
556   }%
557 }
558 \def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
559   \bgroup
560     \glsnoidxstripaccents
561     \protected@xdef\@@glo@sort{#1#2}%
562   \egroup
563   \let\@glo@sort\@@glo@sort
564 }
```

idxstripaccents

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.

```
565 \newcommand*\glsnoidxstripaccents{%
566   \let\IeC\@firstofone
567   \let\add@accent\@secondoftwo
568   \let\@text@composite\@secondoftwo
569   \let\@tabacckludge\@secondoftwo
570   \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
571   \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
572   \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
573   \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
574   \let\' \@firstofone
575   \let\` \@firstofone
576   \let\^ \@firstofone
577   \let\" \@firstofone
578   \let\u \@firstofone
579   \let\t \@firstofone
580   \let\d \@firstofone
581   \let\r \@firstofone
582   \let\= \@firstofone
583   \let\.\@firstofone
584   \let\~ \@firstofone
585   \let\v \@firstofone
586   \let\H \@firstofone
587   \let\c \@firstofone
588   \let\b \@firstofone

589   \let\a \@secondoftwo
590   \def\AE{AE}%
591   \def\ae{ae}%
592   \def\OE{OE}%
```



```

593 \def\oe{oe}%
594 \def\AA{AA}%
595 \def\aa{aa}%
596 \def\L{L}%
597 \def\l{l}%
598 \def\O{O}%
599 \def\o{o}%
600 \def\SS{SS}%
601 \def\ss{ss}%
602 \def\th{th}%

603 \def\TH{TH}%
604 \def\dh{dh}%
605 \def\DH{DH}%
606 }

```

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

```

607 \@ifl@t@r\fmtversion{2019/10/01}
608 {%
609 \appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
610 }
611 {}

```

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.

```

612 \define@boolkey[glS]{sanitize}{description}[true]{%
613 \GlossariesWarning{sanitize={description} package option deprecated}%
614 \ifglS@sanitize@description
615 \glSsetnoexpandfield{desc}%
616 \glSsetnoexpandfield{descplural}%
617 \else
618 \glSsetexpandfield{desc}%
619 \glSsetexpandfield{descplural}%
620 \fi
621 }

622 \define@boolkey[glS]{sanitize}{name}[true]{%
623 \GlossariesWarning{sanitize={name} package option deprecated}%
624 \ifglS@sanitize@name
625 \glSsetnoexpandfield{name}%
626 \else
627 \glSsetexpandfield{name}%
628 \fi
629 }

630 \define@boolkey[glS]{sanitize}{symbol}[true]{%
631 \GlossariesWarning{sanitize={symbol} package option deprecated}%
632 \ifglS@sanitize@symbol
633 \glSsetnoexpandfield{symbol}%

```

```

634 \glsssetnoexpandfield{symbolplural}%
635 \else
636 \glsssetexpandfield{symbol}%
637 \glsssetexpandfield{symbolplural}%
638 \fi
639 }

```

sanitizesort

```

640 \define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
641 \ifglsssanitizesort
642 \glsssetnoexpandfield{sortvalue}%
643 \renewcommand*{\@gls@noidx@setsanitizesort}{%
644 \glsssanitizesorttrue
645 \glsssetnoexpandfield{sortvalue}%
646 }%
647 \else
648 \glsssetexpandfield{sortvalue}%
649 \renewcommand*{\@gls@noidx@setsanitizesort}{%
650 \glsssanitizesortfalse
651 \glsssetexpandfield{sortvalue}%
652 }%
653 \fi
654 }

```

Default setting:

```

655 \glsssanitizesorttrue
656 \glsssetnoexpandfield{sortvalue}%

```

setsanitizesort Default behaviour for \makenoidxglossaries is sanitizesort=false.

```

657 \newcommand*{\@gls@noidx@setsanitizesort}{%
658 \glsssanitizesortfalse
659 \glsssetexpandfield{sortvalue}%
660 }

661 \define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
662 \setbool{glsssanitizesort}{#1}%
663 \ifglsssanitizesort
664 \glsssetnoexpandfield{sortvalue}%
665 \else
666 \glsssetexpandfield{sortvalue}%
667 \fi
668 \GlossariesWarning{sanitize={sort} package option
669 deprecated. Use sanitizesort instead}%
670 }

```

sanitize

```

671 \define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
672 \ifthenelse{\equal{#1}{none}}{}%
673 {%
674 \GlossariesWarning{sanitize package option deprecated}%

```

```

675     \glssetexpandfield{name}%
676     \glssetexpandfield{symbol}%
677     \glssetexpandfield{symbolplural}%
678     \glssetexpandfield{desc}%
679     \glssetexpandfield{descplural}%
680 }%
681 {%
682     \setkeys[gls]{sanitize}{#1}%
683 }%
684 }

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

otranslatorhook \@gls@notranslatorhook has been removed.

s@usetranslator
686 \newcommand*\@gls@usetranslator{%
polyglossia tricks \@ifpackageloaded into thinking that babel has been loaded, so check for
polyglossia as well.
687 \@ifpackageloaded{polyglossia}%
688 {%
689     \let\glsifusetranslator\@secondoftwo
690 }%
691 {%
692     \@ifpackageloaded{babel}%
693     {%
694         \IfFileExists{translator.sty}%
695         {%
696             \RequirePackage{translator}%
697             \let\glsifusetranslator\@firstoftwo
698         }%
699     }%
700 }%
701 {}%
702 }%
703 }

dtranslatordict Checks if given translator dictionary has been loaded.
704 \newcommand{\glsifusedtranslatordict}[3]{%
705     \glsifusetranslator
706     {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
707     {#3}%
708 }

notranslate Provide a synonym for translate=false that can be passed via the document class.
709 \@gls@declareoption{notranslate}{%

```

```

710 \glstranslatefalse
711 \let\@gls@usetranslator\relax
712 \let\glsifusetranslator\@secondoftwo
713 }

```

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

```

714 \define@choicekey{glossaries.sty}{translate}%
715 [\gls@translate@val\gls@translate@nr]%
716 {true,false,babel}[true]%
717 {%
718   \ifcase\gls@translate@nr\relax
719     \glstranslatetrue
720     \renewcommand*\@gls@usetranslator{%
721       \@ifpackageloaded{polyglossia}%
722       {%
723         \let\glsifusetranslator\@secondoftwo
724       }%
725       {%
726         \@ifpackageloaded{babel}%
727         {%
728           \IfFileExists{translator.sty}%
729           {%
730             \RequirePackage{translator}%
731             \let\glsifusetranslator\@firstoftwo
732           }%
733         }%
734       }%
735     }%
736   }%
737 }%
738 \or
739   \glstranslatefalse
740   \let\@gls@usetranslator\relax
741   \let\glsifusetranslator\@secondoftwo
742 \or
743   \glstranslatetrue
744   \let\@gls@usetranslator\relax
745   \let\glsifusetranslator\@secondoftwo
746 \fi
747 }

```

Set the default value:

```

748 \glstranslatefalse
749 \let\glsifusetranslator\@secondoftwo
750 \@ifpackageloaded{translator}%
751 {%
752   \glstranslatetrue
753   \let\glsifusetranslator\@firstoftwo
754 }%

```

```

755 {%
756 \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
757 {
758 \ifpackageloaded{\gls@thissty}%
759 {%
760 \glstranslatetrue
761 \@endfortrue
762 }%
763 {}%
764 }
765 }

```

indexonlyfirst Set whether to only index on first use.

```

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

```

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

```

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

```

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

```

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

```

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

```

771 \define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
772 \ifbool{glsacrdescription}%
773 {}%
774 {}%
775 \renewcommand*{\@gls@sanitizedesc}{}%
776 }%
777 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
778 }

```

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

```

779 \define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
780 \renewcommand*{\@gls@sanitizesymbol}{}%
781 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
782 }

```

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

```

783 \define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
784 \renewcommand*{\@gls@sanitizesymbol}{}%
785 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
786 }

```

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

```

787 \define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
788   \renewcommand*{\@gls@sanitizesymbol}{}}%
789   \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
790 }

dua   Define \newacronym to always use the long forms (i.e. don't use acronyms)
791 \define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
792   \renewcommand*{\@gls@sanitizesymbol}{}}%
793   \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
794 }

shotcuts   Define acronym shortcuts.
795 \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.
796 \newcommand*{\glsorder}{word}

\@glsorder   The ordering information is written to the auxiliary file for makeglossaries, so ignore the
            auxiliary information.
797 \newcommand*{\@glsorder}[1]{}

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

\ifglxindy   Provide boolean to determine whether glxindy or makeindex will be used to sort the glossaries.
800 \newif\ifglxindy

            The default is makeindex:
801 \glxindyfalse

makeindex   Define package option to specify that makeindex will be used to sort the glossaries:
802 \@gls@declareoption{makeindex}{\glxindyfalse}

            The xindy package option may have a value which in turn can be a key=value list. First de-
            fine the keys for this sub-list. The boolean glsnumbers determines whether to automatically
            add the glsnumbers letter group.
803 \define@boolkey[gls]{xindy}{glsnumbers}[true]{}
804 \gls@xindy@glsnumberstrue

y@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.)
805 \def\@xdy@main@language{\language}%

            Define key to set the language
806 \define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}

```

`\gls@codepage` Define the code page. If `\inputencodingname` is defined use that, otherwise have initialise with no codepage.

```

807 \ifcsundef{inputencodingname}{%
808   \def\gls@codepage{}}{%
809   \def\gls@codepage{\inputencodingname}
810 }

```

Define a key to set the code page.

```

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

```

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

```

812 \define@key{glossaries.sty}{xindy}[]{%
813   \glsxindytrue
814   \setkeys[gls]{xindy}{#1}%
815 }

```

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

```

816 \@gls@declareoption{xindygloss}{%
817   \glsxindytrue
818 }

```

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

```

819 \@gls@declareoption{xindynoglsnumbers}{%
820   \glsxindytrue
821   \gls@xindy@glsnumbersfalse
822 }

```

`omakeglossaries`

```

823 \providecommand{\@domakeglossaries}[1]{#1}

```

`isablemakegloss` 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.

```

824 \@gls@declareoption{isablemakegloss}{%
825   \ifdefequal\makeglossaries\@no@makeglossaries
826   {%
827     \GlossariesWarning{Option ‘isablemakegloss’ has no effect
828       (\string\makenoidxglossaries\space has already been used)}%
829   }%
830   {%
831     \ifdefequal\@makeglossary\@gobble
832     {%
833       \GlossariesWarning{Option ‘isablemakegloss’ has no effect
834         (\string\makeglossaries\space has already been used)}%
835     }%
836   }%
837   \renewcommand{\@domakeglossaries}[1]{%

```

```

838         \PackageInfo{glossaries}{\string\makeglossaries\space and
839         \string\makenoidxglossaries\space have been disabled}%
840     }%
841 }%
842 }%
843 }

```

estoremakegloss Cancel the effect of disablemakegloss.

```

844 \@gls@declareoption{restoremakegloss}{%
845   \ifdefequal\makeglossaries\@no@makeglossaries
846   {%
847     \GlossariesWarning{Option ‘restoremakegloss’ has no effect
848     (\string\makenoidxglossaries\space has already been used)}%
849   }%
850   {%
851     \ifdefequal\@makeglossary\@gobble
852     {%
853       \GlossariesWarning{Option ‘restoremakegloss’ has no effect
854       (\string\makeglossaries\space has already been used)}%
855     }%
856     {%
857       \PackageInfo{glossaries}{\string\makeglossaries\space and
858       \string\makenoidxglossaries\space have been restored}%
859       \let\@domakeglossaries\@firstofone
860     }%
861   }%
862 }

```

write@glslabels

```

863 \newcommand*{\@do@write@glslabels}{%
864   \AtEndDocument{\@do@write@glslabels}%
865   \let\@do@write@glslabels\relax
866 }

```

write@glslabels

```

867 \newcommand*{\@do@write@glslabels}{%
868   \newwrite\@gls@labelsfile
869   \immediate\openout\@gls@labelsfile=\jobname.glslabels
870   \protected@edef\@gls@fulllist{\@glo@types}%
871   \ifdefempty{\@ignored@glossaries}{%
872     {\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
873   \forallglsentries[\@gls@fulllist]{\@glsentry}%
874   {%
875     \ifdefempty{\@glsentry}{%
876     {%
877       \ifx\do@write@glslabelsname\@gobble
878       \def\@gls@thisentryname{}%
879     \else
880       \letcs{\@gls@thisentryname}{glo@glsdetoklabel{\@glsentry}@name}%

```



```

881      \@onelevel@sanitize\@gls@thisentryname
882      \fi
883      \immediate\write\@gls@labelsfile
884      {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}}%
885  }%
886 }%
887 \immediate\closeout\@gls@labelsfile
888 }

```

`e@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.

```

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

```

`e@glslabelsname`

```

890 \begingroup
891 \catcode'\^^I12
892 \gdef\@write@glslabelsname#1{\^^I#1}
893 \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.

```

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

```

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

```

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

```

`\ifglsautomake`

```

898 \newif\ifglsautomake

```

`gls@automake@nr`

```

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

```

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

```

900 \define@choicekey{glossaries.sty}{automake}%
901 [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
902 \ifnum\gls@automake@nr=1\relax
903 \glsautomakefalse
904 \else
905 \glsautomaketrue
906 \fi
907 \ifglsautomake
908 \renewcommand*{\@gls@doautomake}{%
909 \PackageError{glossaries}{You must use
910 \string\makeglossaries\space with automake=true}

```

```

911      {%
912          Either remove the automake=true setting or
913          add \string\makeglossaries\space to your document preamble.%
914      }%
915  }%
916  \else
917      \renewcommand*{\@gls@doautomake}{}%
918  \fi
919 }
920 \glsautomakefalse

@gls@doautomake
921 \newcommand*{\@gls@doautomake}{}
922 \AtEndDocument{\@gls@doautomake}

savewrites The savewrites package option is provided to save on the number of write registers.
923 \define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
924     \ifglssavewrites
925         \renewcommand*{\glswritefiles}{\@glswritefiles}%
926     \else
927         \let\glswritefiles\@empty
928     \fi
929 }

Set default:
930 \glssavewritesfalse
931 \let\glswritefiles\@empty

compatible-3.07
932 \define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{}
933 \boolfalse{glscompatible-3.07}

compatible-2.07
934 \define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
    Also set 3.07 compatibility if this option is set.
935     \ifbool{glscompatible-2.07}%
936     {%
937         \booltrue{glscompatible-3.07}%
938     }%
939     {}%
940 }
941 \boolfalse{glscompatible-2.07}

al@makeglossary Store the original definition.
942 \let\gls@original@makeglossary\makeglossary

iginal@glossary Store the original definition.
943 \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.)

```

944 \def\makeglossary{%
945   \GlossariesWarning{Use of \string\makeglossary\space with
946   glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
947   instead. If you \MessageBreak need the original definition of
948   \string\makeglossary\space use \MessageBreak the package options
949   kernelglossredefs=false (to \MessageBreak restore the former definition of
950   \string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
951   conflict)}}%
952   \makeglossaries
953 }

```

`\erride@glossary`

```

954 \newcommand*{\@gls@override@glossary}[1][main]{%
955   \GlossariesWarning{Use of \string\glossary\space with
956   glossaries.sty is deprecated. \MessageBreak Indexing should be performed
957   with the user level \MessageBreak commands, such as \string\gls\space or
958   \string\glsadd. If you need the \MessageBreak original definition of
959   \string\glossary\space use the package \MessageBreak options
960   kernelglossredefs=false (to restore the \MessageBreak former definition of
961   \string\glossary) and nomain (if the \MessageBreak file extensions cause a
962   conflict)}}%
963   \gls@glossary{#1}%
964 }

```

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`

```

965 \if@gls@docloaded
966 \else
967   \def\glossary{\@gls@override@glossary}
968 \fi

```

`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.

```

969 \define@choicekey{glossaries.sty}{kernelglossredefs}%
970 [\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
971 {%
972   \ifcase\gls@debug@nr\relax
973   \def\glossary{\@gls@override@glossary}%
974   \def\makeglossary{%
975     \GlossariesWarning{Use of \string\makeglossary\space with
976       glossaries.sty is deprecated. Use \string\makeglossaries\space
977       instead. If you need the original definition of
978       \string\makeglossary\space use the package options
979       kernelglossredefs=false (to prevent redefinition of
980       \string\makeglossary) and nomain (if the file extensions cause a
981       conflict)}}%
982   \makeglossaries
983   }%
984   \or
985   \let\glossary\gls@original@glossary
986   \let\makeglossary\gls@original@makeglossary
987   \or
988   \def\makeglossary{\makeglossaries}%
989   \renewcommand*{\@gls@override@glossary}[1][main]{%
990     \gls@glossary{##1}%
991   }%
992   \fi
993 }

```

symbols Create a “symbols” glossary type

```

994 \@gls@declareoption{symbols}{%
995   \let\@gls@do@symbolsdef\@gls@symbolsdef
996 }

```

Default is not to define the symbols glossary:

```

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

```

@gls@symbolsdef

```

998 \newcommand*{\@gls@symbolsdef}{%
999   \newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
1000   \newcommand*{\printsymbols}[1][ ]{\printglossary[type=symbols,##1]}%

```

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

```

1001   \newcommand*{\gls@tr@set@symbols@toctitle}{%
1002     \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
1003   }%
1004 }%

```

numbers Create a “symbols” glossary type

```

1005 \@gls@declareoption{numbers}{%
1006   \let\@gls@do@numbersdef\@gls@numbersdef
1007 }

```

Default is not to define the numbers glossary:

```
1008 \newcommand*{\@gls@do@numbersdef}{}
```

@gls@numbersdef

```
1009 \newcommand*{\@gls@numbersdef}{%
1010   \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
1011   \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]%

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

1012   \newcommand*{\gls@tr@set@numbers@toctitle}{%
1013     \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
1014   }%
1015 }
```

index Create an “index” glossary type

```
1016 \@gls@declareoption{index}{%
1017   \ifx\@gls@do@indexdef\@empty
1018     \let\@gls@do@indexdef\@gls@indexdef
1019   \fi
1020 }
```

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

```
1021 \@gls@declareoption{noglossaryindex}{%
1022   \let\@gls@do@indexdef\relax
1023 }
```

Default is not to define index glossary:

```
1024 \newcommand*{\@gls@do@indexdef}{}
```

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

```
1025 \newcommand*{\@gls@indexdef}{%
1026   \newglossary[ilg]{index}{ind}{idx}{\indexname}%
1027   \newcommand*{\printindex}[1][\printglossary[type=index,##1]]%
1028   \newcommand*{\newterm}[2][\%
1029     \newglossaryentry{##2}%
1030     {type={index},name={##2},description={\nopostdesc},##1}}
1031   \let\@gls@do@indexdef\relax
1032 }
```

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

```
1033 \@for\CurrentOption :=\@declaredoptions\do{%
1034   \ifx\CurrentOption\@empty
1035   \else
1036     \@expandtwoargs
1037     \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
1038     \ifin@
1039     \@use@ption
```

```

1040     \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
1041   \fi
1042 \fi
1043 }

```

Now process options passed to the package:

```
1044 \ProcessOptionsX
```

Load backward compatibility stuff:

```
1045 \RequirePackage{glossaries-compatible-307}
```

`\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:

```

1046 \disable@keys{glossaries.sty}{compatible-2.07,%
1047 xindy,xindygloss,xindynoglsnumbers,makeindex,%
1048 acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
1049 nomain,noglossaryindex}

```

Now define `\setupglossaries`:

```

1050 \newcommand*{\setupglossaries}[1]{%
1051   \renewcommand*{\@gls@setacrstyle}{}%
1052   \ifglsacrshortcuts
1053     \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
1054   \else
1055     \def\@gls@setupshortcuts{%
1056       \ifglsacrshortcuts
1057         \DefineAcronymSynonyms
1058       \fi
1059     }%
1060   \fi
1061   \glsacrshortcutsfalse
1062   \let\@gls@do@numbersdef\relax
1063   \let\@gls@do@symbolssdef\relax
1064   \let\@gls@do@indexdef\relax
1065   \let\@gls@do@acronymsdef\relax
1066   \ifglsentrycounter
1067     \let\@gls@doentrycounterdef\relax
1068   \else
1069     \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
1070   \fi
1071   \ifglssubentrycounter
1072     \let\@gls@dosubentrycounterdef\relax
1073   \else
1074     \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
1075   \fi
1076   \setkeys{glossaries.sty}{#1}%
1077   \@gls@setacrstyle
1078   \@gls@setupshortcuts
1079   \@gls@do@acronymsdef
1080   \@gls@do@numbersdef

```

```

1081 \@gls@do@symbolssdef
1082 \@gls@do@indexdef
1083 \@gls@doentrycounterdef
1084 \@gls@dosubentrycounterdef
1085 }

```

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}`).

```

1086 \ifthenelse{\equal{\glscounter}{section}}{%
1087 {%
1088   \ifcsundef{chapter}{}%
1089   {%
1090     \let\@gls@old@chapter\@chapter
1091     \def\@chapter[#1]#2{\@gls@old@chapter[#1]{#2}%
1092       \ifcsundef{hyperdef}{}{\hyperdef{section}{\thesection}{}}}%
1093   }%
1094 }%
1095 {}

```

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

```

1096 \newcommand*{\@gls@onlypremakeg}{}

```

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

```

1097 \newcommand*{\@onlypremakeg}[1]{%
1098   \ifx\@gls@onlypremakeg\@empty
1099     \def\@gls@onlypremakeg{#1}%
1100   \else
1101     \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
1102     \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
1103   \fi
1104 }

```

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

```

1105 \newcommand*{\@disable@onlypremakeg}{%
1106   \@for\@thiscs:=\@gls@onlypremakeg\do{%
1107     \expandafter\@disable@premakecs\@thiscs%
1108   }}

```

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

```

1109 \newcommand*{\@disable@premakecs}[1]{%

```

```

1110 \def#1{\PackageError{glossaries}{\string#1\space may only be
1111 used before \string\makeglossaries}{You can't use
1112 \string#1\space after \string\makeglossaries}}%
1113 }

```

1.3 Predefined Text

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

Main glossary title:

`\glossaryname`

```
1114 \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`

```
1115 \providecommand*\acronymname{Acronyms}
```

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

```

1116 \newcommand*\glstocctitle[1]{%
1117 \def\glossarytocctitle{\csname @glotype@#1@title\endcsname}}

```

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

`\entryname`

```
1118 \providecommand*\entryname{Notation}
```

`\descriptionname`

```
1119 \providecommand*\descriptionname{Description}
```

`\symbolname`

```
1120 \providecommand*\symbolname{Symbol}
```

`\pagelistname`

```
1121 \providecommand*\pagelistname{Page List}
```

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

`\symbolsgroupname`

```
1122 \providecommand*\glsymbolsgroupname{Symbols}
```

`\numbersgroupname`

```
1123 \providecommand*\glsnumbersgroupname{Numbers}
```


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

```
1124 \newcommand*{\glspluralsuffix}{s}
```

`acrpluralsuffix` Default plural suffix for acronyms

```
1125 \newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
```

`acrpluralsuffix`

```
1126 \newcommand*{\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
```

`\seename`

```
1127 \providecommand*{\seename}{see}
```

`\andname`

```
1128 \providecommand*{\andname}{\&}
```

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

`eGlossariesLang`

```
1129 \newcommand*{\RequireGlossariesLang}[1]{%
1130   \@ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
1131 }
```

`sGlossariesLang`

```
1132 \newcommand*{\ProvidesGlossariesLang}[1]{%
1133   \ProvidesFile{glossaries-#1.ldf}%
1134 }
```

`ssarytocaptions` Does nothing if translator hasn't been loaded.

```
1135 \newcommand*{\addglossarytocaptions}[1]{}
```

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

```
1136 \ifglstranslate
```

Load tracklang

```
1137 \RequirePackage{tracklang}
```

Load translator if required.

```
1138 \@gls@usetranslator
```

If using `\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 doesn't define it.)

```
1139 \@ifpackageloaded{translator}
1140   {%
```

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.

```

1141 \ifboolexpr
1142 {
1143   test {\ifdefstring{\trans@languages}{English}}
1144   and not
1145   test {\ifdefstring{\bbl@loaded}{english}}
1146 }
1147 {%
1148   \let\glsifusetranslator\@secondoftwo
1149 }%
1150 {%
1151   \usedictionary{glossaries-dictionary}%
1152   \renewcommand*{\addglossarytocaptions}[1]{%
1153     \ifcsundef{captions#1}{}%
1154     {%
1155       \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
1156       \expandafter\toks@\expandafter{\@gls@tmp
1157         \renewcommand*{\glossaryname}{\translate{Glossary}}}%
1158     }%
1159     \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
1160   }%
1161 }%
1162 }%
1163 }%
1164 {}%

```

Check for tracked languages

```

1165 \AnyTrackedLanguages
1166 {%
1167   \ForEachTrackedDialect{\this@dialect}{%
1168     \IfTrackedLanguageFileExists{\this@dialect}%
1169     {glossaries-}% prefix
1170     {.ldf}%
1171     {%
1172       \RequireGlossariesLang{\CurrentTrackedTag}%
1173     }%
1174     {%
1175       \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
1176     }%
1177   }%
1178 }%
1179 {}%

```

if using translator use translator interface.

```

1180 \glsifusetranslator
1181 {%
1182   \renewcommand*{\glssettoctitle}[1]{%

```

```

1183     \ifcsdef{gls@tr@set@#1@toctitle}%
1184     {%
1185         \csuse{gls@tr@set@#1@toctitle}%
1186     }%
1187     {%
1188         \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
1189     }%
1190 }%
1191 \renewcommand*{\glossaryname}{\translate{Glossary}}%
1192 \renewcommand*{\acronymname}{\translate{Acronyms}}%
1193 \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
1194 \renewcommand*{\descriptionname}{%
1195     \translate{Description (glossaries)}}%
1196 \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
1197 \renewcommand*{\pagelistname}{%
1198     \translate{Page List (glossaries)}}%
1199 \renewcommand*{\glssymbolsgroupname}{%
1200     \translate{Symbols (glossaries)}}%
1201 \renewcommand*{\glsnumbersgroupname}{%
1202     \translate{Numbers (glossaries)}}%
1203 }{}%
1204 \fi

```

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

```
1205 \DeclareRobustCommand*{\nopostdesc}{}

```

`\@nopostdesc` Suppress next description terminator.

```

1206 \newcommand*{\@nopostdesc}{%
1207     \let\org@glspostdescription\glspostdescription
1208     \def\glspostdescription{%
1209         \let\glspostdescription\org@glspostdescription}%
1210 }

```

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

```
1211 \newcommand*{\@no@post@desc}{\nopostdesc}

```

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

```
1212 \newcommand{\glspar}{\par}

```

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

```

1213 \newcommand{\setStyleFile}[1]{%
1214     \renewcommand*{\gls@istfilebase}{#1}%
1215     Just in case \istfilename has been modified.
1216     \ifglxindy
1217         \def\istfilename{\gls@istfilebase.xdy}
1218     \else
1219         \def\istfilename{\gls@istfilebase.ist}

```

```
1219 \fi
1220 }
```

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

```
1221 \@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`

```
1222 \ifglxindy
1223   \def\istfilename{\gls@istfilebase.xdy}
1224 \else
1225   \def\istfilename{\gls@istfilebase.ist}
1226 \fi
```

`gls@istfilebase`

```
1227 \newcommand*{\gls@istfilebase}{\jobname}
```

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

`\@istfilename`

```
1228 \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`

```
1229 \newcommand*{\glscompositor}{.}
```

`lsSetCompositor` Sets the compositor.

```
1230 \newcommand*{\glsSetCompositor}[1]{%
1231   \renewcommand*{\glscompositor}{#1}}
```

Only use before `\makeglossaries`

```
1232 \@onlypremakeg\glsSetCompositor
```

(The page compositor is usually defined as a dash when using `makeindex`, but most of the standard counters used by \TeX 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.

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

```
1233 \newcommand*{\@glsAlphaCompositor}{\glscompositor}
```

AlphaCompositor Sets the alpha compositor.

```
1234 \ifglsxindy
1235   \newcommand*\glsSetAlphaCompositor[1]{%
1236     \renewcommand*\@glsAlphaCompositor{#1}}
1237 \else
1238   \newcommand*\glsSetAlphaCompositor[1]{%
1239     \glsnoxywarning\glsSetAlphaCompositor}
1240 \fi
```

Can only be used before `\makeglossaries`

```
1241 \@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.

```
1242 \newcommand*{\gls@suffixF}{}
```

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

```
1243 \newcommand*{\glsSetSuffixF}[1]{%
1244   \renewcommand*{\gls@suffixF}{#1}}
```

Only has an effect when used before `\makeglossaries`

```
1245 \@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.

```
1246 \newcommand*{\gls@suffixFF}{}
```

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

```
1247 \newcommand*{\glsSetSuffixFF}[1]{%
1248   \renewcommand*{\gls@suffixFF}{#1}%
1249 }
```

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”.

```
1250 \ifcsundef{hyperlink}%
1251 {%
1252   \newcommand*\glsnumberformat[1]{#1}%
1253 }%
1254 {%
```

```

1255 \newcommand*{\glsnumberformat}[1]{\glshyphnumber{#1}}%
1256 }

```

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
1257 \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
1258 \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
1259 \newcommand*{\glossarypreamble}{%
1260   \csuse{@glossarypreamble@\currentglossary}%
1261 }

```

```

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

```

Code provided by Michael Pock.

```

1262 \newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
1263   \ifglossaryexists*{#1}%
1264   {\csgdef{@glossarypreamble@#1}{#2}}%
1265   {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
1266 }

```

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

```
1267 \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`.

```
1268 \newcommand*{\glossarysection}[2][\@gls@title]{%
1269   \def\@gls@title{#2}%
1270   \ifcsundef{phantomsection}%
1271   {%
1272     \@glossarysection{#1}{#2}%
1273   }%
1274   {%
1275     \p@glossarysection{#1}{#2}%
1276   }%

1277   \glsglossarymark{\glossarytoctitle}%
1278 }
```

glsglossarymark Sets the header mark for the glossary. Takes the glossary short (TOC) title as the argument.

```
1279 \ifcsundef{glossarymark}%
1280 {%
1281   \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
1282 }%
1283 {%
1284   \@ifclassloaded{memoir}
1285   {%
1286     \newcommand{\glsglossarymark}[1]{%
1287       \ifglsucmark
1288         \markboth{\memUHead{#1}}{\memUHead{#1}}%
1289       \else
1290         \markboth{#1}{#1}%
1291       \fi
1292     }
1293   }%
1294   {%
1295     \newcommand{\glsglossarymark}[1]{%
1296       \ifglsucmark
1297         \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
1298       \else
1299         \@mkboth{#1}{#1}%
1300       \fi
1301     }
1302   }
1303 }
```

`\glossarymark` Provided for backward compatibility:

```

1304 \providecommand{\glossarymark}[1]{%
1305   \ifglsucmark
1306     \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
1307   \else
1308     \@mkboth{#1}{#1}%
1309   \fi
1310 }

```

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.

`\glossarysection`

```

1311 \newcommand*{\setglossarysection}[1]{%
1312 \setkeys{glossaries.sty}{section=#1}}

```

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

`\glossarysection`

```

1313 \newcommand*{\@glossarysection}[2]{%
1314   \ifdefempty\@glossarysecstar
1315   {%
1316     \csname\@glossarysec\endcsname[#1]{#2}%
1317   }%
1318   {%
1319     \csname\@glossarysec\endcsname*{#2}%
1320     \@gls@toc{#1}{\@glossarysec}%
1321   }%

```

Do automatic labelling if required

```

1322   \@glossaryseclabel
1323 }

```

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.

`\glossarysection`

```

1324 \newcommand*{\@pglossarysection}[2]{%
1325   \glsclearpage
1326   \phantomsection
1327   \ifdefempty\@glossarysecstar
1328   {%
1329     \csname\@glossarysec\endcsname{#2}%
1330   }%
1331   {%
1332     \@gls@toc{#1}{\@glossarysec}%
1333     \csname\@glossarysec\endcsname*{#2}%
1334   }%

```


Do automatic labelling if required

```
1335 \@@glossaryseclabel
1336 }
```

`\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.

```
1337 \newcommand*{\gls@doclearpage}{%
1338   \ifthenelse{\equal{\@@glossarysec}{chapter}}{%
1339     {%
1340       \ifcsundef{cleardoublepage}%
1341       {%
1342         \clearpage
1343       }%
1344     {%
1345       \ifcsdef{if@openright}%
1346       {%
1347         \if@openright
1348         \cleardoublepage
1349       \else
1350       \clearpage
1351     \fi
1352   }%
1353   {%
1354     \cleardoublepage
1355   }%
1356 }%
1357 }%
1358 {}%
1359 }
```

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

```
1360 \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`

```
1361 \newcommand*{\@gls@toc}[2]{%
1362   \ifglstoc
1363   \ifglsnumberline
1364     \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
1365   \else
1366     \addcontentsline{toc}{#2}{#1}%
1367   \fi
1368   \fi
1369 }
```

1.4 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 re-defining `\glsnoxindywarning` to ignore its argument

```
1370 \newcommand*{\glsnoxindywarning}[1]{%
1371   \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
1372 }
```

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

```
1373 \newcommand*{\glsnoindexwarning}[1]{%
1374   \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
1375 }
```

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

```
1376 \ifglsxindy
1377   \edef\@xdyattributes{\string"default\string"}%
1378 \fi
```

\@xdyattributelist Comma-separated list of attributes.

```
1379 \ifglsxindy
1380   \edef\@xdyattributelist{}%
1381 \fi
```

\@xdylocref Define list of markup location references.

```
1382 \ifglsxindy
1383   \def\@xdylocref{}
1384 \fi
```

\@gls@ifinlist

```
1385 \newcommand*{\@gls@ifinlist}[4]{%
1386   \def\@do@ifinlist##1,#1,##2\end@do@ifinlist{%
1387     \def\@gls@listsuffix{##2}%
1388     \ifx\@gls@listsuffix\@empty
1389       #4%
1390     \else
1391       #3%
1392     \fi
1393   }%
1394   \@do@ifinlist,#2,#1,\end@do@ifinlist
1395 }
```

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.

```
1396 \ifglsxindy
1397   \newcommand*{\@xdycounters}{\@glscounter}
```

```

1398 \newcommand*\GlsAddXdyCounters[1]{%
1399   \for\@gls@ctr:=#1\do{%
    Check if already in list before adding.
1400     \edef\@do@addcounter{%
1401       \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
1402       {%
1403         \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
1404           \noexpand\@gls@ctr}%
1405       }%
1406     }%
1407     \@do@addcounter
1408   }
1409 }

    Only has an effect before \writeist:
1410 \@onlypremakeg\GlsAddXdyCounters
1411 \else
1412 \newcommand*\GlsAddXdyCounters[1]{%
1413   \glsnoxindywarning\GlsAddXdyAttribute
1414 }
1415 \fi

```

saddxdycounters Counters must all be identified before adding attributes.

```

1416 \newcommand*\@disabled@glssaddxdycounters{%
1417   \PackageError{glossaries}{\string\GlsAddXdyCounters\space
1418     can't be used after \string\GlsAddXdyAttribute}{Move all
1419     occurrences of \string\GlsAddXdyCounters\space before the first
1420     instance of \string\GlsAddXdyAttribute}%
1421 }

```

AddXdyAttribute Adds an attribute.

```

1422 \ifglxsindy
    First define internal command that adds an attribute for a given counter (2nd argument is
    the counter):
1423 \newcommand*\@glssaddxdyattribute[2]{%
    Add to xindy attribute list
1424   \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
1425     \string"#2#1\string"}%
    Add to xindy markup location.
1426   \expandafter\toks@\expandafter{\@xdylocref}%
1427   \edef\@xdylocref{\the\toks@ ^^J%
1428     (markup-locref
1429     :open \string"\glstildechar n%
1430     \expandafter\string\csname glsX#2X#1\endcsname
1431     \string" ^^J
1432     :close \string"\string" ^^J
1433     :attr \string"#2#1\string"))%

```

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

```
1434 \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
1435   \setentrycounter{##1}{#2}\csname #1\endcsname{##2}%
1436   }%
1437 }
```

High-level command:

```
1438 \newcommand*\GlsAddXdyAttribute[1]{%
```

Add to comma-separated attribute list

```
1439 \ifx\@xdyattributelist\@empty
1440   \edef\@xdyattributelist{#1}%
1441 \else
1442   \edef\@xdyattributelist{\@xdyattributelist,#1}%
1443 \fi
```

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

```
1444 \@for\@this@counter:=\@xdycounters\do{%
1445   \protected@edef\gls@do@addxdyattribute{%
1446     \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
1447   }
1448   \gls@do@addxdyattribute
1449 }%
```

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

```
1450 \let\GlsAddXdyCounters\@disabled@glsaddxdycounters
1451 }
```

Only has an effect before `\writeist`:

```
1452 \@onlypremakeg\GlsAddXdyAttribute
1453 \else
1454 \newcommand*\GlsAddXdyAttribute[1]{%
1455   \glsnoxindywarning\GlsAddXdyAttribute}
1456 \fi
```

`\definedattributes` Add known attributes for all defined counters

```
1457 \ifglxindy
1458 \newcommand*\@gls@addpredefinedattributes{%
1459   \GlsAddXdyAttribute{glsnumberformat}
1460   \GlsAddXdyAttribute{textrm}
1461   \GlsAddXdyAttribute{textsf}
1462   \GlsAddXdyAttribute{texttt}
1463   \GlsAddXdyAttribute{textbf}
1464   \GlsAddXdyAttribute{textmd}
1465   \GlsAddXdyAttribute{textit}
1466   \GlsAddXdyAttribute{textup}
1467   \GlsAddXdyAttribute{textsl}
1468   \GlsAddXdyAttribute{textsc}
1469   \GlsAddXdyAttribute{emph}
1470   \GlsAddXdyAttribute{glshypernumber}
1471   \GlsAddXdyAttribute{hyperrm}
```

```

1472 \GlsAddXdyAttribute{hypersf}
1473 \GlsAddXdyAttribute{hypertt}
1474 \GlsAddXdyAttribute{hyperbf}
1475 \GlsAddXdyAttribute{hypermd}
1476 \GlsAddXdyAttribute{hyperit}
1477 \GlsAddXdyAttribute{hyperup}
1478 \GlsAddXdyAttribute{hypersl}
1479 \GlsAddXdyAttribute{hypersc}
1480 \GlsAddXdyAttribute{hyperemph}

1481 \GlsAddXdyAttribute{glignore}
1482 }
1483 \else
1484 \let\@gls@addpredefinedattributes\relax
1485 \fi

```

`\dyuseralphabets` List of additional alphabets

```
1486 \def\@xdyuseralphabets{}
```

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

```

1487 \ifglsxindy
1488 \newcommand*{\GlsAddXdyAlphabet}[2]{%
1489 \edef\@xdyuseralphabets{%
1490 \@xdyuseralphabets ^^J
1491 (define-alphabet "#1" (#2))}}
1492 \else
1493 \newcommand*{\GlsAddXdyAlphabet}[2]{%
1494 \glsnoxindywarning\GlsAddXdyAlphabet}
1495 \fi

```

This code is only required for xindy:

```
1496 \ifglsxindy
```

`\dy@locationlist` List of predefined location names.

```

1497 \newcommand*{\@gls@xdy@locationlist}{%
1498 roman-page-numbers,%
1499 Roman-page-numbers,%
1500 arabic-page-numbers,%
1501 alpha-page-numbers,%
1502 Alpha-page-numbers,%
1503 Appendix-page-numbers,%
1504 arabic-section-numbers%
1505 }

```

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

an-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.

```

1506 \protected@edef\@gls@roman{\@roman{0}\string"
1507 \string"roman-numbers-lowercase\string" :sep \string"}}%
1508 \@onelevel@sanitize\@gls@roman
1509 \edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
1510 :sep \string"}%
1511 \@onelevel@sanitize\@gls@tmp
1512 \ifx\@gls@tmp\@gls@roman
1513 \expandafter
1514 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
1515 \string"roman-numbers-lowercase\string"%
1516 }%
1517 \else
1518 \expandafter
1519 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
1520 :sep \string"\@gls@roman\string"%
1521 }%
1522 \fi

```

an-page-numbers Upper case Roman numerals (I, II, ...).

```

1523 \expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
1524 \string"roman-numbers-uppercase\string"%
1525 }%

```

ic-page-numbers Arabic numbers (1, 2, ...).

```

1526 \expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
1527 \string"arabic-numbers\string"%
1528 }%

```

ha-page-numbers Lower case alphabetical (a, b, ...).

```

1529 \expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
1530 \string"alpha\string"%
1531 }%

```

ha-page-numbers Upper case alphabetical (A, B, ...).

```

1532 \expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
1533 \string"ALPHA\string"%
1534 }%

```

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

```

1535 \expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
1536 \string"ALPHA\string"
1537 :sep \string"\@glsAlphacompositor\string"
1538 \string"arabic-numbers\string"%
1539 }

```

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

```

1540 \expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
1541   \string"arabic-numbers\string"
1542   :sep \string"\glscompositor\string"
1543   \string"arabic-numbers\string"%
1544   }%

```

erlocationdefs List of additional location definitions (separated by ^^J)

```

1545 \def\@xdyuserlocationdefs{}

```

erlocationnames List of additional user location names

```

1546 \def\@xdyuserlocationnames{}

```

End of xindy-only block:

```

1547 \fi

```

xdycrossrefhook Hook used after writing cross-reference class information.

```

1548 \ifglsxindy
1549 \newcommand\@xdycrossrefhook{}
1550 \fi

```

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

```

1551 \ifglsxindy
1552 \newcommand*\GlsAddXdyLocation[3][{}]{%
1553   \def\@gls@tmp{#1}%
1554   \ifx\@gls@tmp\@empty
1555     \edef\@xdyuserlocationdefs{%
1556       \@xdyuserlocationdefs ^^J%
1557       (define-location-class \string"#2\string"^^J\space\space
1558       \space(:sep \string"{}\glsopenbrace\string" #3
1559       :sep \string"\glsclosebrace\string"))
1560     }%
1561   \else
1562     \edef\@xdyuserlocationdefs{%
1563       \@xdyuserlocationdefs ^^J%
1564       (define-location-class \string"#2\string"^^J\space\space
1565       \space(:sep "\glsopenbrace"
1566       #1
1567       :sep "\glsclosebrace\glsopenbrace" #3
1568       :sep "\glsclosebrace"))
1569     }%
1570   \fi

1571   \edef\@xdyuserlocationnames{%
1572     \@xdyuserlocationnames^^J\space\space\space
1573     \string"#2\string"}%
1574 }

```

Only has an effect before \writeist:

```
1575 \onlypremakeg\GlsAddXdyLocation
1576 \else
1577 \newcommand*{\GlsAddXdyLocation}[2]{%
1578 \glsnnoxindywarning\GlsAddXdyLocation}
1579 \fi
```

ationclassorder Define location class order

```
1580 \ifglxindy
1581 \def\@xdylocationclassorder{^^J\space\space\space
1582 \string"roman-page-numbers\string"^^J\space\space\space
1583 \string"arabic-page-numbers\string"^^J\space\space\space
1584 \string"arabic-section-numbers\string"^^J\space\space\space
1585 \string"alpha-page-numbers\string"^^J\space\space\space
1586 \string"Roman-page-numbers\string"^^J\space\space\space
1587 \string"Alpha-page-numbers\string"^^J\space\space\space
1588 \string"Appendix-page-numbers\string"
1589 \@xdyuserlocationnames^^J\space\space\space
1590 \string"see\string"
1591 }
1592 \fi
```

Change the location order.

ationClassOrder

```
1593 \ifglxindy
1594 \newcommand*\GlsSetXdyLocationClassOrder[1]{%
1595 \def\@xdylocationclassorder{#1}}
1596 \else
1597 \newcommand*\GlsSetXdyLocationClassOrder[1]{%
1598 \glsnnoxindywarning\GlsSetXdyLocationClassOrder}
1599 \fi
```

\@xdysortrules Define sort rules

```
1600 \ifglxindy
1601 \def\@xdysortrules{}
1602 \fi
```

\GlsAddSortRule Add a sort rule

```
1603 \ifglxindy
1604 \newcommand*\GlsAddSortRule[2]{%
1605 \expandafter\toks@\expandafter{\@xdysortrules}%
1606 \protected@edef\@xdysortrules{\the\toks@ ^^J
1607 (sort-rule \string"#1\string" \string"#2\string")}%
1608 }
1609 \else
1610 \newcommand*\GlsAddSortRule[2]{%
1611 \glsnnoxindywarning\GlsAddSortRule}
1612 \fi
```


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

```
1613 \ifglxindy
1614   \def\xdyrequiredstyles{tex}
1615 \fi
```

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

```
1616 \ifglxindy
1617   \newcommand*\GlsAddXdyStyle[1]{%
1618     \edef\xdyrequiredstyles{\xdyrequiredstyles,#1}}%
1619 \else
1620   \newcommand*\GlsAddXdyStyle[1]{%
1621     \glsnnoxindywarning\GlsAddXdyStyle}
1622 \fi
```

`GlsSetXdyStyles` Reset the list of required styles

```
1623 \ifglxindy
1624   \newcommand*\GlsSetXdyStyles[1]{%
1625     \edef\xdyrequiredstyles{#1}}
1626 \else
1627   \newcommand*\GlsSetXdyStyles[1]{%
1628     \glsnnoxindywarning\GlsSetXdyStyles}
1629 \fi
```

`\findrootlanguage` This used to determine the root language, using a bit of trickery since babel doesn't supply the information, but now that babel is once again actively maintained, we can't do this any more, so `\findrootlanguage` is no longer available. Now provide a command that does nothing (in case it's been patched), but this may be removed completely in the future.

```
1630 \newcommand*\findrootlanguage{}
```

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

```
1631 \def\@xdylanguage#1#2{}
```

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

```
1632 \ifglxindy
1633   \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
1634     \ifglossaryexists{#1}{%
1635       \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
1636     }{%
1637       \PackageError{glossaries}{Can't set language type for
1638         glossary type '#1' --- no such glossary}{%
1639         You have specified a glossary type that doesn't exist}}
1640 \else
1641   \newcommand*\GlsSetXdyLanguage[2][]{}
```

```

1642 \glsnoxywarning\GlsSetXdyLanguage}
1643 \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.

```

1644 \def\@gls@codepage#1#2{}

```

`sSetXdyCodePage` Define command to set the code page.

```

1645 \ifglsxindy
1646 \newcommand*\GlsSetXdyCodePage[1]{%
1647 \renewcommand*\@gls@codepage{#1}%
1648 }

```

Suggested by egreg:

```

1649 \AtBeginDocument{%
1650 \ifx\gls@codepage\@empty
1651 \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
1652 \fi
1653 }
1654 \else
1655 \newcommand*\GlsSetXdyCodePage[1]{%
1656 \glsnoxywarning\GlsSetXdyCodePage}
1657 \fi

```

`xdylettergroups` Store letter group definitions.

```

1658 \ifglsxindy
1659 \ifglsexindy@glslnumbers
1660 \def\@xdylettergroups{(define-letter-group
1661 \string\glslnumbers\string^^J\space\space\space
1662 :prefixes (\string"0\string" \string"1\string"
1663 \string"2\string" \string"3\string" \string"4\string"
1664 \string"5\string" \string"6\string" \string"7\string"
1665 \string"8\string" \string"9\string")^^J\space\space\space
1666 \@xdynumbergrouporder)}
1667 \else
1668 \def\@xdylettergroups{}
1669 \fi
1670 \fi

```

`sAddLetterGroup` 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.

```

1671 \newcommand*\GlsAddLetterGroup[2]{%
1672 \expandafter\toks@\expandafter{\@xdylettergroups}%
1673 \protected@edef\@xdylettergroups{\the\toks@^^J%
1674 (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
1675 }%

```

1.5 Loops and conditionals

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

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

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

```
1676 \newcommand*{\forall glossaries}[3][\@glo@types]{%
1677   \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
1678 }
```

`\forall acronyms`

```
1679 \newcommand*{\forall acronyms}[2]{%
1680   \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
1681 }
```

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

```
\forall glsentries[<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.

```
1682 \newcommand*{\forall glsentries}[3][\glsdefaulttype]{%
1683   \protected@edef\@glo@list{\csname glo@list@#1\endcsname}%
1684   \@for#2:=\@glo@list\do
1685     {%
1686       \ifdefempty{#2}{\fi}%
1687     }%
1688 }
```

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

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

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

```
1689 \newcommand*{\forall glsentries}[3][\@glo@types]{%
1690   \expandafter\forall glossaries\expandafter[#1]{\@this@glo@}%
1691   {%
1692     \forall glsentries[\@this@glo@]{#2}{#3}%
1693   }%
1694 }
```

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

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

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

```
1695 \newcommand{\ifglossaryexists}{%
1696   \ifstar\s@ifglossaryexists\ifglossaryexists
1697 }
```

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

```
1698 \newcommand{\@ifglossaryexists}[3]{%
1699   \ifcsundef{@glotype@#1@out}{#3}{#2}%
1700 }
```

`\ifglossaryexists` Starred form includes ignored glossaries.

```
1701 \newcommand{\s@ifglossaryexists}[3]{%
1702   \ifcsundef{@glolist@#1}{#3}{#2}%
1703 }
```

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

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

(Note, don't use `\detokenize` or it will cause commands like `\glsaddall` to fail.) Since re-defining `\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`

```
1704 \newcommand*{\glsdetoklabel}[1]{#1}
```

`\glsentryexists` 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.

```
1705 \newcommand{\ifglsentryexists}[3]{%
1706   \ifcsundef{@glo@\glsdetoklabel{#1}@name}{#3}{#2}%
1707 }
```

`\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>*.

```

1708 \newcommand*{\ifglsused}[3]{%
1709   \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
1710 }

```

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

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

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

```

1711 \newcommand{\glsdoifexists}[2]{%
1712   \ifglsentryexists{#1}{#2}{%
1713     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
1714       has not been defined}{You need to define a glossary entry before you
1715       can use it.}%
1716 }

```

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

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

```

1717 \newcommand{\glsdoifnoexists}[2]{%
1718   \ifglsentryexists{#1}{%
1719     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
1720       been defined}{#2}%
1721 }

```

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

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

```

1722 \newcommand{\glsdoifexistsorwarn}[2]{%
1723   \ifglsentryexists{#1}{#2}{%
1724     \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
1725       has not been defined}%
1726   }%
1727 }

```

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

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

```

1728 \newcommand{\glsdoifexistsordo}[3]{%
1729   \ifglsentryexists{#1}{#2}{%
1730     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
1731       has not been defined}{You need to define a glossary entry before you
1732       can use it.}%
1733     #3%

```

```

1734 }%
1735 }

```

arynoexistsordo

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

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

```

1736 \newcommand{\doifglossarynoexistsordo}[3]{%
1737   \ifglossaryexists*{#1}%
1738   {%
1739     \PackageError{glossaries}{Glossary type ‘#1’ already exists}{%
1740       #3%
1741     }%
1742   {#2}%
1743 }

```

fglshaschildren

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

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

```

1744 \newrobustcmd{\ifglshaschildren}[3]{%
1745   \glstoifexists{#1}%
1746   {%
1747     \def\do@glshaschildren{#3}%
1748     \protected@edef\@gls@thislabel{\glstoifexists{#1}}%
1749     \expandafter\for@glstentries\expandafter
1750     [\csname glo@\@gls@thislabel @type\endcsname]
1751     {\glo@label}%
1752     {%
1753       \letcs\glo@parent{glo@\glo@label @parent}%
1754       \ifdefequal\@gls@thislabel\glo@parent
1755       {%
1756         \def\do@glshaschildren{#2}%
1757         \@endfortrue
1758       }%
1759     }%
1760   }%
1761   \do@glshaschildren
1762 }%
1763 }

```

\ifglshasparent

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

```

1764 \newcommand{\ifglshasparent}[3]{%
1765   \glsdoifexists{#1}%
1766   {%
1767     \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
1768   }%
1769 }

```

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

```

1770 \newcommand*{\ifglshasdesc}[3]{%
1771   \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
1772   {#3}%
1773   {#2}%
1774 }

```

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

```

1775 \newcommand*{\ifglstdesscsuppressed}[3]{%
1776   \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
1777   {#2}%
1778   {#3}%
1779 }

```

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

```

1780 \newrobustcmd*{\ifglshassymbol}[3]{%
1781   \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
1782   \ifdefempty\@glo@symbol
1783   {#3}%
1784   {%
1785     \ifdefequal\@glo@symbol\@gls@default@value
1786     {#3}%
1787     {#2}%
1788   }%
1789 }

```

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

```

1790 \newrobustcmd*{\ifglshaslong}[3]{%
1791   \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
1792   \ifdefempty\@glo@long
1793   {#3}%
1794   {%
1795     \ifdefequal\@glo@long\@gls@default@value
1796     {#3}%
1797     {#2}%
1798   }%
1799 }

```

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

```

1800 \newrobustcmd*{\ifglshasshort}[3]{%

```

```

1801 \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
1802 \ifdefempty\@glo@short
1803   {#3}%
1804   {%
1805     \ifdefequal\@glo@short\@gls@default@value
1806       {#3}%
1807       {#2}%
1808   }%
1809 }

```

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

```

1810 \newrobustcmd*{\ifglshasfield}[4]{%
1811   \glsdoifexists{#2}%
1812   {%
1813     \letcs{\@glo@thisvalue}{glo@\glsdetoklabel{#2}@#1}%

```

First check supplied field label is defined.

```

1814   \ifdef\@glo@thisvalue
1815   {%

```

Is defined, so now check if empty.

```

1816     \ifdefempty\@glo@thisvalue
1817     {%

```

Is empty, so doesn't have field set.

```

1818       #4%
1819     }%
1820   {%

```

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

```

1821     \ifdefequal\@glo@thisvalue\@gls@default@value
1822     {%

```

Value is set to the default value.

```

1823       #4%
1824     }%
1825   {%

```

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

```

1826     \let\glscurrentfieldvalue\@glo@thisvalue
1827     #3%
1828   }%
1829 }%
1830 }%
1831 {%

```

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

```

1832   \@gls@fetchfield{\@gls@thisfield}{#1}%

```


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

```
1833 \ifdef\@gls@thisfield
1834 {%
```

Is defined, so now check if empty.

```
1835 \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@@gls@thisfield}%
1836 \ifdefempty\@glo@thisvalue
1837 {%
```

Is empty so field hasn't been set.

```
1838 #4%
1839 }%
1840 {%
```

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

```
1841 \ifdequal\@glo@thisvalue\@gls@default@value
1842 {%
```

Value is set to the default value.

```
1843 #4%
1844 }%
1845 {%
```

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

```
1846 \let\glscurrentfieldvalue\@glo@thisvalue
1847 #3%
1848 }%
1849 }%
1850 }%
1851 {%
```

Not defined.

```
1852 \GlossariesWarning{Unknown entry field '#1'}%
1853 #4%
1854 }%
1855 }%
1856 }%
1857 }
```

rentfieldvalue

```
1858 \newcommand*{\glscurrentfieldvalue}{{}
```

1.6 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

```
1859 \newcommand*{\@glo@types}{,}
```

`\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.

```
1860 \newcommand*\@gls@provide@newglossary{%
1861   \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
   Only need to do this once.
1862   \let\@gls@provide@newglossary\relax
1863 }
```

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

```
1864 \newcommand*\defglsentryfmt}[2][\glsdefaulttype]{%
1865   \csgdef{gls@#1@entryfmt}{#2}%
1866 }
```

`\gls@doentryfmt`

```
1867 \newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
```

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

```
1868 \newcommand*\@gls@forbidtexext}[1]{%
1869   \ifboolexpr{test {\ifdefstring{#1}{tex}}
1870             or test {\ifdefstring{#1}{TEX}}}
1871   {%
1872     \def#1{nottex}%
1873     \PackageError{glossaries}%
1874       {Forbidden '.tex' extension replaced with '.nottex'}%
1875       {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
1876         Don't use '.tex' as an extension for a temporary file.}%
1877   }%
1878   {%
1879   }%
1880 }
```

`\gls@gobbleopt` Discard optional argument.

```
1881 \newcommand*\gls@gobbleopt{}\new@ifnextchar[{\@gls@gobbleopt}{}
1882 \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`

```
1883 \newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
```

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

```
1884 \newcommand*{\s@newglossary}[2]{%
1885 \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
1886 }
```

`\ns@newglossary` Define the unstarred version.

```
1887 \newcommand*{\ns@newglossary}[5][glg]{%
1888 \doifglossarynoexistsordo{#2}%
1889 {%
```

Check if default has been set

```
1890 \ifundef\glsdefaultttype
1891 {%
1892 \gdef\glsdefaultttype{#2}%
1893 }{}}%
```

Add this to the list of glossary types:

```
1894 \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.

```
1895 \expandafter\gdef\csname glolist@#2\endcsname{,}%
```

Store the file extensions:

```
1896 \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
1897 \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
1898 \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
1899 \expandafter\@gls@forbidtexext\csname @glotype@#2@log\endcsname
1900 \expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
1901 \expandafter\@gls@forbidtexext\csname @glotype@#2@out\endcsname
```

Store the title:

```
1902 \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
1903 \@gls@provide@newglossary
1904 \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 re-defined by the user later if required (see `\defglsentry`). This may already have been defined if this has been specified as a list of acronyms.

```
1905 \ifcsundef{gls@#2@entryfmt}%
1906 {%
1907 \defglsentryfmt[#2]{\glsentryfmt}%
1908 }%
1909 {}%
```

Define sort counter if required:

```
1910 \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.)

```
1911 \ifnextchar[{\@gls@setcounter{#2}}{%
1912   {\@gls@setcounter{#2}[\glscounter]}}%
1913 }%
1914 {%
1915   \gls@gobbleopt
1916 }%
1917 }
```

`\altnewglossary`

```
1918 \newcommand*{\altnewglossary}[3]{%
1919   \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
1920 }
```

Only define new glossaries in the preamble:

```
1921 \@onlypreamble{\newglossary}
```

Only define new glossaries before `\makeglossaries`

```
1922 \@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 \LaTeX , `\@newglossary` simply ignores its arguments.

`\@newglossary`

```
1923 \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`

```
1924 \def\@gls@setcounter#1[#2]{%
1925   \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
1926   \ifglsxindy
1927     \GlsAddXdyCounters{#2}%
1928   \fi
1929 }
```

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

`@gls@getcounter`

```
1930 \newcommand*{\@gls@getcounter}[1]{%
1931   \csname @glotype@#1@counter\endcsname
1932 }
```

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

```
1933 \glsdefmain
```

Define the “acronym” glossaries if required.

```
1934 \@gls@do@acronymsdef
```

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

```
1935 \@gls@do@symbolsdef
```

```
1936 \@gls@do@numbersdef
```

```
1937 \@gls@do@indexdef
```

`\ignoredglossary` 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.

```
1938 \newcommand*\newignoredglossary}[1]{%
1939   \ifdefempty\@ignored@glossaries
1940   {%
1941     \protected@edef\@ignored@glossaries{#1}%
1942   }%
1943   {%
1944     \protected@eappto\@ignored@glossaries{,#1}%
1945   }%
1946   \csgdef{glolist@#1}{,}%
1947   \ifcsundef{gls@#1@entryfmt}%
1948   {%
1949     \defglsentryfmt[#1]{\glsentryfmt}%
1950   }%
1951   {}%
1952   \ifdefempty\@gls@nohyperlist
1953   {%
1954     \renewcommand*\@gls@nohyperlist{#1}%
1955   }%
1956   {%
1957     \protected@eappto\@gls@nohyperlist{,#1}%
1958   }%
1959 }
```

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

```
1960 \newcommand*\@ignored@glossaries{}
```

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

```
1961 \newcommand*\ifignoredglossary}[3]{%
1962   \protected@edef\@gls@igtype{#1}%
1963   \expandafter\DTLifinlist\expandafter
1964   {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
1965 }
```

1.7 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.

```
1966 \define@key{glossentry}{name}{%
1967 \def\@glo@name{#1}%
1968 }
```

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`.

```
1969 \define@key{glossentry}{description}{%
1970 \def\@glo@desc{#1}%
1971 }
```

descriptionplural

```
1972 \define@key{glossentry}{descriptionplural}{%
1973 \def\@glo@descplural{#1}%
1974 }
```

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`.

```
1975 \define@key{glossentry}{sort}{%
1976 \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.

```
1977 \define@key{glossentry}{text}{%
1978 \def\@glo@text{#1}%
1979 }
```

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.

```
1980 \define@key{glossentry}{plural}{%
1981 \def\@glo@plural{#1}%
1982 }
```

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.

```
1983 \define@key{glossentry}{first}{%
1984 \def\@glo@first{#1}%
1985 }
```

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.

```
1986 \define@key{glossentry}{firstplural}{%
1987 \def\@glo@firstplural{#1}%
1988 }
```

s@default@value

```
1989 \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).

```
1990 \define@key{glossentry}{symbol}{%
1991 \def\@glo@symbol{#1}%
1992 }
```

symbolplural

```
1993 \define@key{glossentry}{symbolplural}{%
1994 \def\@glo@symbolplural{#1}%
1995 }
```

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

```
1996 \define@key{glossentry}{type}{%
1997 \def\@glo@type{#1}}
```

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

```
1998 \define@key{glossentry}{counter}{%
1999 \ifcsundef{c@#1}%
2000 {%
2001 \PackageError{glossaries}%
2002 {There is no counter called ‘#1’}%
2003 {%
2004 The counter key should have the name of a valid counter
2005 as its value%
2006 }%
2007 }%
```

```

2008  {%
2009    \def\@glo@counter{#1}%
2010  }%
2011 }

```

see The see key specifies a list of cross-references

```

2012 \define@key{glossentry}{see}{%
2013   \gls@set@xr@key{see}{\@glo@see}{#1}%
2014 }

```

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

Assign a cross-reference key.

```

2015 \newcommand*{\gls@set@xr@key}[3]{%
2016   \renewcommand*{\gls@xr@key}{#1}%
2017   \gls@checkseeallowed
2018   \def#2{#3}%
2019   \@glo@seeautonumberlist
2020 }

```

`\gls@xr@key`

```

2021 \newcommand*{\gls@xr@key}{see}

```

`checkseeallowed`

```

2022 \newcommand*{\gls@checkseeallowed}{%
2023   \@gls@see@noindex
2024 }

```

`ed@preambleonly`

```

2025 \newcommand*{\gls@checkseeallowed@preambleonly}{%
2026   \GlossariesWarning
2027   {'\gls@xr@key' key doesn't have any effect when used in the document
2028     environment. Move the definition to the preamble
2029     after \string\makeglossaries\space
2030     or \string\makenoidxglossaries}%
2031 }

```

parent The parent key specifies the parent entry, if required.

```

2032 \define@key{glossentry}{parent}{%
2033 \def\@glo@parent{#1}}

```

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

```

2034 \define@choicekey{glossentry}{nonumberlist}%
2035   [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
2036 {%
2037   \ifcase\gls@nonumberlist@nr\relax
2038     \def\@glo@prefix{\glsnonextpages}%

```



```

2039 \gls@savenonumberlist{true}%
2040 \else
2041 \def\glo@prefix{\glsnextpages}%
2042 \gls@savenonumberlist{false}%
2043 \fi
2044 }

```

`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.

```

2045 \newcommand*{\@gls@savenonumberlist}[1]{%

```

`nitnonumberlist`

```

2046 \newcommand*{\@gls@initnonumberlist}{%

```

`nitnonumberlist`

```

2047 \newcommand*{\@gls@storenonumberlist}[1]{%

```

`savenonumberlist` Allow the `nonumberlist` value to be saved.

```

2048 \newcommand*{\@gls@enablesavenonumberlist}{%
2049 \renewcommand*{\@gls@initnonumberlist}{%
2050 \undef\glo@nonumberlist
2051 }%
2052 \renewcommand*{\@gls@savenonumberlist}[1]{%
2053 \def\glo@nonumberlist{##1}%
2054 }%
2055 \renewcommand*{\@gls@storenonumberlist}[1]{%
2056 \ifdef\glo@nonumberlist
2057 {%
2058 \cslet{glo@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
2059 }%
2060 }%
2061 }%
2062 \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
2063 }

```

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

`user1`

```

2064 \define@key{glossentry}{user1}{%
2065 \def\glo@useri{#1}%
2066 }

```

`user2`

```

2067 \define@key{glossentry}{user2}{%
2068 \def\glo@userii{#1}%
2069 }

```

user3

```
2070 \define@key{glossentry}{user3}{%
2071   \def\@glo@useriii{#1}%
2072 }
```

user4

```
2073 \define@key{glossentry}{user4}{%
2074   \def\@glo@useriv{#1}%
2075 }
```

user5

```
2076 \define@key{glossentry}{user5}{%
2077   \def\@glo@userv{#1}%
2078 }
```

user6

```
2079 \define@key{glossentry}{user6}{%
2080   \def\@glo@uservi{#1}%
2081 }
```

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.

```
2082 \define@key{glossentry}{short}{%
2083   \def\@glo@short{#1}%
2084 }
```

shortplural This key is provided for use by \newacronym.

```
2085 \define@key{glossentry}{shortplural}{%
2086   \def\@glo@shortpl{#1}%
2087 }
```

long This key is provided for use by \newacronym.

```
2088 \define@key{glossentry}{long}{%
2089   \def\@glo@long{#1}%
2090 }
```

longplural This key is provided for use by \newacronym.

```
2091 \define@key{glossentry}{longplural}{%
2092   \def\@glo@longpl{#1}%
2093 }
```

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

```
2094 \newcommand*{\@glsnname}{%
2095   \PackageError{glossaries}{name key required in
2096     \string\newglossaryentry\space for entry '@glo@label'}{You
2097     haven't specified the entry name}}
```

```

\@glsnodesc Define command to generate error if description key is missing.
2098 \newcommand*\@glsnodesc{%
2099   \PackageError{glossaries}
2100   {%
2101     description key required in \string\newglossaryentry\space
2102     for entry '\@glo@label'%
2103   }%
2104   {%
2105     You haven't specified the entry description%
2106   }%
2107 }%

lsdefaultplural Now obsolete. Don't use.
2108 \newcommand*\@glsdefaultplural{}{}

missingnumberlist Define a command to generate warning when numberlist not set.
2109 \newcommand*\@gls@missingnumberlist}[1]{%
2110   ??%
2111   \ifglssavenumberlist
2112     \GlossariesWarning{Missing number list for entry '#1'.
2113       Maybe makeglossaries + rerun required}%
2114   \else
2115     \PackageError{glossaries}%
2116     {Package option 'savenumberlist=true' required}%
2117     {%
2118       You must use the 'savenumberlist' package option
2119       to reference location lists.%
2120     }%
2121   \fi
2122 }

@glsdefaultsort Define command to set default sort.
2123 \newcommand*\@glsdefaultsort{}\@glo@name}

\gls@level Register to increment entry levels.
2124 \newcount\gls@level

@noexpand@field
2125 \newcommand{\@@gls@noexpand@field}[3]{%
2126   \expandafter\global\expandafter
2127   \let\csname glo@#1@#2\endcsname#3%
2128 }

noexpand@fields
2129 \newcommand{\@gls@noexpand@fields}[4]{%
2130   \ifcsdef{gls@assign@#3@field}
2131   {%
2132     \ifdefequal{#4}\@gls@default@value}%

```

```

2133     {%
2134         \edef\@gls@value{\expandonce{#1}}%
2135         \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
2136     }%
2137     {%
2138         \csuse{gls@assign@#3@field}{#2}{#4}%
2139     }%
2140 }%
2141 {%
2142     \ifdefequal{#4}{\@gls@default@value}%
2143     {%
2144         \edef\@gls@value{\expandonce{#1}}%
2145         \@gls@noexpand@field{#2}{#3}{\@gls@value}%
2146     }%
2147     {%
2148         \@gls@noexpand@field{#2}{#3}{#4}%
2149     }%
2150 }%
2151 }

```

ls@expand@field

```

2152 \newcommand{\@gls@expand@field}[3]{%
2153     \expandafter
2154     \protected@xdef\csname glo@#1@#2\endcsname{#3}%
2155 }

```

s@expand@fields

```

2156 \newcommand{\@gls@expand@fields}[4]{%
2157     \ifcsdef{gls@assign@#3@field}
2158     {%
2159         \ifdefequal{#4}{\@gls@default@value}%
2160         {%
2161             \edef\@gls@value{\expandonce{#1}}%
2162             \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
2163         }%
2164         {%
2165             \expandafter\@gls@startswitexpandonce#4\relax\relax\gls@endcheck
2166             {%
2167                 \@gls@expand@field{#2}{#3}{#4}%
2168             }%
2169             {%
2170                 \csuse{gls@assign@#3@field}{#2}{#4}%
2171             }%
2172         }%
2173     }%
2174     {%
2175         \ifdefequal{#4}{\@gls@default@value}%
2176         {%

```

```

2177      \@gls@expand@field{#2}{#3}{#1}%
2178    }%
2179    {%
2180      \@gls@expand@field{#2}{#3}{#4}%
2181    }%
2182  }%
2183 }

```

switexpandonce

```

2184 \def\@gls@expandonce{\expandonce}
2185 \def\@gls@startswitexpandonce#1#2\gls@endcheck#3#4{%
2186   \def\@gls@tmp{#1}%
2187   \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
2188 }

```

ls@assign@field

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

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.

```
2189 \let\gls@assign@field\@gls@expand@fields
```

glsexpandfields

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

```

2190 \newcommand*{\glsexpandfields}{%
2191   \let\gls@assign@field\@gls@expand@fields
2192 }

```

snoexpandfields

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

```

2193 \newcommand*{\glsnoexpandfields}{%
2194   \let\gls@assign@field\@gls@noexpand@fields
2195 }

```

ewglossaryentry

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

```
2196 \newrobustcmd{\newglossaryentry}[2]{%
```

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

```

2197   \glsdoifnoexists{#1}%
2198   {%
2199     \gls@defglossaryentry{#1}{#2}%
2200   }%
2201 }

```

ewglossaryentry

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.

```

2202 \newcommand*{\gls@defdocnewglossaryentry}{%
2203   \let\newglossaryentry\new@glossaryentry
2204 }

```

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

```

2205 \newrobustcmd{\provideglossaryentry}[2]{%
2206   \ifglentryexists{#1}%
2207   }{%
2208   {%
2209     \gls@defglossaryentry{#1}{#2}%
2210   }%
2211 }
2212 \@onlypreamble{\provideglossaryentry}

```

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

```

2213 \newcommand*{\gls@glsdefs@labellist}{}

```

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

```

2214 \newcommand*{\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}

```

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

```

2215 \newcommand*{\gls@docdef@labellist}{}

```

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

```

2216 \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.

```

2217 \newrobustcmd{\new@glossaryentry}[2]{%
2218   \ifundef\@gls@deffile
2219   {%
2220     \global\newwrite\@gls@deffile
2221     \immediate\openout\@gls@deffile=\jobname.glsdefs
2222   }%
2223   {%

```

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.

```

2224   \ifglentryexists{#1}%
2225   {%
2226     \xifinlist{#1}{\gls@docdef@labellist}%
2227     {%

```

This entry has already been defined in the document environment.

```

2228     \GlossariesWarning{Entry ‘#1’ has already been defined in the
2229       document environment (ignoring new definition)}%
2230   }%

```

```

2231   {%
2232     \xifinlist{#1}{\gls@glsdefs@labellist}%
2233   {%

```

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

```

2234     \gls@defglossaryentry{#1}{#2}%
2235     \gls@docdef@addlabel{#1}%
2236     \@gls@writedef{#1}%
2237   }%
2238   {%

```

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

```

2239     \GlossariesWarning{Entry ‘#1’ has already been defined in
2240       the preamble (in which case remove the duplicate definition)
2241       or you have just updated to glossaries.sty
2242       v4.47 (in which case delete ‘\jobname.glsdefs’)}%
2243   }%
2244 }%
2245 }%
2246 {%

```

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

```

2247     \gls@defglossaryentry{#1}{#2}%
2248     \gls@docdef@addlabel{#1}%
2249     \@gls@writedef{#1}%
2250   }%
2251 }

```

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.

```

2252 \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.

```

2253 \AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}

```

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

```

2254 \newcommand*{\gls@begindocdefs}{%
2255   \@gls@enablesavenonumberlist
2256   \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@relax}%
2257   \makeatletter
2258   \InputIfFileExists{\jobname.glsdefs}{\gls@restoreat}{\gls@restoreat}%
2259   \@gls@restoreat
2260   \undef\@gls@restoreat
2261   \gls@defdocnewglossaryentry
2262 }

```

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

```
2263 \newcommand*{\glsdefs@newdocentry}[2]{%
2264   \ifglstryexists{#1}%
2265   {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
2266     (if you have moved the definition there from the document
2267     environment, delete \jobname.glsdefs)}}%
2268   {%
2269     \gls@glsdefs@addlabel{#1}%
2270     \gls@defglossaryentry{#1}{#2}%
2271   }%
2272 }
```

`\@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.

```
2273 \newcommand*{\@gls@writedef}[1]{%
2274   \immediate\write\@gls@deffile
2275   {%
2276     \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
2277     \expandafter\@gobble\string{\glspercentchar%
2278   }%
```

Write key value information:

```
2279   \@for\@gls@map:=\@gls@keymap\do
2280   {%
2281     \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
2282     \ifdef\glo@value
2283     {%
2284       \@onelevel@sanitize\glo@value
2285       \immediate\write\@gls@deffile
2286       {%
2287         \expandafter\@firstoftwo\@gls@map
2288         =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%
2289         \glspercentchar
2290       }%
2291     }%
2292   }%
2293 }
```

Provide hook:

```
2294   \glswritedefhook
2295   \immediate\write\@gls@deffile
2296   {%
2297     \glspercentchar^^J%
2298     \expandafter\@gobble\string\}\glspercentchar^^J%
2299   }%
2300 }
```

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


```

2301 \newcommand*{\@gls@keymap}{%
2302   {name}{name},%
2303   {sort}{sortvalue},% unescaped sort value
2304   {type}{type},%
2305   {first}{first},%
2306   {firstplural}{firstpl},%
2307   {text}{text},%
2308   {plural}{plural},%
2309   {description}{desc},%
2310   {descriptionplural}{descplural},%
2311   {symbol}{symbol},%
2312   {symbolplural}{symbolplural},%
2313   {user1}{useri},%
2314   {user2}{userii},%
2315   {user3}{useriii},%
2316   {user4}{useriv},%
2317   {user5}{userv},%
2318   {user6}{uservi},%
2319   {long}{long},%
2320   {longplural}{longpl},%
2321   {short}{short},%
2322   {shortplural}{shortpl},%
2323   {counter}{counter},%
2324   {parent}{parent}%
2325 }

```

@gls@fetchfield

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

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

```
2326 \newcommand*{\@gls@fetchfield}[2]{%
```

Ensure user field name is fully expanded

```
2327   \edef\@gls@thisval{#2}%
```

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

```

2328   \@for\@gls@map:=\@gls@keymap\do{%
2329     \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
2330     \ifdefequal{\@this@key}{\@gls@thisval}%
2331     {%

```

Found it.

```
2332       \edef#1{\expandafter\@secondoftwo\@gls@map}%
```

Break out of loop.

```

2333       \@endfortrue
2334     }%
2335   }%
2336 }%
2337 }

```

lsaddstoragekey

```
\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.

```
2338 \newcommand*{\glsaddstoragekey}{\ifstar\@sglsaddstoragekey\@glsaddstoragekey}
```

Starred version switches on expansion for this key.

```
2339 \newcommand*{\@sglsaddstoragekey}[1]{%
2340   \key@ifundefined{glossentry}{#1}%
2341   {%
2342     \expandafter\newcommand\expandafter*\expandafter
2343     {\csname gls@assign@#1@field\endcsname}[2]{%
2344       \@gls@expand@field{##1}{#1}{##2}%
2345     }%
2346   }%
2347   {}%
2348   \@glsaddstoragekey{#1}%
2349 }
```

Unstarred version doesn't override default expansion.

```
2350 \newcommand*{\@glsaddstoragekey}[3]{%
```

Check the specified key doesn't already exist.

```
2351   \key@ifundefined{glossentry}{#1}%
2352   {%
```

Set up the key.

```
2353     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
2354     \appto\@gls@keymap{,{#1}{#1}}%
```

Set the default value.

```
2355     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
```

Assignment code.

```
2356     \appto\@newglossaryentryposthook{%
2357       \letcs{\@glo@tmp}{@glo@#1}%
2358       \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
2359     }%
```

Define the no-link commands.

```
2360     \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
2361   }%
2362   {%
2363     \PackageError{glossaries}{Key ‘#1’ already exists}{}%
2364   }%
2365 }
```

\glsaddkey

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

Allow user to add their own custom keys.

```
2366 \newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
```

Starred version switches on expansion for this key.

```
2367 \newcommand*{\@sglsaddkey}[1]{%
2368   \key@ifundefined{glossentry}{#1}%
2369   {%
2370     \expandafter\newcommand\expandafter*\expandafter
2371       {\csname gls@assign@#1@field\endcsname}[2]{%
2372         \@gls@expand@field{##1}{#1}{##2}%
2373       }%
2374   }%
2375   {}%
2376   \@glsaddkey{#1}%
2377 }
```

Unstarred version doesn't override default expansion.

```
2378 \newcommand*{\@glsaddkey}[7]{%
```

Check the specified key doesn't already exist.

```
2379   \key@ifundefined{glossentry}{#1}%
2380   {%
```

Set up the key.

```
2381     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
2382     \appto\@gls@keymap{, {#1}{#1}}%
```

Set the default value.

```
2383     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
```

Assignment code.

```
2384     \appto\@newglossaryentryposthook{%
2385       \letcs{\@glo@tmp}{@glo@#1}%
2386       \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
2387     }%
```

Define the no-link commands.

```
2388     \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
2389     \newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
```

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

```
2390     \ifcsdef{@gls@user@#1@}%
2391     {%
2392       \PackageError{glossaries}%
2393       {Can't define '\string#5' as helper command
2394       '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
2395       }%
2396     }%
2397     {%
2398       \expandafter\newcommand\expandafter*\expandafter
2399       {\csname @gls@user@#1\endcsname}[2] []{%
2400         \new@ifnextchar[%
2401           {\csuse{@gls@user@#1@}{##1}{##2}}%
```

```

2402         {\csuse{@gls@user@#1@}{##1}{##2} []}}%
2403     \csdef{@gls@user@#1@}##1##2[##3]{%
2404         \@gls@field@link{##1}{##2}{#3{##2}##3}%
2405     }%
2406     \newrobustcmd*{#5}{%
2407         \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
2408 }%

```

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

```

2409     \ifcsdef{@Gls@user@#1@}%
2410     {%
2411         \PackageError{glossaries}%
2412         {Can't define '\string#6' as helper command
2413         '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
2414     }%
2415 }%
2416 {%
2417     \expandafter\newcommand\expandafter*\expandafter
2418     {\csname @Gls@user@#1\endcsname}[2] []{%
2419         \new@ifnextchar[%
2420             {\csuse{@Gls@user@#1@}{##1}{##2}}%
2421             {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
2422     \csdef{@Gls@user@#1@}##1##2[##3]{%
2423         \@gls@field@link{##1}{##2}{#4{##2}##3}%
2424     }%
2425     \newrobustcmd*{#6}{%
2426         \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
2427 }%

```

Finally the all caps version:

```

2428     \ifcsdef{@GLS@user@#1@}%
2429     {%
2430         \PackageError{glossaries}%
2431         {Can't define '\string#7' as helper command
2432         '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
2433     }%
2434 }%
2435 {%
2436     \expandafter\newcommand\expandafter*\expandafter
2437     {\csname @GLS@user@#1\endcsname}[2] []{%
2438         \new@ifnextchar[%
2439             {\csuse{@GLS@user@#1@}{##1}{##2}}%
2440             {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
2441     \csdef{@GLS@user@#1@}##1##2[##3]{%
2442         \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
2443     }%
2444     \newrobustcmd*{#7}{%
2445         \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%

```

```

2446 }%
2447 }%
2448 {%
2449 \PackageError{glossaries}{Key ‘#1’ already exists}{}%
2450 }%
2451 }

```

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

```

2452 \newcommand{\glsfieldxdef}[3]{%
2453 \glsdoifexists{#1}%
2454 {%
2455 \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2456 \ifcsdef{glo@\@glo@label @#2}%
2457 {%
2458 \protected@csxdef{glo@\@glo@label @#2}{#3}%
2459 }%
2460 {%
2461 \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2462 }%
2463 }%
2464 }

```

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

```

2465 \newcommand{\glsfielddedef}[3]{%
2466 \glsdoifexists{#1}%
2467 {%
2468 \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2469 \ifcsdef{glo@\@glo@label @#2}%
2470 {%
2471 \protected@csedef{glo@\@glo@label @#2}{#3}%
2472 }%
2473 {%
2474 \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2475 }%
2476 }%
2477 }

```

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

```

2478 \newcommand{\glsfieldgdef}[3]{%
2479 \glsdoifexists{#1}%

```

```

2480 {%
2481   \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2482   \ifcsdef{glo@\@glo@label @#2}%
2483   {%
2484     \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
2485   }%
2486   {%
2487     \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2488   }%
2489 }%
2490 }

```

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

```

2491 \newcommand{\glsfielddef}[3]{%
2492   \glsdoifexists{#1}%
2493   {%
2494     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2495     \ifcsdef{glo@\@glo@label @#2}%
2496     {%
2497       \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
2498     }%
2499     {%
2500       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2501     }%
2502   }%
2503 }

```

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

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

```

2504 \newcommand{\glsfieldfetch}[3]{%
2505   \glsdoifexists{#1}%
2506   {%
2507     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2508     \ifcsdef{glo@\@glo@label @#2}%
2509     {%
2510       \letcs#3{glo@\@glo@label @#2}%
2511     }%
2512     {%
2513       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2514     }%
2515   }%
2516 }

```

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

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

```

2517 \newcommand{\ifglsfieldeq}[5]{%
2518   \glsdoifexists{#1}%
2519   {%
2520     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2521     \ifcsdef{glo@\@glo@label @#2}%
2522     {%
2523       \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
2524     }%
2525     {%
2526       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2527     }%
2528   }%
2529 }
```

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

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

```

2530 \newcommand{\ifglsfielddedef}[5]{%
2531   \glsdoifexists{#1}%
2532   {%
2533     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2534     \ifcsdef{glo@\@glo@label @#2}%
2535     {%
2536       \expandafter\ifdefstrequal
2537       \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
2538     }%
2539     {%
2540       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2541     }%
2542   }%
2543 }
```

`\ifglsfielddcseq` `\ifglsfielddcseq{<label>}{<field>}{<cs name>}{<true>}{<false>}`

As above but uses `\ifcsstrequal` instead of `\ifdefstrequal`

```

2544 \newcommand{\ifglsfielddcseq}[5]{%
2545   \glsdoifexists{#1}%
2546   {%
2547     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2548     \ifcsdef{glo@\@glo@label @#2}%
2549     {%
2550       \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
2551     }%
2552   }%
2553 }
```

```

2552   {%
2553       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2554   }%
2555 }%
2556 }

```

gls.writedefhook

```

2557 \newcommand*{\gls.writedefhook}{}

```

gls.assign@desc

```

2558 \newcommand*{\gls.assign@desc}[1]{%
2559   \gls.assign@field{#1}{desc}{\@glo@desc}%
2560   \gls.assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
2561 }

```

newglossaryentry

```

2562 \newcommand{\longnewglossaryentry}[3]{%
2563   \glsdoifnoexists{#1}%
2564   {%
2565     \bgroup
2566     \let\@org@newglossaryentryprehook\@newglossaryentryprehook
2567     \long\def\@newglossaryentryprehook{%
2568       \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
2569       \@org@newglossaryentryprehook
2570     }%
2571     \renewcommand*{\gls.assign@desc}[1]{%
2572       \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
2573       \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
2574     }
2575     \gls@defglossaryentry{#1}{#2}%
2576   \egroup
2577 }
2578 }

```

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

```

2579 \@onlypreamble{\longnewglossaryentry}

```

deglossaryentry As the above but only defines the entry if it doesn’t already exist.

```

2580 \newcommand{\longprovideglossaryentry}[3]{%
2581   \ifglstryexists{#1}{%
2582     {\longnewglossaryentry{#1}{#2}{#3}}%
2583   }
2584 \@onlypreamble{\longprovideglossaryentry}

```

defglossaryentry

```

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

```

Defines a new entry without checking if it already exists.

```

2585 \newcommand{\gls@defglossaryentry}[2]{%

```


Prevent any further use of \GlsSetQuote:

```
2586 \let\GlsSetQuote\gls@nosetquote
```

Store label

```
2587 \protected@edef\@glo@label{\glsdetoklabel{#1}}%
```

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

```
2588 \let\glslabel\@glo@label
```

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

```
2589 \let\@glo@name\@gls@noname
```

```
2590 \let\@glo@desc\@gls@nodesc
```

```
2591 \let\@glo@descplural\@gls@default@value
```

```
2592 \let\@glo@type\@gls@default@value
```

```
2593 \let\@glo@symbol\@gls@default@value
```

```
2594 \let\@glo@symbolplural\@gls@default@value
```

```
2595 \let\@glo@text\@gls@default@value
```

```
2596 \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.)

```
2597 \let\@glo@first\@gls@default@value
```

```
2598 \let\@glo@firstplural\@gls@default@value
```

Set the default sort:

```
2599 \let\@glo@sort\@gls@default@value
```

Set the default counter:

```
2600 \let\@glo@counter\@gls@default@value
```

```
2601 \def\@glo@see{}%
```

```
2602 \def\@glo@parent{}%
```

```
2603 \def\@glo@prefix{}%
```

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

```
2604 \@gls@initnonumberlist
```

```
2605 \def\@glo@useri{}%
```

```
2606 \def\@glo@userii{}%
```

```
2607 \def\@glo@useriii{}%
```

```
2608 \def\@glo@useriv{}%
```

```
2609 \def\@glo@userv{}%
```

```
2610 \def\@glo@uservi{}%
```

```

2611 \def\@glo@short{}%
2612 \def\@glo@shortpl{}%
2613 \def\@glo@long{}%
2614 \def\@glo@longpl{}%

```

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

```
2615 \@newglossaryentryprehook
```

Extract key-val information from third parameter:

```
2616 \setkeys{glossentry}{#2}%
```

Check there is a default glossary.

```

2617 \ifundef\glsdefaulttype
2618 {%
2619   \PackageError{glossaries}%
2620     {No default glossary type (have you used ‘nomain’ by mistake?)}%
2621     {If you use package option ‘nomain’ you must define
2622      a new glossary before you can define entries}%
2623 }%
2624 {}%

```

Assign type. This must be fully expandable

```

2625 \gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
2626 \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.

```

2627 \ifcsundef{glolist@\@glo@type}%
2628 {%
2629   \PackageError{glossaries}%
2630     {Glossary type ‘\@glo@type’ has not been defined}%
2631     {You need to define a new glossary type, before making entries
2632      in it}%
2633 }%
2634 {}%

```

Check if it's an ignored glossary

```

2635 \ifignoredglossary\@glo@type
2636 {%

```

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

```

2637   \ifx\@glo@desc\@glsnodelsc
2638     \let\@glo@desc\@empty
2639   \fi
2640 }%
2641 {%
2642 }%
2643 \protected@edef\@glolist@{\csname glolist@\@glo@type\endcsname}%
2644 \expandafter\protected@xdef\csname glolist@\@glo@type\endcsname{%
2645   \@glolist@{\@glo@label},}%
2646 }%

```

Initialise level to 0.

```

2647 \gls@level=0\relax

```

Has this entry been assigned a parent?

```

2648 \ifx\@glo@parent\@empty

```

Doesn't have a parent. Set `\glo@<label>@parent` to empty.

```

2649 \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
2650 \else

```

Has a parent. Check to ensure this entry isn't its own parent.

```

2651 \ifdefequal\@glo@label\@glo@parent%
2652 {%
2653 \PackageError{glossaries}{Entry '@glo@label' can't be its own parent}{}%
2654 \def\@glo@parent{}%
2655 \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
2656 }%
2657 {%

```

Check the parent exists:

```

2658 \ifglentryexists{\@glo@parent}%
2659 {%

```

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

```

2660 \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
2661 \@glo@parent}%

```

Determine level.

```

2662 \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
2663 \advance\gls@level by 1\relax

```

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

```

2664 \ifx\@glo@name\@gls@name
2665 \expandafter\let\expandafter\@glo@name
2666 \csname glo@\@glo@parent @name\endcsname

```

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

```

2667 \ifx\@glo@plural\@gls@default@value
2668 \expandafter\let\expandafter\@glo@plural
2669 \csname glo@\@glo@parent @plural\endcsname
2670 \fi
2671 \fi
2672 }%
2673 {%

```

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

```

2674 \PackageError{glossaries}%
2675 {%
2676 Invalid parent '@glo@parent'
2677 for entry '@glo@label' - parent doesn't exist%
2678 }%
2679 {%
2680 Parent entries must be defined before their children%

```

```

2681     }%
2682     \def\@glo@parent{}%
2683     \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
2684 }%
2685 }%
2686 \fi

Set the level for this entry
2687 \expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%

Define commands associated with this entry:
2688 \gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
2689 \letcs\@glo@sort{glo@\@glo@label @sortvalue}%
2690 \gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
2691 \expandafter\gls@assign@field\expandafter
2692     {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
2693     {\@glo@label}{plural}{\@glo@plural}%
2694 \expandafter\gls@assign@field\expandafter
2695     {\csname glo@\@glo@label @text\endcsname}%
2696     {\@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.
2697 \ifx\@glo@first\@gls@default@value
2698     \expandafter\gls@assign@field\expandafter
2699         {\csname glo@\@glo@label @plural\endcsname}%
2700         {\@glo@label}{firstpl}{\@glo@firstplural}%
2701 \else
2702     \expandafter\gls@assign@field\expandafter
2703         {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
2704         {\@glo@label}{firstpl}{\@glo@firstplural}%
2705 \fi

2706 \ifcsundef{@glotype@\@glo@type @counter}%
2707 {%
2708     \def\@glo@defaultcounter{\glscounter}%
2709 }%
2710 {%
2711     \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
2712 }%
2713 \gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
2714 \gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
2715 \gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
2716 \gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
2717 \gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
2718 \gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
2719 \gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
2720 \gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
2721 \gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
2722 \gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
2723 \gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%

```

```

2724 \ifx\@glo@name\@glsnoname
2725   \@glsnoname
2726   \let\@glo@name\@gls@default@value
2727 \fi
2728 \gls@assign@field{\@glo@label}{name}{\@glo@name}%

```

Set default numberlist if not defined:

```

2729 \ifcsundef{glo@\@glo@label @numberlist}%
2730 {%
2731   \csxdef{glo@\@glo@label @numberlist}{%
2732     \noexpand\@gls@missingnumberlist{\@glo@label}}%
2733 }%
2734 {}%

```

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

```

2735 \@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.

```

2736 \def\@glo@@desc{\@glo@first}%
2737 \ifx\@glo@desc\@glo@@desc
2738   \let\@glo@desc\@glo@first
2739 \fi
2740 \ifx\@glo@desc\@gls@nodesc
2741   \@gls@nodesc
2742   \let\@glo@desc\@gls@default@value
2743 \fi
2744 \gls@assign@desc{\@glo@label}%

```

Set the sort key for this entry:

```

2745 \@gls@defsort{\@glo@type}{\@glo@label}%

2746 \def\@glo@@symbol{\@glo@text}%
2747 \ifx\@glo@symbol\@glo@@symbol
2748   \let\@glo@symbol\@glo@text
2749 \fi
2750 \gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
2751 \expandafter
2752   \gls@assign@field\expandafter
2753   {\csname glo@\@glo@label @symbol\endcsname}
2754   {\@glo@label}{symbolplural}{\@glo@symbolplural}%

```

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

```

2755 \expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
2756   \noexpand\global
2757   \noexpand\let\expandafter\noexpand
2758   \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
2759 }%
2760 \expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
2761   \noexpand\global

```

```

2762      \noexpand\let\expandafter\noexpand
2763      \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
2764 }%
2765 \csname glo@\@glo@label @flagfalse\endcsname

```

Sort out any cross-referencing if required.

```

2766 \@glo@autosee

```

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

```

2767 \ifignoredglossary\@glo@type
2768 {%
2769   \csdef{glo@\@glo@label @index}{}%
2770 }
2771 {%
2772   \do@glo@storeentry{\@glo@label}%
2773 }%

```

Define entry counters if enabled:

```

2774 \@newglossaryentry@defcounters

```

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

```

2775 \@newglossaryentryposthook
2776 }

```

\@glo@autosee Automatically implement \glssee.

```

2777 \newcommand*{\@glo@autosee}{%
2778   \ifdefvoid\@glo@see{}%
2779   {%
2780     \protected@edef\@do@glssee{%
2781       \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
2782       \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
2783     \do@glssee
2784   }%
2785   \@glo@autoseehook
2786 }%

```

glo@autoseehook

```

2787 \newcommand*{\@glo@autoseehook}{}

```

aryentryprehook Allow extra information to be added to glossary entries:

```

2788 \newcommand*{\@newglossaryentryprehook}{}

```

ryentryposthook Allow extra information to be added to glossary entries:

```

2789 \newcommand*{\@newglossaryentryposthook}{}

```

try@defcounters

```

2790 \newcommand*{\@newglossaryentry@defcounters}{}

```

`\glsmoveentry` Moves entry whose label is given by first argument to the glossary named in the second argument.

```

2791 \newcommand*{\glsmoveentry}[2]{%
2792   \protected@edef\glo@thislabel{\glsdetoklabel{#1}}%
2793   \protected@edef\glo@type{\csname glo@\glo@thislabel @type\endcsname}%
2794   \def\glo@list{,}%
2795   \for\glsentries[\glo@type]{\glo@label}%
2796   {%
2797     \ifdefequal\@glo@thislabel\glo@label
2798     {\protected@eappto\glo@list{\glo@label,}}%
2799   }%
2800   \cslet\glo@list@\glo@type{\glo@list}%
2801   \csdef\glo@\@glo@thislabel @type{#2}%
2802 }

```

`\glossaryentryfield` Indicate what command should be used to display each entry in the glossary. (This enables the glossaries-accsupp package to use `\accsuppglossaryentryfield` instead.)

```

2803 \ifglxindy
2804   \newcommand*{\@glossaryentryfield}{\string\glossentry}
2805 \else
2806   \newcommand*{\@glossaryentryfield}{\string\glossentry}
2807 \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.)

```

2808 \ifglxindy
2809   \newcommand*{\@glossarysubentryfield}{%
2810     \string\subglossentry}
2811 \else
2812   \newcommand*{\@glossarysubentryfield}{%
2813     \string\subglossentry}
2814 \fi

```

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

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.)

```

2815 \newcommand{\@glo@storeentry}[1]{%

```

Escape makeindex/xindy special characters in the label:

```

2816   \protected@edef\@glo@esclabel{#1}%
2817   \@gls@checkmkidxchars\@glo@esclabel

```

Get the sort string and escape any special characters

```

2818   \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
2819   \@gls@checkmkidxchars\@glo@sort

```

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

```

2820 \@gls@checkmkidxchars\@glo@prefix

Get the parent, if one exists
2821 \protected@edef\@glo@parent{\csname glo@#1@parent\endcsname}%

Write the information to the glossary file.
2822 \ifglxindy

Store using xindy syntax.
2823 \ifx\@glo@parent\@empty

Entry doesn't have a parent
2824 \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
2825 (\string"\@glo@sort\string" %
2826 \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
2827 }%
2828 \else

Entry has a parent
2829 \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
2830 \csname glo@\@glo@parent @index\endcsname
2831 (\string"\@glo@sort\string" %
2832 \string"\@glo@prefix\@glossarysubentryfield
2833 {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
2834 }%
2835 \fi
2836 \else

Store using makeindex syntax.
2837 \ifx\@glo@parent\@empty

Sanitize \@glo@prefix
2838 \@onelevel@sanitize\@glo@prefix

Entry doesn't have a parent
2839 \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
2840 \@glo@sort\@gls@actualchar\@glo@prefix
2841 \@glossaryentryfield{\@glo@esclabel}%
2842 }%
2843 \else

Entry has a parent
2844 \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
2845 \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
2846 \@glo@sort\@gls@actualchar\@glo@prefix
2847 \@glossarysubentryfield
2848 {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
2849 }%
2850 \fi
2851 \fi
2852 }

```


1.8 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.

`@ifnotmeasuring`

```
2853 \AtBeginDocument{%
2854   \ifpackageloaded{amsmath}%
2855   {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
2856   }%
2857 }
2858 \newcommand*{\@gls@ifnotmeasuring}[1]{%
2859   \ifmeasuring@
2860   \else
2861     #1%
2862   \fi
2863 }
2864 \newcommand*\gls@ifnotmeasuring[1]{#1}
```

`lspatchtabularx` 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.

```
2865 \def\@gls@patchtabularx#1\hbox#2#3!!{%
2866   \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
2867 }
2868 \newcommand*\glspatchtabularx{%
2869   \ifdef\TX@trial
2870   {%
2871     \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
2872     \let\glspatchtabularx\relax
2873   }%
2874   {%
2875   }
```

`\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.

```
2876 \newcommand*{\glsreset}[1]{%
2877   \gls@ifnotmeasuring
2878   {%
2879     \glsdoifexists{#1}%
2880     {%
2881       \@glsreset{#1}%
2882     }%
2883   }%
2884 }
```

`\glslocalreset` As above, but with only a local effect:

```

2885 \newcommand*{\glslocalreset}[1]{%
2886   \gls@ifnotmeasuring
2887   {%
2888     \glsdoifexists{#1}%
2889     {%
2890       \@glslocalreset{#1}%
2891     }%
2892   }%
2893 }

```

`\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.

```

2894 \newcommand*{\glsunset}[1]{%
2895   \gls@ifnotmeasuring
2896   {%
2897     \glsdoifexists{#1}%
2898     {%
2899       \@glsunset{#1}%
2900     }%
2901   }%
2902 }

```

`\glslocalunset` As above, but with only a local effect:

```

2903 \newcommand*{\glslocalunset}[1]{%
2904   \gls@ifnotmeasuring
2905   {%
2906     \glsdoifexists{#1}%
2907     {%
2908       \@glslocalunset{#1}%
2909     }%
2910   }%
2911 }

```

`\@glslocalunset` Local unset. This defaults to just `\@glslocalunset` but is changed by `\glsenableentrycount`.

```

2912 \newcommand*{\@glslocalunset}{\@glslocalunset}

```

`@@glslocalunset` Local unset without checks.

```

2913 \newcommand*{\@glslocalunset}[1]{%
2914   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
2915 }

```

`\@glsunset` Global unset. This defaults to just `\@glsunset` but is changed by `\glsenableentrycount`.

```

2916 \newcommand*{\@glsunset}{\@glsunset}

```

`\@@glsunset` Global unset without checks.

```

2917 \newcommand*{\@@glsunset}[1]{%
2918   \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
2919 }

```

`\@glslocalreset` Local reset. This defaults to just `\@@glslocalreset` but is changed by `\glsenableentrycount`.

```
2920 \newcommand*{\@glslocalreset}{\@@glslocalreset}
```

`\@@glslocalreset` Local reset without checks.

```
2921 \newcommand*{\@@glslocalreset}[1]{%
2922   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
2923 }
```

`\@glsreset` Global reset. This defaults to just `\@@glsreset` but is changed by `\glsenableentrycount`.

```
2924 \newcommand*{\@glsreset}{\@@glsreset}
```

`\@@glsreset` Global reset without checks.

```
2925 \newcommand*{\@@glsreset}[1]{%
2926   \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
2927 }
```

Reset all entries for the named glossaries (supplied in a comma-separated list). Syntax:
`\glsresetall[⟨glossary-list⟩]`

`\glsresetall`

```
2928 \newcommand*{\glsresetall}[1][\@glo@types]{%
2929   \forallglsentries[#1]{\@glsentry}%
2930   {%
2931     \glsreset{\@glsentry}%
2932   }%
2933 }
```

As above, but with only a local effect:

`\glslocalresetall`

```
2934 \newcommand*{\glslocalresetall}[1][\@glo@types]{%
2935   \forallglsentries[#1]{\@glsentry}%
2936   {%
2937     \glslocalreset{\@glsentry}%
2938   }%
2939 }
```

Unset all entries for the named glossaries (supplied in a comma-separated list). Syntax:
`\glsunsetall[⟨glossary-list⟩]`

`\glsunsetall`

```
2940 \newcommand*{\glsunsetall}[1][\@glo@types]{%
2941   \forallglsentries[#1]{\@glsentry}%
2942   {%
2943     \glsunset{\@glsentry}%
2944   }%
2945 }
```

As above, but with only a local effect:

lslocalunsetall

```
2946 \newcommand*{\glslocalunsetall}[1][\@glo@types]{%
2947   \forallglsentries[#1]{\@glsentry}%
2948   {%
2949     \glslocalunset{\@glsentry}%
2950   }%
2951 }
```

1.9 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 \TeX counter or even an explicit \TeX 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.

`try@defcounters` Define entry fields to keep track of how many times that entry has been marked as used.

```
2952 \newcommand*{\@newglossaryentry@defcounters}{%
2953   \csdef{glo@\@glo@label @currcount}{0}%
2954   \csdef{glo@\@glo@label @prevcount}{0}%
2955 }
```

`nableentrycount` Enables tracking of how many times an entry has been marked as used.

```
2956 \newcommand*{\glsenableentrycount}{%
```

Enable new entry fields.

```
2957   \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
```

Disable `\newglossaryentry` in the document environment.

```
2958   \renewcommand*{\gls@defdocnewglossaryentry}{%
2959     \renewcommand*{\newglossaryentry}[2]{%
2960       \PackageError{glossaries}{\string\newglossaryentry\space
2961         may only be used in the preamble when entry counting has
2962         been activated}{If you use \string\glsenableentrycount\space
2963         you must place all entry definitions in the preamble not in
2964         the document environment}%
2965     }%
2966   }%
```

Define commands `\glsentrycurrcount` and `\glsentryprevcount` to access these new fields. Default to zero if undefined.

```
2967   \newcommand*{\glsentrycurrcount}[1]{%
2968     \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
2969     {0}{\@gls@entry@field{##1}{currcount}}%
2970   }%
2971   \newcommand*{\glsentryprevcount}[1]{%
```

```

2972 \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
2973 {0}{\@gls@entry@field{##1}{prevcount}}}%
2974 }%

```

Make the unset and reset functions also increment or reset the entry counter.

```

2975 \renewcommand*{\@glsunset}[1]{%
2976   \@glsunset{##1}%
2977   \@gls@increment@currcount{##1}%
2978 }%
2979 \renewcommand*{\@glslocalunset}[1]{%
2980   \@glslocalunset{##1}%
2981   \@gls@local@increment@currcount{##1}%
2982 }%
2983 \renewcommand*{\@glsreset}[1]{%
2984   \@glsreset{##1}%
2985   \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
2986 }%
2987 \renewcommand*{\@glslocalreset}[1]{%
2988   \@glslocalreset{##1}%
2989   \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
2990 }%

```

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.)

```

2991 \def\@cgl's@##1##2[##3]{%
2992   \ifnum\glsentryprevcount{##2}=1\relax
2993     \cgl'sformat{##2}{##3}%
2994     \glsunset{##2}%
2995   \else
2996     \@gls@{##1}{##2}[##3]%
2997   \fi
2998 }%

```

Similarly for the analogous commands. No case change plural:

```

2999 \def\@cgl'spl@##1##2[##3]{%
3000   \ifnum\glsentryprevcount{##2}=1\relax
3001     \cgl'splformat{##2}{##3}%
3002     \glsunset{##2}%
3003   \else
3004     \@cgl'spl@{##1}{##2}[##3]%
3005   \fi
3006 }%

```

First letter uppercase singular:

```

3007 \def\@cGls@##1##2[##3]{%
3008   \ifnum\glsentryprevcount{##2}=1\relax
3009     \cGlsformat{##2}{##3}%
3010     \glsunset{##2}%
3011   \else
3012     \@Gls@{##1}{##2}[##3]%
3013   \fi

```

3014 }%

First letter uppercase plural:

```
3015 \def\@cGlsp1@##1##2[##3]{%
3016   \ifnum\glentryprevcount{##2}=1\relax
3017     \cGlsp1format{##2}{##3}%
3018     \glunset{##2}%
3019   \else
3020     \@Glsp1@{##1}{##2}[##3]%
3021   \fi
3022 }%
```

Write information to aux file at the end of the document

```
3023 \AtEndDocument{\@gls@write@entrycounts}%
```

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

```
3024 \renewcommand*{\@gls@entry@count}[2]{%
3025   \csxdef{glo@\glsetoklabel{##1}@prevcount}{##2}%
3026 }%
```

\glsenableentrycount may only be used once and only in the preamble.

```
3027 \let\glsenableentrycount\relax
3028 }
3029 \@onlypreamble\glsenableentrycount
```

ement@currcount

```
3030 \newcommand*{\@gls@increment@currcount}[1]{%
3031   \csxdef{glo@\glsetoklabel{##1}@currcount}{%
3032     \number\numexpr\glentrycurrcount{##1}+1}%
3033 }
```

ement@currcount

```
3034 \newcommand*{\@gls@local@increment@currcount}[1]{%
3035   \csxdef{glo@\glsetoklabel{##1}@currcount}{%
3036     \number\numexpr\glentrycurrcount{##1}+1}%
3037 }
```

ite@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.)

```
3038 \newcommand*{\@gls@write@entrycounts}{%
3039   \immediate\write\@auxout
3040     {\string\providecommand*\string\@gls@entry@count}[2]{}}%
3041   \forallglentries{\@glentry}{%
3042     \ifglused{\@glentry}%
3043     {\immediate\write\@auxout
3044       {\string\@gls@entry@count{\@glentry}{\glentrycurrcount{\@glentry}}}}%
3045     {}}%
```

```

3046 }%
3047 }

gls@entry@count Default behaviour is to ignore arguments. Activated by \glsenableentrycount.
3048 \newcommand*{\@gls@entry@count}[2]{%

\cgl's Define command that works like \gls but behaves differently if the entry count function is
enabled. (If not enabled, it behaves the same as \gls but issues a warning.)
3049 \newrobustcmd*{\cgl's}{\@gls@hyp@opt\@cgl's}

\@cgl's Defined the un-starred form. Need to determine if there is a final optional argument
3050 \newcommand*{\@cgl's}[2][{}]{%
3051 \new@ifnextchar[{\@cgl's@{#1}{#2}}{\@cgl's@{#1}{#2}[]}%
3052 }

\@cgl's@ Read in the final optional argument. This defaults to same behaviour as \gls but issues a
warning.
3053 \def\@cgl's@#1#2[#3]{%
3054 \GlossariesWarning{\string\cgl's\space is defaulting to
3055 \string\gls\space since you haven't enabled entry counting}%
3056 \@gls@{#1}{#2}[#3]%
3057 }

\cgl'sformat Format used by \cgl's if entry only used once on previous run. The first argument is the label,
the second argument is the insert text.
3058 \newcommand*{\cgl'sformat}[2]{%
3059 \ifgl'shaslong{#1}{\gl'sentrylong{#1}}{\gl'sentryfirst{#1}}#2%
3060 }

\cGl's Define command that works like \Gl's but behaves differently if the entry count function is
enabled. (If not enabled, it behaves the same as \Gl's but issues a warning.)
3061 \newrobustcmd*{\cGl's}{\@gls@hyp@opt\@cGl's}

\@cGl's Defined the un-starred form. Need to determine if there is a final optional argument
3062 \newcommand*{\@cGl's}[2][{}]{%
3063 \new@ifnextchar[{\@cGl's@{#1}{#2}}{\@cGl's@{#1}{#2}[]}%
3064 }

\@cGl's@ Read in the final optional argument. This defaults to same behaviour as \Gl's but issues a
warning.
3065 \def\@cGl's@#1#2[#3]{%
3066 \GlossariesWarning{\string\cGl's\space is defaulting to
3067 \string\Gl's\space since you haven't enabled entry counting}%
3068 \@Gl's@{#1}{#2}[#3]%
3069 }

```

`\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.

```
3070 \newcommand*{\cGlsformat}[2]{%
3071   \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
3072 }
```

`\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.)

```
3073 \newrobustcmd*{\cglsp1}{\@gls@hyp@opt\@cglsp1}
```

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

```
3074 \newcommand*{\@cglsp1}[2][{}]{%
3075   \new@ifnextchar[\@cglsp1@{#1}{#2}}{\@cglsp1@{#1}{#2}[{}]}%
3076 }
```

`\@cglsp1@` Read in the final optional argument. This defaults to same behaviour as `\glsp1` but issues a warning.

```
3077 \def\@cglsp1@#1#2[#3]{%
3078   \GlossariesWarning{\string\cglsp1\space is defaulting to
3079     \string\glsp1\space since you haven't enabled entry counting}%
3080   \@glsp1@{#1}{#2}[#3]%
3081 }
```

`\cglsp1format` Format used by `\cglsp1` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```
3082 \newcommand*{\cglsp1format}[2]{%
3083   \ifglshaslong{#1}{\glsp1long{#1}}{\glsp1firstplural{#1}}#2%
3084 }
```

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

```
3085 \newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
```

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

```
3086 \newcommand*{\@cGlspl}[2][{}]{%
3087   \new@ifnextchar[\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[{}]}%
3088 }
```

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

```
3089 \def\@cGlspl@#1#2[#3]{%
3090   \GlossariesWarning{\string\cGlspl\space is defaulting to
3091     \string\Glspl\space since you haven't enabled entry counting}%
3092   \@Glspl@{#1}{#2}[#3]%
3093 }
```


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

```
3094 \newcommand*{\cGlsplformat}[2]{%
3095   \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
3096 }
```

1.10 Loading files containing glossary entries

Glossary entries can be defined in an external file. These external files can contain `\newglossaryentry` and `\newacronym` commands.¹

`\loadglsentries[<type>]{<filename>}`

This command will input the file using `\input`. The optional argument specifies to which glossary the entries should be assigned if they haven't used the type key. If the optional argument is not specified, the default glossary is used. Only those entries used in the document (via `\glslink`, `\gls`, `\glspl` and uppercase variants or `\glsadd` and `\glsaddall` will appear in the glossary). The mandatory argument is the filename (with or without `.tex` extension).

`\loadglsentries`

```
3097 \newcommand*{\loadglsentries}[2][\@gls@default]{%
3098   \let\@gls@default\glsdefaulttype
3099   \def\glsdefaulttype{#1}\input{#2}%
3100   \let\glsdefaulttype\@gls@default
3101 }
```

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

```
3102 \@onlypreamble{\loadglsentries}
```

1.11 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`

```
3103 \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`. To

¹ and any other valid \TeX code that can be used in the preamble.

ensure backward compatibility, the default use the old `\glsdisplay` and `\glsdisplayfirst` style of commands

```
3104 \newcommand*{\glsentryfmt}{%
3105   \@gls@default@entryfmt\glsdisplayfirst\glsdisplay
3106 }
```

Format that provides backwards compatibility:

```
3107 \newcommand*{\@gls@default@entryfmt}[2]{%
3108   \ifdefempty\glscustomtext
3109   {%
3110     \glsifplural
3111     {%
```

Plural form

```
3112     \glscapscase
3113     {%
```

Don't adjust case

```
3114     \ifglsused\glslabel
3115     {%
```

Subsequent use

```
3116         #2{\glsentryplural{\glslabel}}%
3117         {\glsentrydescplural{\glslabel}}%
3118         {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
3119     }%
3120     {%
```

First use

```
3121         #1{\glsentryfirstplural{\glslabel}}%
3122         {\glsentrydescplural{\glslabel}}%
3123         {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
3124     }%
3125     }%
3126     {%
```

Make first letter upper case

```
3127     \ifglsused\glslabel
3128     {%
```

Subsequent use. (Expansion was used in version 3.07 and below in case the name wasn't the first thing to be displayed, but now the user can sort out the upper casing in `\defglsentryfmt`, which avoids the issues caused by fragile commands.)

```
3129     \ifbool{glscompatible-3.07}%
3130     {%
3131       \protected@edef\@glo@etext{%
3132         #2{\glsentryplural{\glslabel}}%
3133         {\glsentrydescplural{\glslabel}}%
3134         {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3135       \xmakefirstuc\@glo@etext
3136     }%
```

```

3137      {%
3138      #2{\Glsentryplural{\glslabel}}%
3139      {\glsentrydescplural{\glslabel}}%
3140      {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
3141      }%
3142      }%
3143      {%

```

First use

```

3144      \ifbool{glscompatible-3.07}%
3145      {%
3146      \protected@edef\@glo@etext{%
3147      #1{\Glsentryfirstplural{\glslabel}}%
3148      {\glsentrydescplural{\glslabel}}%
3149      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3150      \xmakefirstuc\@glo@etext
3151      }%
3152      {%
3153      #1{\Glsentryfirstplural{\glslabel}}%
3154      {\glsentrydescplural{\glslabel}}%
3155      {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
3156      }%
3157      }%
3158      }%
3159      {%

```

Make all upper case

```

3160      \ifglsused\glslabel
3161      {%

```

Subsequent use

```

3162      \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
3163      {\glsentrydescplural{\glslabel}}%
3164      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3165      }%
3166      {%

```

First use

```

3167      \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
3168      {\glsentrydescplural{\glslabel}}%
3169      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3170      }%
3171      }%
3172      }%
3173      {%

```

Singular form

```

3174      \glscapscase
3175      {%

```

Don't adjust case

```

3176      \ifglsused\glslabel

```

```

3177      {%
Subsequent use
3178      #2{\glsentrytext{\glslabel}}%
3179      {\glsentrydesc{\glslabel}}%
3180      {\glsentrysymbol{\glslabel}}{\glsinsert}%
3181      }%
3182      {%

First use
3183      #1{\glsentryfirst{\glslabel}}%
3184      {\glsentrydesc{\glslabel}}%
3185      {\glsentrysymbol{\glslabel}}{\glsinsert}%
3186      }%
3187      }%
3188      {%

Make first letter upper case
3189      \ifglused\glslabel
3190      {%

Subsequent use
3191      \ifbool{glscompatible-3.07}%
3192      {%
3193      \protected@edef\@glo@etext{%
3194      #2{\glsentrytext{\glslabel}}%
3195      {\glsentrydesc{\glslabel}}%
3196      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
3197      \xmakefirstuc\@glo@etext
3198      }%
3199      {%
3200      #2{\Glsentrytext{\glslabel}}%
3201      {\glsentrydesc{\glslabel}}%
3202      {\glsentrysymbol{\glslabel}}{\glsinsert}%
3203      }%
3204      }%
3205      {%

First use
3206      \ifbool{glscompatible-3.07}%
3207      {%
3208      \protected@edef\@glo@etext{%
3209      #1{\glsentryfirst{\glslabel}}%
3210      {\glsentrydesc{\glslabel}}%
3211      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
3212      \xmakefirstuc\@glo@etext
3213      }%
3214      {%
3215      #1{\Glsentryfirst{\glslabel}}%
3216      {\glsentrydesc{\glslabel}}%
3217      {\glsentrysymbol{\glslabel}}{\glsinsert}%

```

```

3218         }%
3219     }%
3220 }%
3221 {%

    Make all upper case
3222     \ifglused\glslabel
3223     {%

        Subsequent use
3224         \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}}%
3225         {\glsentrydesc{\glslabel}}}%
3226         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3227     }%
3228     {%

        First use
3229         \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}}%
3230         {\glsentrydesc{\glslabel}}}%
3231         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3232     }%
3233 }%
3234 }%
3235 }%
3236 {%

    Custom text provided in \glsdisp
3237     \ifglused{\glslabel}%
3238     {%

        Subsequent use
3239         #2{\glscustomtext}%
3240         {\glsentrydesc{\glslabel}}}%
3241         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3242     }%
3243     {%

        First use
3244         #1{\glscustomtext}%
3245         {\glsentrydesc{\glslabel}}}%
3246         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3247     }%
3248 }%
3249 }

```

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

```

3250 \newcommand*{\glsentryfmt}{%
3251     \ifdefempty\glscustomtext
3252     {%
3253         \glsifplural
3254         {%

```

Plural form

3255 \glscapscase
3256 {%

Don't adjust case

3257 \ifglused\glslabel
3258 {%

Subsequent use

3259 \glentryplural{\glslabel}\glinsert
3260 }%
3261 {%

First use

3262 \glentryfirstplural{\glslabel}\glinsert
3263 }%
3264 }%
3265 {%

Make first letter upper case

3266 \ifglused\glslabel
3267 {%

Subsequent use.

3268 \Glsentryplural{\glslabel}\glinsert
3269 }%
3270 {%

First use

3271 \Glsentryfirstplural{\glslabel}\glinsert
3272 }%
3273 }%
3274 {%

Make all upper case

3275 \ifglused\glslabel
3276 {%

Subsequent use

3277 \mfirstucMakeUppercase
3278 {\glentryplural{\glslabel}\glinsert}%
3279 }%
3280 {%

First use

3281 \mfirstucMakeUppercase
3282 {\glentryfirstplural{\glslabel}\glinsert}%
3283 }%
3284 }%
3285 }%
3286 {%

Singular form

```
3287 \glscapscase
3288 {%
```

Don't adjust case

```
3289 \ifglused\glslabel
3290 {%
```

Subsequent use

```
3291 \glentrytext{\glslabel}\glsinsert
3292 }%
3293 {%
```

First use

```
3294 \glentryfirst{\glslabel}\glsinsert
3295 }%
3296 }%
3297 {%
```

Make first letter upper case

```
3298 \ifglused\glslabel
3299 {%
```

Subsequent use

```
3300 \Glentrytext{\glslabel}\glsinsert
3301 }%
3302 {%
```

First use

```
3303 \Glentryfirst{\glslabel}\glsinsert
3304 }%
3305 }%
3306 {%
```

Make all upper case

```
3307 \ifglused\glslabel
3308 {%
```

Subsequent use

```
3309 \mfirstucMakeUppercase{\glentrytext{\glslabel}\glsinsert}%
3310 }%
3311 {%
```

First use

```
3312 \mfirstucMakeUppercase{\glentryfirst{\glslabel}\glsinsert}%
3313 }%
3314 }%
3315 }%
3316 }%
3317 {%
```

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

```
3318 \glscustomtext\glsinsert
```

```

3319 }%
3320 }

```

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

```

3321 \newcommand*{\glsgenacfmt}{%
3322   \ifdefempty\glscustomtext
3323   {%
3324     \ifglused\glslabel
3325     {%

```

Subsequent use:

```

3326     \glsifplural
3327     {%

```

Subsequent plural form:

```

3328     \glscapscase
3329     {%

```

Subsequent plural form, don't adjust case:

```

3330     \acronymfont{\glstryshortpl{\glslabel}}\glsinsert
3331     }%
3332     {%

```

Subsequent plural form, make first letter upper case:

```

3333     \acronymfont{\Glstryshortpl{\glslabel}}\glsinsert
3334     }%
3335     {%

```

Subsequent plural form, all caps:

```

3336     \mfirstucMakeUppercase
3337     {\acronymfont{\glstryshortpl{\glslabel}}\glsinsert}%
3338     }%
3339     }%
3340     {%

```

Subsequent singular form

```

3341     \glscapscase
3342     {%

```

Subsequent singular form, don't adjust case:

```

3343     \acronymfont{\glstryshort{\glslabel}}\glsinsert
3344     }%
3345     {%

```

Subsequent singular form, make first letter upper case:

```

3346     \acronymfont{\Glstryshort{\glslabel}}\glsinsert
3347     }%
3348     {%

```

Subsequent singular form, all caps:

```

3349     \mfirstucMakeUppercase
3350     {\acronymfont{\glstryshort{\glslabel}}\glsinsert}%

```


3351 }%
 3352 }%
 3353 }%
 3354 {%

First use:

3355 \glsifplural
 3356 {%

First use plural form:

3357 \glscapscase
 3358 {%

First use plural form, don't adjust case:

3359 \genplacrformat{\glslabel}{\glsinsert}%
 3360 }%
 3361 {%

First use plural form, make first letter upper case:

3362 \Genplacrformat{\glslabel}{\glsinsert}%
 3363 }%
 3364 {%

First use plural form, all caps:

3365 \mfirstucMakeUppercase
 3366 {\genplacrformat{\glslabel}{\glsinsert}}}%
 3367 }%
 3368 }%
 3369 {%

First use singular form

3370 \glscapscase
 3371 {%

First use singular form, don't adjust case:

3372 \genacrformat{\glslabel}{\glsinsert}%
 3373 }%
 3374 {%

First use singular form, make first letter upper case:

3375 \Genacrformat{\glslabel}{\glsinsert}%
 3376 }%
 3377 {%

First use singular form, all caps:

3378 \mfirstucMakeUppercase
 3379 {\genacrformat{\glslabel}{\glsinsert}}}%
 3380 }%
 3381 }%
 3382 }%
 3383 }%
 3384 {%

User supplied text.

```
3385 \glscustomtext
3386 }%
3387 }
```

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

The full format used by \glsgenacfmt (singular).

```
3388 \newcommand*{\genacrfullformat}[2]{%
3389 \glentrylong{#1}#2\space
3390 (\protect\firstacronymfont{\glentryshort{#1}})%
3391 }
```

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

As above but makes the first letter upper case.

```
3392 \newcommand*{\Genacrfullformat}[2]{%
3393 \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
3394 \xmakefirstuc\gls@text
3395 }
```

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

The full format used by \glsgenacfmt (plural).

```
3396 \newcommand*{\genplacrfullformat}[2]{%
3397 \glentrylongpl{#1}#2\space
3398 (\protect\firstacronymfont{\glentryshortpl{#1}})%
3399 }
```

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

As above but makes the first letter upper case.

```
3400 \newcommand*{\Genplacrfullformat}[2]{%
3401 \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
3402 \xmakefirstuc\gls@text
3403 }
```

glsdisplayfirst Deprecated. Kept for backward compatibility.

```
3404 \newcommand*{\glsdisplayfirst}[4]{#1#4}
```

\glsdisplay Deprecated. Kept for backward compatibility.

```
3405 \newcommand*{\glsdisplay}[4]{#1#4}
```

`\defglsdisplay` Deprecated. Kept for backward compatibility.

```
3406 \newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
3407   \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
3408   Use \string\defglsentryfmt\space instead}%
3409   \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
3410   \edef\@gls@doentrydef{%
3411     \noexpand\defglsentryfmt[#1]{%
3412       \noexpand\ifcsdef{gls@#1@displayfirst}%
3413       {%
3414         \noexpand\@@gls@default@entryfmt
3415         {\noexpand\csuse{gls@#1@displayfirst}}}%
3416         {\noexpand\csuse{gls@#1@display}}}%
3417       }%
3418       {%
3419         \noexpand\@@gls@default@entryfmt
3420         {\noexpand\glsdisplayfirst}%
3421         {\noexpand\csuse{gls@#1@display}}}%
3422       }%
3423     }%
3424   }%
3425   \@gls@doentrydef
3426 }
```

`\glsdisplayfirst` Deprecated. Kept for backward compatibility.

```
3427 \newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
3428   \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
3429   Use \string\defglsentryfmt\space instead}%
3430   \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
3431   \edef\@gls@doentrydef{%
3432     \noexpand\defglsentryfmt[#1]{%
3433       \noexpand\ifcsdef{gls@#1@display}%
3434       {%
3435         \noexpand\@@gls@default@entryfmt
3436         {\noexpand\csuse{gls@#1@displayfirst}}}%
3437         {\noexpand\csuse{gls@#1@display}}}%
3438       }%
3439       {%
3440         \noexpand\@@gls@default@entryfmt
3441         {\noexpand\csuse{gls@#1@displayfirst}}}%
3442         {\noexpand\glsdisplay}%
3443       }%
3444     }%
3445   }%
3446   \@gls@doentrydef
3447 }
```

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 \TeX 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 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.

```
3448 \define@key{glslink}{counter}{%
3449   \ifcsundef{c@#1}%
3450   {%
3451     \PackageError{glossaries}%
3452     {There is no counter called '#1'}%
3453     {%
3454       The counter key should have the name of a valid counter
3455       as its value%
3456     }%
3457   }%
3458   {%
3459     \def\@gls@counter{#1}%
3460   }%
3461 }
```

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.

```
3462 \define@key{glslink}{format}{%
3463   \def\@glsnumberformat{#1}}
```

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.

```
3464 \define@boolkey{glslink}{hyper}[true]{}
```

Initialise hyper key.

```
3465 \ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
```

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.

```
3466 \define@boolkey{glslink}{local}[true]{}
```

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.

```
3467 \newcommand*\glslinkvar[3]{#1}
```

`\glsifhyper` Now deprecated.

```
3468 \newcommand*\glsifhyper[2]{%
3469 \glslinkvar{#1}{#2}{#1}%
3470 \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
3471 you mean \string\glsifhyperon\space or \string\glslinkvar?}%
3472 }
```

`\@gls@hyp@opt` Used by the commands such as `\glslink` to determine whether to modify the hyper option.

```
3473 \newcommand*\@gls@hyp@opt[1]{%
3474 \let\glslinkvar\@firstofthree
3475 \let\@gls@hyp@opt@cs#1\relax
3476 \@ifstar{\s@gls@hyp@opt}%
3477 {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
3478 }
```

`\s@gls@hyp@opt` Starred version

```
3479 \newcommand*\s@gls@hyp@opt[1][]{%
3480 \let\glslinkvar\@secondofthree
3481 \@gls@hyp@opt@cs[hyper=false,#1]}
```

`\p@gls@hyp@opt` Plus version

```
3482 \newcommand*\p@gls@hyp@opt[1][]{%
3483 \let\glslinkvar\@thirdofthree
3484 \@gls@hyp@opt@cs[hyper=true,#1]}
```

Syntax:

```
\glslink[options]{label}{text}
```

Display `<text>` in the document, and add the entry information for `<label>` into the relevant glossary. The optional argument should be a key value list using the `\glslink` keys defined above.

There is also a starred version:

```
\glslink*{options}{label}{text}
```

which is equivalent to `\glslink[hyper=false, options]{label}{text}`

First determine which version is being used:

`\glslink`

```
3485 \newrobustcmd*{\glslink}{%  
3486 \@gls@hyp@opt\@gls@@link  
3487 }
```

`\@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.

```
3488 \newcommand*{\@gls@@link}[3][\@gls@link]{%  
3489 \glsdoifexistsordo{#2}%  
3490 {%  
3491 \let\do@gls@link@checkfirsthyper\relax  
3492 \@gls@link[#1]{#2}{#3}%  
3493 }%
```

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

```
3494 \glstextformat{#3}%  
3495 }%  
  
3496 \glspostlinkhook  
3497 }
```

`glspostlinkhook`

```
3498 \newcommand*{\glspostlinkhook}{}
```

`checkfirsthyper` Check for first use and switch off hyper key if hyperlink not wanted. (Should be off if first use and `hyper=false` is on or if first use and both the entry is in an acronym list and the `acrfootnote` setting is on.) This assumes the glossary type is stored in `\glstype` and the label is stored in `\glslabel`.

```
3499 \newcommand*{\@gls@link@checkfirsthyper}{%  
3500 \ifglsused{\glslabel}%  
3501 {%  
3502 }%  
3503 {%  
3504 \gls@checkisacronymlist\glstype  
3505 \ifglshyperfirst  
3506 \if@glsisacronymlist  
3507 \ifglsacrfootnote  
3508 \KV@glslink@hyperfalse  
3509 \fi  
3510 \fi  
3511 \else  
3512 \KV@glslink@hyperfalse  
3513 \fi  
3514 }%
```

Allow user to hook into this

```
3515 \glslinkcheckfirsthyperhook  
3516 }
```

`checkfirsthyperhook` Allow used to hook into the `\@gls@link@checkfirsthyper` macro

```
3517 \newcommand*{\glslinkcheckfirsthyperhook}{}

linkpostsetkeys
3518 \newcommand*{\glslinkpostsetkeys}{}

\glsifhyperon Check the value of the hyper key:
3519 \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}

ablehyperinlist Disable hyperlink if in the “nohyper” list.
3520 \newcommand*{\do@glssdisablehyperinlist}{%
3521   \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
3522   {\KV@glslink@hyperfalse}}%
3523 }

let@glslink@opts Hook to set default options for \@glslink.
3524 \newcommand*{\@gls@setdefault@glslink@opts}{}

\@gls@link
3525 \def\@gls@link[#1]#2#3{%
  Inserting \leavevmode suggested by Donald Arseneau (avoids problem with tabularx).
3526   \leavevmode
3527   \protected@edef\glslabel{\glsdetoklabel{#2}}%
  Save options in \@gls@link@opts and label in \@gls@link@label
3528   \def\@gls@link@opts{#1}%
3529   \let\@gls@link@label\glslabel
3530   \def\@glsnumberformat{glsnumberformat}%
3531   \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  If this is in one of the “nohypertypes” glossaries, suppress the hyperlink by default
3532   \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  Save original setting
3533   \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  Set defaults:
3534   \@gls@setdefault@glslink@opts
  Switch off hyper setting if the glossary type has been identified in nohyperlist.
3535   \do@glssdisablehyperinlist
  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.
3536   \do@gls@link@checkfirsthyper
3537   \setkeys{glslink}{#1}%
  Add a hook for the user to customise things after the keys have been set.
3538   \glslinkpostsetkeys
```

```

Store the entry's counter in \theglentrycounter
3539 \@gls@saveentrycounter

Define sort key if necessary:
3540 \@gls@setsort{\glslabel}%

(De-tok'ing done by \@do@wrglossary)
3541 \@do@wrglossary{#2}%
3542 \ifKV@glslink@hyper
3543 \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
3544 \else

3545 \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
3546 \fi

Restore original setting
3547 \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
3548 }

\glolinkprefix
3549 \newcommand*{\glolinkprefix}{glo:}

glsentrycounter Set default value of entry counter
3550 \def\glsentrycounter{\glscounter}%

saveentrycounter Need to check if using equation counter in align environment:
3551 \newcommand*{\@gls@saveentrycounter}{%
3552 \def\@gls@Hcounter{}%

Are we using equation counter?
3553 \ifthenelse{\equal{\@gls@counter}{equation}}{%
3554 {

If we're in align environment, \xatlevel@ will be defined. (Can't test for \@currentvir as
may be inside an inner environment.)
3555 \ifcsundef{xatlevel@}%
3556 {%
3557 \edef\theglentrycounter{\expandafter\noexpand
3558 \csname the\@gls@counter\endcsname}%
3559 }%
3560 {%
3561 \ifx\xatlevel@\@empty
3562 \edef\theglentrycounter{\expandafter\noexpand
3563 \csname the\@gls@counter\endcsname}%
3564 \else
3565 \savecounters@
3566 \advance\c@equation by 1\relax
3567 \edef\theglentrycounter{\csname the\@gls@counter\endcsname}%

```


Check if hyperref version of this counter

```

3568     \ifcsundef{theH\@gls@counter}%
3569     {%
3570         \def\@gls@Hcounter{\theglsentrycounter}%
3571     }%
3572     {%
3573         \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
3574     }%
3575     \protected@edef\theHglentrycounter{\@gls@Hcounter}%
3576     \restorecounters@
3577 \fi
3578 }%
3579 }%
3580 {%

```

Not using equation counter so no special measures:

```

3581     \edef\theglsentrycounter{\expandafter\noexpand
3582         \csname the\@gls@counter\endcsname}%
3583 }%

```

Check if hyperref version of this counter

```

3584 \ifx\@gls@Hcounter\@empty
3585     \ifcsundef{theH\@gls@counter}%
3586     {%
3587         \def\theHglentrycounter{\theglsentrycounter}%
3588     }%
3589     {%
3590         \protected@edef\theHglentrycounter{\expandafter\noexpand
3591             \csname theH\@gls@counter\endcsname}%
3592     }%
3593 \fi
3594 }

```

`t@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.

```

3595 \def\@set@glo@numformat#1#2#3#4{%
3596     \expandafter\@glo@check@mkidxrangechar#3\@nil
3597     \protected@edef#1{%
3598         \@glo@prefix setentrycounter[#4]{#2}%
3599         \expandafter\string\csname\@glo@suffix\endcsname
3600     }%
3601     \@gls@checkmkidxchars#1%
3602 }

```

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.

```

3603 \def\@glo@check@mkidxrangechar#1#2\@nil{%
3604 \if#1(\relax
3605   \def\@glo@prefix{()%
3606   \if\relax#2\relax
3607     \def\@glo@suffix{glsnumberformat}%
3608   \else
3609     \def\@glo@suffix{#2}%
3610   \fi
3611 \else
3612   \if#1)\relax
3613     \def\@glo@prefix{}}%
3614   \if\relax#2\relax
3615     \def\@glo@suffix{glsnumberformat}%
3616   \else
3617     \def\@glo@suffix{#2}%
3618   \fi
3619 \else
3620   \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
3621 \fi
3622 \fi}

```

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

```

3623 \newcommand*{\@gls@escbsdq}[1]{%
3624   \def\@gls@checkedmkidx{%
3625     \let\gls@xdystring=#1\relax
3626     \@onelevel@sanitize\gls@xdystring
3627     \edef\do@gls@xdycheckbackslash{%
3628       \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
3629       \@backslashchar\@backslashchar\noexpand\null}%
3630     \do@gls@xdycheckbackslash
3631     \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
3632     \def\@gls@checkedmkidx{%
3633       \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
3634       \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%

```

Unsanitize\gls@numberpage,\gls@alphpage,\gls@Alphpage and\gls@romanpage (thanks to David Carlisle for the suggestion.)

```

3635   \@for\@gls@tmp:=\gls@protected@pagefmts\do
3636   {%
3637     \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
3638     \@onelevel@sanitize\@gls@sanitized@tmp
3639     \edef\gls@dosubst{%
3640       \noexpand\DTLsubstituteall\noexpand\gls@xdystring
3641       {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
3642     }%
3643     \gls@dosubst
3644   }%

```

Assign to required control sequence

```

3645   \let#1=\gls@xdystring

```

3646 }

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

checkmkidxchars

```
3647 \newcommand{\@gls@checkmkidxchars}[1]{%
3648   \ifglxindy
3649     \@gls@escbsdq{#1}%
3650   \else
3651     \def\@gls@checkedmkidx{%
3652       \expandafter\@gls@checkquote#1\@nil""\null
3653       \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3654       \def\@gls@checkedmkidx{%
3655         \expandafter\@gls@checkescquote#1\@nil\""\null
3656         \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3657         \def\@gls@checkedmkidx{%
3658           \expandafter\@gls@checkescactual#1\@nil\??\null
3659           \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3660           \def\@gls@checkedmkidx{%
3661             \expandafter\@gls@checkactual#1\@nil??\null
3662             \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3663             \def\@gls@checkedmkidx{%
3664               \expandafter\@gls@checkbar#1\@nil||\null
3665               \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3666               \def\@gls@checkedmkidx{%
3667                 \expandafter\@gls@checkescbar#1\@nil\\|\null
3668                 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3669                 \def\@gls@checkedmkidx{%
3670                   \expandafter\@gls@checklevel#1\@nil!!\null
3671                   \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3672                   \def\@gls@checkedmkidx{%
3673                     \expandafter\@gls@checkesclevel#1\@nil\!|\null
3674                     \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3675                     \def\@gls@checkedmkidx{%
3676                       \edef\@do@gls@checkesc{%
3677                         \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
3678                         \@backslashchar\@backslashchar\noexpand\null
3679                       }%
3680                       \@do@gls@checkesc
3681                       \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3682                     \fi
3683 }
```

Update the control sequence and strip trailing \@nil:

s@updatechecked

```
3684 \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
```

\@gls@tmpb Define temporary token

```
3685 \newtoks\@gls@tmpb
```

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

```
3686 \def\@gls@checkquote#1"#2"#3\null{%
3687   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3688   \toks@={#1}%
3689   \ifx\null#2\null
3690   \ifx\null#3\null
3691     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3692     \def\@gls@checkquote{\relax}%
3693   \else
3694     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3695       \@gls@quotechar\@gls@quotechar\@gls@quotechar}%
3696     \def\@gls@checkquote{\@gls@checkquote#3\null}%
3697   \fi
3698 \else
3699   \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3700     \@gls@quotechar\@gls@quotechar}%
3701   \ifx\null#3\null
3702     \def\@gls@checkquote{\@gls@checkquote#2""\null}%
3703   \else
3704     \def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
3705   \fi
3706 \fi
3707 \@gls@checkquote
3708 }
```

s@checkescquote Do the same for \":

```
3709 \def\@gls@checkescquote#1\"#2\"#3\null{%
3710   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3711   \toks@={#1}%
3712   \ifx\null#2\null
3713   \ifx\null#3\null
3714     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3715     \def\@gls@checkescquote{\relax}%
3716   \else
3717     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3718       \@gls@quotechar\string\"@gls@quotechar
3719       \@gls@quotechar\string\"@gls@quotechar}%
3720     \def\@gls@checkescquote{\@gls@checkescquote#3\null}%
3721   \fi
3722 \else
3723   \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3724     \@gls@quotechar\string\"@gls@quotechar}%
3725   \ifx\null#3\null
3726     \def\@gls@checkescquote{\@gls@checkescquote#2\"\" \null}%
3727   \else
3728     \def\@gls@checkescquote{\@gls@checkescquote#2\"#3\null}%
3729   \fi
3730 \fi
3731 \@gls@checkescquote
```

3732 }

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

```
3733 \def\@gls@checkescactual#1\?#2\?#3\null{%
3734 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3735 \toks@={#1}%
3736 \ifx\null#2\null
3737 \ifx\null#3\null
3738 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3739 \def\@gls@checkescactual{\relax}%
3740 \else
3741 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3742 \@gls@quotechar\string\" \@gls@actualchar
3743 \@gls@quotechar\string\" \@gls@actualchar}%
3744 \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
3745 \fi
3746 \else
3747 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3748 \@gls@quotechar\string\" \@gls@actualchar}%
3749 \ifx\null#3\null
3750 \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
3751 \else
3752 \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
3753 \fi
3754 \fi
3755 \@gls@checkescactual
3756 }
```

gls@checkeschar Similarly for \|:

```
3757 \def\@gls@checkeschar#1\|#2\|#3\null{%
3758 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3759 \toks@={#1}%
3760 \ifx\null#2\null
3761 \ifx\null#3\null
3762 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3763 \def\@gls@checkeschar{\relax}%
3764 \else
3765 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3766 \@gls@quotechar\string\" \@gls@encapchar
3767 \@gls@quotechar\string\" \@gls@encapchar}%
3768 \def\@gls@checkeschar{\@gls@checkeschar#3\null}%
3769 \fi
3770 \else
3771 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3772 \@gls@quotechar\string\" \@gls@encapchar}%
3773 \ifx\null#3\null
3774 \def\@gls@checkeschar{\@gls@checkeschar#2\|\|\null}%
3775 \else
3776 \def\@gls@checkeschar{\@gls@checkeschar#2\|#3\null}%

```

```

3777 \fi
3778 \fi
3779 \@gls@checkescbar
3780 }

```

s@checkesclevel Similarly for \!:

```

3781 \def\@gls@checkesclevel#1\!#2\!#3\null{%
3782 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3783 \toks@={#1}%
3784 \ifx\null#2\null
3785 \ifx\null#3\null
3786 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3787 \def\@gls@checkesclevel{\relax}%
3788 \else
3789 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3790 \@gls@quotechar\string"\@gls@levelchar
3791 \@gls@quotechar\string"\@gls@levelchar}%
3792 \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
3793 \fi
3794 \else
3795 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3796 \@gls@quotechar\string"\@gls@levelchar}%
3797 \ifx\null#3\null
3798 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
3799 \else
3800 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
3801 \fi
3802 \fi
3803 \@gls@checkesclevel
3804 }

```

\@gls@checkbar and for |:

```

3805 \def\@gls@checkbar#1|#2|#3\null{%
3806 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3807 \toks@={#1}%
3808 \ifx\null#2\null
3809 \ifx\null#3\null
3810 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3811 \def\@gls@checkbar{\relax}%
3812 \else
3813 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3814 \@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
3815 \def\@gls@checkbar{\@gls@checkbar#3\null}%
3816 \fi
3817 \else
3818 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3819 \@gls@quotechar\@gls@encapchar}%
3820 \ifx\null#3\null
3821 \def\@gls@checkbar{\@gls@checkbar#2||\null}%

```

```

3822 \else
3823 \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
3824 \fi
3825 \fi
3826 \@gls@checkbar
3827 }

```

@gls@checklevel and for !:

```

3828 \def\@gls@checklevel#1!#2!#3\null{%
3829 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3830 \toks@={#1}%
3831 \ifx\null#2\null
3832 \ifx\null#3\null
3833 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3834 \def\@gls@checklevel{\relax}%
3835 \else
3836 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3837 \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
3838 \def\@gls@checklevel{\@gls@checklevel#3\null}%
3839 \fi
3840 \else
3841 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3842 \@gls@quotechar\@gls@levelchar}%
3843 \ifx\null#3\null
3844 \def\@gls@checklevel{\@gls@checklevel#2!!\null}%
3845 \else
3846 \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
3847 \fi
3848 \fi
3849 \@gls@checklevel
3850 }

```

gls@checkactual and for ?:

```

3851 \def\@gls@checkactual#1?#2?#3\null{%
3852 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3853 \toks@={#1}%
3854 \ifx\null#2\null
3855 \ifx\null#3\null
3856 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3857 \def\@gls@checkactual{\relax}%
3858 \else
3859 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3860 \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
3861 \def\@gls@checkactual{\@gls@checkactual#3\null}%
3862 \fi
3863 \else
3864 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3865 \@gls@quotechar\@gls@actualchar}%
3866 \ifx\null#3\null

```

```

3867     \def\@gls@checkactual{\@gls@checkactual#2??\null}%
3868     \else
3869     \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
3870     \fi
3871     \fi
3872     \@gls@checkactual
3873 }

```

\@gls@checkesc and for \:

```

3874 \bgroup
3875 \catcode'\|0\relax
3876 \catcode'\|12\relax
3877 |gdef|@gls@checkesc#1\#2\#3|null{%
3878   |@gls@tmpb=|expandafter{|@gls@checkedmkidx}%
3879   |toks@={#1}%
3880   |ifx|null#2|null
3881     |ifx|null#3|null
3882     |edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@}%
3883     |def|@gls@checkesc{|relax}%
3884     |else
3885     |edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
3886       |@gls@quotechar|@backslashchar|@gls@quotechar|@backslashchar}%
3887     |def|@gls@checkesc{|@gls@checkesc#3|null}%
3888     |fi
3889     |else
3890     |edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
3891       |@gls@quotechar|@backslashchar}%
3892     |ifx|null#3|null
3893     |def|@gls@checkesc{|@gls@checkesc#2\\|null}%
3894     |else
3895     |def|@gls@checkesc{|@gls@checkesc#2\#3|null}%
3896     |fi
3897     |fi
3898     |@gls@checkesc
3899 }
3900 |egroup

```

s@xdycheckquote As before but for use with xindy

```

3901 \def\@gls@xdycheckquote#1"#2"#3\null{%
3902   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3903   \toks@={#1}%
3904   \ifx\null#2\null
3905     \ifx\null#3\null
3906     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3907     \def\@gls@xdycheckquote{\relax}%
3908     \else
3909     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3910       \string\string\}%
3911     \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%

```



```

3912     \fi
3913 \else
3914     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3915         \string\}%
3916     \ifx\null#3\null
3917         \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
3918     \else
3919         \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
3920     \fi
3921 \fi
3922 \@gls@xdycheckquote
3923 }

```

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

```

3924 \edef\def\@gls@xdycheckbackslash{%
3925     \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
3926         ##2\@backslashchar##3\noexpand\null{%
3927     \noexpand\@gls@tmpb=\noexpand\expandafter
3928         {\noexpand\@gls@checkedmkidx}%
3929     \noexpand\toks@={##1}%
3930     \noexpand\ifx\noexpand\null##2\noexpand\null
3931     \noexpand\ifx\noexpand\null##3\noexpand\null
3932     \noexpand\edef\noexpand\@gls@checkedmkidx{%
3933         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
3934     \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
3935     \noexpand\else
3936     \noexpand\edef\noexpand\@gls@checkedmkidx{%
3937         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
3938         \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
3939     \noexpand\def\noexpand\@gls@xdycheckbackslash{%
3940         \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
3941     \noexpand\fi
3942     \noexpand\else
3943     \noexpand\edef\noexpand\@gls@checkedmkidx{%
3944         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
3945         \@backslashchar\@backslashchar}%
3946     \noexpand\ifx\noexpand\null##3\noexpand\null
3947     \noexpand\def\noexpand\@gls@xdycheckbackslash{%
3948         \noexpand\@gls@xdycheckbackslash##2\@backslashchar
3949         \@backslashchar\noexpand\null}%
3950     \noexpand\else
3951     \noexpand\def\noexpand\@gls@xdycheckbackslash{%
3952         \noexpand\@gls@xdycheckbackslash##2\@backslashchar
3953         ##3\noexpand\null}%
3954     \noexpand\fi
3955     \noexpand\fi
3956     \noexpand\@gls@xdycheckbackslash
3957 }%
3958 }

```

Now go ahead and define \@gls@xdycheckbackslash

```
3959 \def@gls@xdycheckbackslash
```

lsdohypertarget

```
3960 \newlength\gls@tmplen
3961 \newcommand*{\glsdohypertarget}[2]{%
3962   \@glsshowtarget{#1}%
3963   \settoheight{\gls@tmplen}{#2}%
3964   \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
3965 }
```

\glsdohyperlink

```
3966 \newcommand*{\glsdohyperlink}[2]{%
3967   \@glsshowtarget{#1}%
3968   \hyperlink{#1}{#2}%
3969 }
```

lsdonohyperlink

```
3970 \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.

```
3971 \ifcsundef{hyperlink}%
3972 {%
3973   \let\@glslink\glsdonohyperlink
3974 }%
3975 {%
3976   \let\@glslink\glsdohyperlink
3977 }
```

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

```
3978 \ifcsundef{hypertarget}%
3979 {%
3980   \let\@glstarget\@secondoftwo
3981 }%
3982 {%
3983   \let\@glstarget\glsdohypertarget
3984 }
```

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

glsdisablehyper

```
3985 \newcommand{\glsdisablehyper}{%
3986   \KV@glslink@hyperfalse
3987   \let\@glslink\glsdonohyperlink
3988   \let\@glstarget\@secondoftwo
3989 }
```

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

`\glsenablehyper`

```
3990 \newcommand{\glsenablehyper}{%
3991   \KV@glslink@hypertrue
3992   \let\@glslink\glsdohyperlink
3993   \let\@glstarget\glsdohypertarget
3994 }
```

Provide some convenience commands if not already defined:

```
3995 \providecommand{\@firstofthree}[3]{#1}
3996 \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`

```
3997 \newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
```

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

`\@gls`

```
3998 \newcommand*{\@gls}[2][]{%
3999   \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
4000 }
```

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

```
4001 \def\@gls@#1#2[#3]{%
4002   \glsdoifexists{#2}%
4003   {%
4004     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

4005     \let\glsifplural\@secondoftwo
4006     \let\glsapscase\@firstofthree
4007     \let\glsustomtext\@empty
4008     \def\glsinsert{#3}%
```

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

```
4009 \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.

```
4010 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4011 \ifKV@gls@link@local
```

```
4012 \glslocalunset{#2}%
```

```
4013 \else
```

```
4014 \glsunset{#2}%
```

```
4015 \fi
```

```
4016 }%
```

```
4017 \gls@postlinkhook
```

```
4018 }
```

\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

```
4019 \newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
```

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

```
4020 \newcommand*{\@Gls}[2] [] {%
```

```
4021 \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}%
```

```
4022 }
```

\@Gls@ Read in the final optional argument:

```
4023 \def\@Gls@#1#2[#3] {%
```

```
4024 \glsdoifexists{#2}%
```

```
4025 {%
```

```
4026 \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
4027 \let\glsifplural\@secondoftwo
```

```
4028 \let\gls@scapscase\@secondofthree
```

```
4029 \let\gls@customtext\@empty
```

```
4030 \def\glsinsert{#3}%
```

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

```
4031 \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.

```
4032 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4033 \ifKV@glslink@local
4034 \glsllocalunset{#2}%
4035 \else
4036 \glsunset{#2}%
4037 \fi
4038 }%

4039 \glspostlinkhook
4040 }
```

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

\GLS

```
4041 \newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
```

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

```
4042 \newcommand*{\@GLS}[2][\@GLS@{#1}{#2}]{\@GLS@{#1}{#2}[]}%
4043 \new@ifnextchar[\@GLS@{#1}{#2}]{\@GLS@{#1}{#2}[]}%
4044 }
```

\@GLS@ Read in the final optional argument:

```
4045 \def\@GLS@#1#2[#3]{%
4046 \glsdoidexists{#2}%
4047 {%
4048 \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

4049 \let\glsifplural\@secondoftwo
4050 \let\glscapscase\@thirdofthree
4051 \let\glscustomtext\@empty
4052 \def\glsinsert{#3}%
```

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

```
4053 \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.

```
4054 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4055 \ifKV@glslink@local
4056 \glsllocalunset{#2}%
4057 \else
4058 \glsunset{#2}%
4059 \fi
4060 }%

4061 \glspostlinkhook
4062 }
```

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

`\glspl`

```
4063 \newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
```

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

```
4064 \newcommand*{\@glspl}[2] [] {%
4065   \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
4066 }
```

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

```
4067 \def\@glspl@#1#2[#3] {%
4068   \glsdoifexists{#2}%
4069   {%
4070     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

4071     \let\glsifplural\@firstoftwo
4072     \let\glsapscase\@firstofthree
4073     \let\glscustomtext\@empty
4074     \def\glsinsert{#3}%
```

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

```
4075   \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.

```
4076   \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4077   \ifKV@glslink@local
4078     \glslocalunset{#2}%
4079   \else
4080     \glsunset{#2}%
4081   \fi
4082 }%
```

```
4083 \glspostlinkhook
4084 }
```

`\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`

```
4085 \newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
```

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

```
4086 \newcommand*{\@Glspl}[2] [] {%
4087   \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
4088 }
```

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

```
4089 \def\@Glspl@#1#2[#3]{%
4090   \glsdoifexists{#2}%
4091   {%
4092     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
4093     \let\glsifplural\@firstoftwo
4094     \let\glscapscase\@secondofthree
4095     \let\glscustomtext\@empty
4096     \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 `\@gl@link` sets `\glstype`.

```
4097   \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

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

```
4098   \@gl@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4099   \ifKV@gl@link@local
4100     \gl@localunset{#2}%
4101   \else
4102     \gl@unset{#2}%
4103   \fi
4104 }%
4105 \gl@postlinkhook
4106 }
```

`\GLSp1` behaves like `\glsp1` except that all the link text is converted to uppercase.

`\GLSp1`

```
4107 \newrobustcmd*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
```

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

```
4108 \newcommand*{\@GLSp1}[2][ ]{%
4109   \new@ifnextchar[\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[ ]}%
4110 }
```

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

```
4111 \def\@GLSp1@#1#2[#3]{%
4112   \glsdoifexists{#2}%
4113   {%
4114     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
4115     \let\glsifplural\@firstoftwo
4116     \let\glscapscase\@thirdofthree
4117     \let\glscustomtext\@empty
4118     \def\glsinsert{#3}%
```

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

```
4119 \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.

```
4120 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4121 \ifKV@gls@link@local
```

```
4122 \glslocalunset{#2}%
```

```
4123 \else
```

```
4124 \glsunset{#2}%
```

```
4125 \fi
```

```
4126 }%
```

```
4127 \gls@postlinkhook
```

```
4128 }
```

`\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:

```
4129 \newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
```

Defined the un-starred form.

`\@glsdisp`

```
4130 \newcommand*{\@glsdisp}[3] [] {%
```

```
4131 \glsdoifexists{#2}{%
```

```
4132 \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
4133 \let\glsifplural\@secondoftwo
```

```
4134 \let\gls@scapscase\@firstofthree
```

```
4135 \def\gls@customtext{#3}%
```

```
4136 \def\glsinsert{}%
```

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

```
4137 \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.

```
4138 \@gls@link[#1]{#2}{\@glo@text}%
```


Indicate that this entry has now been used

```
4139 \ifKV@glslink@local
4140 \glsllocalunset{#2}%
4141 \else
4142 \glunset{#2}%
4143 \fi
4144 }%

4145 \glspostlinkhook
4146 }
```

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

```
4147 \newcommand*{\@gl@link@nocheckfirsthyper}{}
```

@gl@field@link

```
4148 \newcommand{\@gl@field@link}[3]{%
4149 \glsoifexists{#2}%
4150 {%
4151 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4152 \@gl@link[#1]{#2}{#3}%
4153 }%

4154 \glspostlinkhook
4155 }
```

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

\glstext

```
4156 \newrobustcmd*{\glstext}{\@gl@hyp@opt\@glstext}
```

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

```
4157 \newcommand*{\@glstext}[2][{}]{%
4158 \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4159 \def\@glstext@#1#2[#3]{%
4160 \@gl@field@link{#1}{#2}{\glstrytext{#2}#3}%
4161 }
```

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

\GLStext

```
4162 \newrobustcmd*{\GLStext}{\@gl@hyp@opt\@GLStext}
```

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

```
4163 \newcommand*{\@GLStext}[2][{}]{%
4164 \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4165 \def\@GLStext@#1#2[#3]{%
4166   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrytext{#2}#3}}%
4167 }
```

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

`\Glstext`

```
4168 \newrobustcmd*{\Glstext}{\@gls@hyp@opt\@GLstext}
```

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

```
4169 \newcommand*{\@GLstext}[2] [] {%
4170   \new@ifnextchar[{\@GLstext@{#1}{#2}}{\@GLstext@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4171 \def\@GLstext@#1#2[#3]{%
4172   \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
4173 }
```

`\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`

```
4174 \newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
```

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

```
4175 \newcommand*{\@glsfirst}[2] [] {%
4176   \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4177 \def\@glsfirst@#1#2[#3]{%
4178   \@gls@field@link{#1}{#2}{\glstentryfirst{#2}#3}%
4179 }
```

`\Glsfirst` behaves like `\glsfirst` except it displays the first letter in uppercase.

`\Glsfirst`

```
4180 \newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
```

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

```
4181 \newcommand*{\@Glsfirst}[2] [] {%
4182   \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4183 \def\@Glsfirst@#1#2[#3]{%
4184   \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
4185 }
```

`\GLSfirst` behaves like `\Glsfirst` except it displays the text in uppercase.

`\GLSfirst`

```
4186 \newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
```

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

```
4187 \newcommand*{\@GLSfirst}[2] [] {%  
4188   \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4189 \def\@GLSfirst@#1#2[#3] {%  
4190   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%  
4191 }
```

`\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`

```
4192 \newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
```

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

```
4193 \newcommand*{\@glsplural}[2] [] {%  
4194   \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4195 \def\@glsplural@#1#2[#3] {%  
4196   \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}}%  
4197 }
```

`\Glsplural` behaves like `\glsplural` except that the first letter is converted to uppercase.

`\Glsplural`

```
4198 \newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
```

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

```
4199 \newcommand*{\@Glsplural}[2] [] {%  
4200   \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4201 \def\@Glsplural@#1#2[#3] {%  
4202   \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}}%  
4203 }
```

`\GLSplural` behaves like `\glsplural` except that the text is converted to uppercase.

`\GLSplural`

```
4204 \newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
```

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

```
4205 \newcommand*{\@GLSplural}[2] [] {%  
4206   \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4207 \def\@GLSplural@#1#2[#3] {%  
4208   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%  
4209 }
```

`\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`

```
4210 \newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
```

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

```
4211 \newcommand*{\@glsfirstplural}[2] [] {%
```

```
4212   \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4213 \def\@glsfirstplural@#1#2[#3] {%
```

```
4214   \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
```

```
4215 }
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the first letter is converted to uppercase.

`\Glsfirstplural`

```
4216 \newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
```

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

```
4217 \newcommand*{\@Glsfirstplural}[2] [] {%
```

```
4218   \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4219 \def\@Glsfirstplural@#1#2[#3] {%
```

```
4220   \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
```

```
4221 }
```

`\GLSfirstplural` behaves like `\glsfirstplural` except that the link text is converted to uppercase.

`\GLSfirstplural`

```
4222 \newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
```

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

```
4223 \newcommand*{\@GLSfirstplural}[2] [] {%
```

```
4224   \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4225 \def\@GLSfirstplural@#1#2[#3] {%
```

```
4226   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
```

```
4227 }
```

`\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`

```
4228 \newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
```

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

```
4229 \newcommand*{\@glsname}[2] [] {%
```

```
4230   \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4231 \def\@glsname@#1#2[#3]{%
4232   \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
4233 }
```

\Glsname behaves like \glsname except that the first letter is converted to uppercase.

\Glsname

```
4234 \newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
```

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

```
4235 \newcommand*{\@Glsname}[2][\%
4236   \new@ifnextchar[\@Glsname@{#1}{#2}]{\@Glsname@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4237 \def\@Glsname@#1#2[#3]{%
4238   \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
4239 }
```

\GLSname behaves like \glsname except that the link text is converted to uppercase.

\GLSname

```
4240 \newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
```

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

```
4241 \newcommand*{\@GLSname}[2][\%
4242   \new@ifnextchar[\@GLSname@{#1}{#2}]{\@GLSname@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4243 \def\@GLSname@#1#2[#3]{%
4244   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
4245 }
```

\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

```
4246 \newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
```

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

```
4247 \newcommand*{\@glsdesc}[2][\%
4248   \new@ifnextchar[\@glsdesc@{#1}{#2}]{\@glsdesc@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4249 \def\@glsdesc@#1#2[#3]{%
4250   \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
4251 }
```

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

\Glsdesc

```
4252 \newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
```

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

```
4253 \newcommand*{\@GLSdesc}[2] [] {%  
4254   \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4255 \def\@GLSdesc@#1#2[#3] {%  
4256   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%  
4257 }
```

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

\GLSdesc

```
4258 \newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
```

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

```
4259 \newcommand*{\@GLSdesc}[2] [] {%  
4260   \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4261 \def\@GLSdesc@#1#2[#3] {%  
4262   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%  
4263 }
```

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

\glsdescplural

```
4264 \newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
```

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

```
4265 \newcommand*{\@glsdescplural}[2] [] {%  
4266   \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4267 \def\@glsdescplural@#1#2[#3] {%  
4268   \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%  
4269 }
```

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

\Glsdescplural

```
4270 \newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
```

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

```
4271 \newcommand*{\@Glsdescplural}[2] [] {%  
4272   \new@ifnextchar [{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4273 \def\@Glsdescplural@#1#2[#3] {%  
4274   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%  
4275 }
```

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

`\GLSdescplural`

```
4276 \newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
```

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

```
4277 \newcommand*{\@GLSdescplural}[2] [] {%
```

```
4278   \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4279 \def\@GLSdescplural@#1#2[#3] {%
```

```
4280   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
```

```
4281 }
```

`\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`

```
4282 \newrobustcmd*{\glsymbol}{\@gls@hyp@opt\@glsymbol}
```

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

```
4283 \newcommand*{\@glsymbol}[2] [] {%
```

```
4284   \new@ifnextchar[{\@glsymbol@{#1}{#2}}{\@glsymbol@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4285 \def\@glsymbol@#1#2[#3] {%
```

```
4286   \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}}%
```

```
4287 }
```

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

`\Glsymbol`

```
4288 \newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
```

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

```
4289 \newcommand*{\@Glsymbol}[2] [] {%
```

```
4290   \new@ifnextchar[{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4291 \def\@Glsymbol@#1#2[#3] {%
```

```
4292   \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}}%
```

```
4293 }
```

`\GLSsymbol` behaves like `\glsymbol` except that the link text is converted to uppercase.

`\GLSsymbol`

```
4294 \newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
```

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

```
4295 \newcommand*{\@GLSsymbol}[2] [] {%
```

```
4296   \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4297 \def\@GLSsymbol@#1#2[#3]{%
4298   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbol{#2}#3}}%
4299 }
```

`\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`

```
4300 \newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
```

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

```
4301 \newcommand*{\@glsymbolplural}[2][ ]{%
4302   \new@ifnextchar[{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
4303 \def\@glsymbolplural@#1#2[#3]{%
4304   \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
4305 }
```

`\Glsymbolplural` behaves like `\glsymbolplural` except that the first letter is converted to uppercase.

`Glsymbolplural`

```
4306 \newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
```

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

```
4307 \newcommand*{\@Glsymbolplural}[2][ ]{%
4308   \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
4309 \def\@Glsymbolplural@#1#2[#3]{%
4310   \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
4311 }
```

`\GLSsymbolplural` behaves like `\glsymbolplural` except that the link text is converted to uppercase.

`GLSsymbolplural`

```
4312 \newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
```

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

```
4313 \newcommand*{\@GLSsymbolplural}[2][ ]{%
4314   \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
4315 \def\@GLSsymbolplural@#1#2[#3]{%
4316   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}#3}}%
4317 }
```

`\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`

```
4318 \newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
```

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

```
4319 \newcommand*{\@glsuseri}[2] [] {%
```

```
4320   \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
```

Read in the final optional argument:

```
4321 \def\@glsuseri@#1#2[#3]{%
```

```
4322   \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
```

```
4323 }
```

`\Glsuseri` behaves like `\glsuseri` except that the first letter is converted to uppercase.

`\Glsuseri`

```
4324 \newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
```

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

```
4325 \newcommand*{\@Glsuseri}[2] [] {%
```

```
4326   \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
```

Read in the final optional argument:

```
4327 \def\@Glsuseri@#1#2[#3]{%
```

```
4328   \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
```

```
4329 }
```

`\GLSuseri` behaves like `\glsuseri` except that the link text is converted to uppercase.

`\GLSuseri`

```
4330 \newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
```

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

```
4331 \newcommand*{\@GLSuseri}[2] [] {%
```

```
4332   \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
```

Read in the final optional argument:

```
4333 \def\@GLSuseri@#1#2[#3]{%
```

```
4334   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}%
```

```
4335 }
```

`\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`

```
4336 \newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
```

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

```
4337 \newcommand*{\@glsuserii}[2] [] {%
```

```
4338   \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}]
```

Read in the final optional argument:

```
4339 \def\@glsuserii@#1#2[#3]{%
```

```
4340   \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
```

```
4341 }
```

\Glsuserii behaves like \glsuserii except that the first letter is converted to uppercase.

\Glsuserii

```
4342 \newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
```

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

```
4343 \newcommand*{\@Glsuserii}[2] [] {%
```

```
4344   \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4345 \def\@Glsuserii@#1#2[#3] {%
```

```
4346   \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
```

```
4347 }
```

\GLSuserii behaves like \glsuserii except that the link text is converted to uppercase.

\GLSuserii

```
4348 \newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
```

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

```
4349 \newcommand*{\@GLSuserii}[2] [] {%
```

```
4350   \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4351 \def\@GLSuserii@#1#2[#3] {%
```

```
4352   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserii{#2}#3}}%
```

```
4353 }
```

\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

```
4354 \newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
```

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

```
4355 \newcommand*{\@glsuseriii}[2] [] {%
```

```
4356   \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4357 \def\@glsuseriii@#1#2[#3] {%
```

```
4358   \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
```

```
4359 }
```

\Glsuseriii behaves like \glsuseriii except that the first letter is converted to uppercase.

\Glsuseriii

```
4360 \newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
```

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

```
4361 \newcommand*{\@Glsuseriii}[2] [] {%
```

```
4362   \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4363 \def\@Glsuseriii@#1#2[#3]{%
4364   \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
4365 }
```

\Glsuseriii behaves like \glsuseriii except that the link text is converted to uppercase.

\Glsuseriii

```
4366 \newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
```

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

```
4367 \newcommand*{\@Glsuseriii}[2][\@Glsuseriii@#1]{%
4368   \new@ifnextchar[\@Glsuseriii@{#1}{#2}]{\@Glsuseriii@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4369 \def\@Glsuseriii@#1#2[#3]{%
4370   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
4371 }
```

\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

```
4372 \newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
```

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

```
4373 \newcommand*{\@glsuseriv}[2][\@glsuseriv@#1]{%
4374   \new@ifnextchar[\@glsuseriv@{#1}{#2}]{\@glsuseriv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4375 \def\@glsuseriv@#1#2[#3]{%
4376   \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
4377 }
```

\Glsuseriv behaves like \glsuseriv except that the first letter is converted to uppercase.

\Glsuseriv

```
4378 \newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
```

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

```
4379 \newcommand*{\@Glsuseriv}[2][\@Glsuseriv@#1]{%
4380   \new@ifnextchar[\@Glsuseriv@{#1}{#2}]{\@Glsuseriv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4381 \def\@Glsuseriv@#1#2[#3]{%
4382   \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
4383 }
```

\GLSuseriv behaves like \glsuseriv except that the link text is converted to uppercase.

\GLSuseriv

```
4384 \newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
```

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

```
4385 \newcommand*{\@GLSuseriv}[2] [] {%  
4386   \new@ifnextchar [{\@GLSuseriv@{#1}{#2}}]{\@GLSuseriv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4387 \def\@GLSuseriv@#1#2[#3] {%  
4388   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}%  
4389 }
```

\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

```
4390 \newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
```

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

```
4391 \newcommand*{\@glsuserv}[2] [] {%  
4392   \new@ifnextchar [{\@glsuserv@{#1}{#2}}]{\@glsuserv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4393 \def\@glsuserv@#1#2[#3] {%  
4394   \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}}%  
4395 }
```

\Glsuserv behaves like \glsuserv except that the first letter is converted to uppercase.

\Glsuserv

```
4396 \newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
```

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

```
4397 \newcommand*{\@Glsuserv}[2] [] {%  
4398   \new@ifnextchar [{\@Glsuserv@{#1}{#2}}]{\@Glsuserv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4399 \def\@Glsuserv@#1#2[#3] {%  
4400   \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}}%  
4401 }
```

\GLSuserv behaves like \glsuserv except that the link text is converted to uppercase.

\GLSuserv

```
4402 \newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
```

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

```
4403 \newcommand*{\@GLSuserv}[2] [] {%  
4404   \new@ifnextchar [{\@GLSuserv@{#1}{#2}}]{\@GLSuserv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4405 \def\@GLSuserv@#1#2[#3] {%  
4406   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%  
4407 }
```

\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

```
4408 \newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
```

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

```
4409 \newcommand*{\@glsuservi}[2][\@glsuservi@{#1}{#2}]{\@glsuservi@{#1}{#2}[]}
```

```
4410 \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4411 \def\@glsuservi@#1#2[#3]{%
```

```
4412 \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
```

```
4413 }
```

\Glsuservi behaves like \glsuservi except that the first letter is converted to uppercase.

\Glsuservi

```
4414 \newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
```

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

```
4415 \newcommand*{\@Glsuservi}[2][\@Glsuservi@{#1}{#2}]{\@Glsuservi@{#1}{#2}[]}
```

```
4416 \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4417 \def\@Glsuservi@#1#2[#3]{%
```

```
4418 \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
```

```
4419 }
```

\GLSuservi behaves like \glsuservi except that the link text is converted to uppercase.

\GLSuservi

```
4420 \newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
```

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

```
4421 \newcommand*{\@GLSuservi}[2][\@GLSuservi@{#1}{#2}]{\@GLSuservi@{#1}{#2}[]}
```

```
4422 \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4423 \def\@GLSuservi@#1#2[#3]{%
```

```
4424 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}#3}}%
```

```
4425 }
```

Now deal with acronym related keys. First the short form:

\acrshort

```
4426 \newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
```

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

```
4427 \newcommand*{\@ns@acrshort}[2][\@ns@acrshort@{#1}{#2}]{\@ns@acrshort@{#1}{#2}[]}
```

```
4428 \new@ifnextchar[{\@ns@acrshort@{#1}{#2}}{\@ns@acrshort@{#1}{#2}[]}]
```

```
4429 }
```

Read in the final optional argument:

```
4430 \def\@ns@acrshort#1#2[#3]{%
```

```
4431 \glsdoifexists{#2}%
```

```
4432 {%
```

```

4433 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

4434 \let\gl@sifplural\@secondoftwo
4435 \let\gl@scapscase\@firstofthree
4436 \let\gl@insert\@empty
4437 \def\gl@customtext{%
4438 \acronymfont{\gl@sentryshort{#2}}#3%
4439 }%

Call \@gl@link Note that \@gl@link sets \glstype.
4440 \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4441 }%

4442 \glspostlinkhook
4443 }

```

\Acrshort

```

4444 \newrobustcmd*{\Acrshort}{\@gl@hyp@opt\@ns@Acrshort}

Define the un-starred form. Need to determine if there is a final optional argument
4445 \newcommand*{\ns@Acrshort}[2][{}]{%
4446 \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2}[]}%
4447 }

Read in the final optional argument:
4448 \def\@Acrshort#1#2[#3]{%
4449 \gl@doifexists{#2}%
4450 {%
4451 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

4452 \def\gl@label{#2}%
4453 \let\gl@sifplural\@secondoftwo
4454 \let\gl@scapscase\@secondofthree
4455 \let\gl@insert\@empty
4456 \def\gl@customtext{%
4457 \acronymfont{\gl@sentryshort{#2}}#3%
4458 }%

Call \@gl@link Note that \@gl@link sets \glstype.
4459 \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4460 }%

4461 \glspostlinkhook
4462 }

```

\ACRshort

```

4463 \newrobustcmd*{\ACRshort}{\@gl@hyp@opt\@ns@ACRshort}

```

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

```
4464 \newcommand*{\ns@ACRshort}[2][\%
4465   \new@ifnextchar[\@ACRshort{#1}{#2}]{\@ACRshort{#1}{#2}[]}%
4466 }
```

Read in the final optional argument:

```
4467 \def\@ACRshort#1#2[#3]{%
4468   \glsdoifexists{#2}%
4469   {%
4470     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
4471     \def\glslabel{#2}%
4472     \let\glsifplural\@secondoftwo
4473     \let\glsapscase\@thirdofthree
4474     \let\glsinsert\@empty
4475     \def\glscustomtext{%
4476       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
4477     }%
```

Call \@gls@link Note that \@gls@link sets \glsstyle.

```
4478   \@gls@link[#1]{#2}{\csname gls\@glsstyle @entryfmt\endcsname}%
4479   }%
4480   \glspostlinkhook
4481 }
```

Short plural:

\acrshortpl

```
4482 \newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
```

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

```
4483 \newcommand*{\ns@acrshortpl}[2][\%
4484   \new@ifnextchar[\@acrshortpl{#1}{#2}]{\@acrshortpl{#1}{#2}[]}%
4485 }
```

Read in the final optional argument:

```
4486 \def\@acrshortpl#1#2[#3]{%
4487   \glsdoifexists{#2}%
4488   {%
4489     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
4490     \def\glslabel{#2}%
4491     \let\glsifplural\@firstoftwo
4492     \let\glsapscase\@firstofthree
4493     \let\glsinsert\@empty
4494     \def\glscustomtext{%
4495       \acronymfont{\glsentryshortpl{#2}}#3%
4496     }%
```

Call \@gls@link Note that \@gls@link sets \glstype.

```
4497   \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4498   }%

4499   \glspostlinkhook
4500 }
```

\Acrshortpl

```
4501 \newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
```

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

```
4502 \newcommand*{\ns@Acrshortpl}[2] [] {%
4503   \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
4504 }
```

Read in the final optional argument:

```
4505 \def\@Acrshortpl#1#2[#3]{%
4506   \glsdoidexists{#2}%
4507   {%

4508     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

4509     \def\glslabel{#2}%
4510     \let\glsifplural\@firstoftwo
4511     \let\glscapscase\@secondofthree
4512     \let\glsinsert\@empty
4513     \def\glscustomtext{%
4514       \acronymfont{\Glsentryshortpl{#2}}#3%
4515     }%
```

Call \@gls@link Note that \@gls@link sets \glstype.

```
4516   \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4517   }%

4518   \glspostlinkhook
4519 }
```

\ACRshortpl

```
4520 \newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
```

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

```
4521 \newcommand*{\ns@ACRshortpl}[2] [] {%
4522   \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
4523 }
```

Read in the final optional argument:

```
4524 \def\@ACRshortpl#1#2[#3]{%
4525   \glsdoidexists{#2}%
4526   {%

4527     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```



```

4528 \def\glslabel{#2}%
4529 \let\glsifplural\@firstoftwo
4530 \let\glsifscapscase\@thirdofthree
4531 \let\glsinsert\@empty
4532 \def\glscustomtext{%
4533     \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
4534 }%

Call \@gls@link Note that \@gls@link sets \glstype.
4535 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4536 }%

4537 \glspostlinkhook
4538 }

```

\acrlong

```

4539 \newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}

```

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

```

4540 \newcommand*{\ns@acrlong}[2][{}]{%
4541 \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
4542 }

```

Read in the final optional argument:

```

4543 \def\@acrlong#1#2[#3]{%
4544 \glsdoifexists{#2}%
4545 {%
4546 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

4547 \def\glslabel{#2}%
4548 \let\glsifplural\@secondoftwo
4549 \let\glsifscapscase\@firstofthree
4550 \let\glsinsert\@empty

```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```

4551 \def\glscustomtext{%
4552     \glsentrylong{#2}#3%
4553 }%

```

Call \@gls@link Note that \@gls@link sets \glstype.

```

4554 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4555 }%

4556 \glspostlinkhook
4557 }

```

\Acrlong

```

4558 \newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}

```

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

```
4559 \newcommand*{\ns@Acrlong}[2][{}]{%
4560   \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
4561 }
```

Read in the final optional argument:

```
4562 \def\@Acrlong#1#2[#3]{%
4563   \glsdoifexists{#2}%
4564   {%
4565     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4566     \def\glslabel{#2}%
4567     \let\glsifplural\@secondoftwo
4568     \let\glscapscase\@secondofthree
4569     \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4570   \def\glscustomtext{%
4571     \Glsentrylong{#2}#3%
4572   }%
```

Call \@gl@link. Note that \@gl@link sets \glstype.

```
4573   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4574   }%
4575   \glspostlinkhook
4576 }
```

\ACRlong

```
4577 \newrobustcmd*{\ACRlong}{\@gl@hyp@opt\ns@ACRlong}
```

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

```
4578 \newcommand*{\ns@ACRlong}[2][{}]{%
4579   \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
4580 }
```

Read in the final optional argument:

```
4581 \def\@ACRlong#1#2[#3]{%
4582   \glsdoifexists{#2}%
4583   {%
4584     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4585     \def\glslabel{#2}%
4586     \let\glsifplural\@secondoftwo
4587     \let\glscapscase\@thirdofthree
4588     \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4589 \def\glscustomtext{%
4590 \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
4591 }%
```

Call \@gls@link. Note that \@gls@link sets \glstype.

```
4592 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4593 }%

4594 \glspostlinkhook
4595 }
```

Short plural:

\acrlongpl

```
4596 \newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\@ns@acrlongpl}
```

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

```
4597 \newcommand*{\ns@acrlongpl}[2][{}]{%
4598 \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
4599 }
```

Read in the final optional argument:

```
4600 \def\@acrlongpl#1#2[#3]{%
4601 \glsdoidexists{#2}%
4602 {%

4603 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

4604 \def\glslabel{#2}%
4605 \let\glsifplural\@firstoftwo
4606 \let\glscapscase\@firstofthree
4607 \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4608 \def\glscustomtext{%
4609 \glsentrylongpl{#2}#3%
4610 }%
```

Call \@gls@link. Note that \@gls@link sets \glstype.

```
4611 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4612 }%

4613 \glspostlinkhook
4614 }
```

\Acrlongpl

```
4615 \newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
```

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

```
4616 \newcommand*{\ns@Acrlongpl}[2][\%
4617   \new@ifnextchar[\@Acrlongpl{#1}{#2}]{\@Acrlongpl{#1}{#2}[]}%
4618 }
```

Read in the final optional argument:

```
4619 \def\@Acrlongpl#1#2[#3]{%
4620   \glsdoifexists{#2}%
4621   {%
4622     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4623     \def\glslabel{#2}%
4624     \let\glsifplural\@firstoftwo
4625     \let\glscapscase\@secondofthree
4626     \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4627   \def\glscustomtext{%
4628     \Glsentrylongpl{#2}#3%
4629   }%
```

Call \@gl@link. Note that \@gl@link sets \glstype.

```
4630   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4631   }%
4632   \glspostlinkhook
4633 }
```

\ACRlongpl

```
4634 \newrobustcmd*{\ACRlongpl}{\@gl@hyp@opt\ns@ACRlongpl}
```

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

```
4635 \newcommand*{\ns@ACRlongpl}[2][\%
4636   \new@ifnextchar[\@ACRlongpl{#1}{#2}]{\@ACRlongpl{#1}{#2}[]}%
4637 }
```

Read in the final optional argument:

```
4638 \def\@ACRlongpl#1#2[#3]{%
4639   \glsdoifexists{#2}%
4640   {%
4641     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4642     \def\glslabel{#2}%
4643     \let\glsifplural\@firstoftwo
4644     \let\glscapscase\@thirdofthree
4645     \let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```

4646 \def\glscustomtext{%
4647 \mfirstucMakeUppercase{\glsenrylongpl{#2}#3}%
4648 }%

Call \@gls@link. Note that \@gls@link sets \glstype.
4649 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4650 }%

4651 \glspostlinkhook
4652 }

```

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>}
```

```

4653 \newcommand*{\@gls@entry@field}[2]{%
4654 \csname glo@\glsdetoklabel{#1}@\#2\endcsname
4655 }

```

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

```

4656 \newcommand*{\glsletentryfield}[3]{%
4657 \letcs{#1}{glo@\glsdetoklabel{#2}@\#3}%
4658 }

```

`\Gls@entry@field` Generic first letter uppercase version.

```
\@Gls@entry@field{<label>}{<field>}
```

```

4659 \newcommand*{\@Gls@entry@field}[2]{%
4660 \glsdoifexistsordo{#1}%
4661 {%
4662 \letcs{@glo@text}{glo@\glsdetoklabel{#1}@\#2}%
4663 \ifdef{@glo@text
4664 {%
4665 \xmakefirstuc{@glo@text}%
4666 }%
4667 }%
4668 ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
4669 entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry

```

```

4670      label and the field name}%
4671    }%
4672  }%
4673  {%
4674    ??%
4675  }%
4676 }

```

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`

```

4677 \newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}

```

`\Glsentryname`

```

4678 \newrobustcmd*{\Glsentryname}[1]{%
4679   \@Gls@entryname{#1}%
4680 }

```

`\@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. First the default behaviour:

```

4681 \newcommand*{\@Gls@entryname}[1]{%
4682   \@Gls@entry@field{#1}{name}%
4683 }

```

`\ls@acentryname` Now the behaviour when `\setacronymstyle` is used:

```

4684 \newcommand*{\@Gls@acentryname}[1]{%
4685   \ifglshaslong{#1}%
4686   {%
4687     \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
4688     \@gls@getbody is defined by mfirstuc (which used to be part of glossaries).
4689     \expandafter\@gls@getbody\@glo@text{}\@nil
4690     \expandafter\ifx\@gls@body\glsentrylong\relax
4691       \expandafter\Glsentrylong\@gls@rest
4692     \else
4693       \expandafter\ifx\@gls@body\glsentryshort\relax
4694         \expandafter\Glsentryshort\@gls@rest
4695       \else
4696         \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.

```

4696     {%
4697       \let\glsentryshort\Glsentryshort
4698       \@glo@text

```

```

4699      }%
4700      \else

4701          \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
4702          {%
4703              \let\glsentryshort\Glsentryshort
4704              \@glo@text
4705          }%
4706          \else
4707              \xmakefirstuc{\@glo@text}%
4708          \fi
4709      \fi
4710  \fi
4711 \fi
4712 }%
4713 {%

```

Not an acronym

```

4714      \@Gls@entry@field{#1}{name}%
4715  }%
4716 }

```

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`

```

4717 \newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}

```

`\Glsentrydesc`

```

4718 \newrobustcmd*{\Glsentrydesc}[1]{%
4719   \@Gls@entry@field{#1}{desc}}%
4720 }

```

Plural form:

`entrydescplural`

```

4721 \newcommand*{\glsentrydescplural}[1]{%
4722   \@gls@entry@field{#1}{descplural}}%
4723 }

```

`entrydescplural`

```

4724 \newrobustcmd*{\Glsentrydescplural}[1]{%
4725   \@Gls@entry@field{#1}{descplural}}%
4726 }

```

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`

```
4727 \newcommand*{\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
```

`\Glsentrytext`

```
4728 \newrobustcmd*{\Glsentrytext}[1]{%
4729   \@Gls@entry@field{#1}{text}%
4730 }
```

Get the plural form:

`\glsentryplural`

```
4731 \newcommand*{\glsentryplural}[1]{%
4732   \@gls@entry@field{#1}{plural}%
4733 }
```

`\Glsentryplural`

```
4734 \newrobustcmd*{\Glsentryplural}[1]{%
4735   \@Gls@entry@field{#1}{plural}%
4736 }
```

Get the symbol associated with this entry. The argument is the label associated with the entry.

`\glsentrysymbol`

```
4737 \newcommand*{\glsentrysymbol}[1]{%
4738   \@gls@entry@field{#1}{symbol}%
4739 }
```

`\Glsentrysymbol`

```
4740 \newrobustcmd*{\Glsentrysymbol}[1]{%
4741   \@Gls@entry@field{#1}{symbol}%
4742 }
```

Plural form:

`\glsentrysymbolplural`

```
4743 \newcommand*{\glsentrysymbolplural}[1]{%
4744   \@gls@entry@field{#1}{symbolplural}%
4745 }
```

`\Glsentrysymbolplural`

```
4746 \newrobustcmd*{\Glsentrysymbolplural}[1]{%
4747   \@Gls@entry@field{#1}{symbolplural}%
4748 }
```

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`

```
4749 \newcommand*{\glsentryfirst}[1]{%
4750   \@gls@entry@field{#1}{first}%
4751 }
```

`\Glsentryfirst`

```
4752 \newrobustcmd*{\Glsentryfirst}[1]{%
4753   \@Gls@entry@field{#1}{first}%
4754 }
```

Get the plural form (as specified by the `firstplural` key when the entry was defined).

`\glsentryfirstplural`

```
4755 \newcommand*{\glsentryfirstplural}[1]{%
4756   \@gls@entry@field{#1}{firstpl}%
4757 }
```

`\Glsentryfirstplural`

```
4758 \newrobustcmd*{\Glsentryfirstplural}[1]{%
4759   \@Gls@entry@field{#1}{firstpl}%
4760 }
```

`\glsentrytitlecase`

```
4761 \newrobustcmd*{\@glsentrytitlecase}[2]{%
4762   \glsdoifexists{#1}%
4763   {%
4764     \glsfieldfetch{#1}{#2}{\@gls@value}%
4765     \xcapitalisewords{\@gls@value}%
4766   }%
4767 }
4768 \ifdef\texorpdfstring
4769 {
4770   \newcommand*{\glsentrytitlecase}[2]{%
4771     \texorpdfstring
4772       {\@glsentrytitlecase{#1}{#2}}%
4773       {\@gls@entry@field{#1}{#2}}%
4774   }
4775 }
4776 {
4777   \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
4778 }
```

Display the glossary type with which this entry is associated (as specified by the `type` key used when the entry was defined)

`\glsentrytype`

```
4779 \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.

```

\glsentrysort
4780 \newcommand*{\glsentrysort}[1]{%
4781   \@gls@entry@field{#1}{sort}%
4782 }

\glsentryparent  Expands to the label of the entry's parent.
4783 \newcommand*{\glsentryparent}[1]{%
4784   \@gls@entry@field{#1}{parent}%
4785 }

\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.
4786 \newcommand*{\glsentryuseri}[1]{%
4787   \@gls@entry@field{#1}{useri}%
4788 }

\Glsentryuseri
4789 \newrobustcmd*{\Glsentryuseri}[1]{%
4790   \@Gls@entry@field{#1}{useri}%
4791 }

\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.
4792 \newcommand*{\glsentryuserii}[1]{%
4793   \@gls@entry@field{#1}{userii}%
4794 }

\Glsentryuserii
4795 \newrobustcmd*{\Glsentryuserii}[1]{%
4796   \@Gls@entry@field{#1}{userii}%
4797 }

\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.
4798 \newcommand*{\glsentryuseriii}[1]{%
4799   \@gls@entry@field{#1}{useriii}%
4800 }

\Glsentryuseriii
4801 \newrobustcmd*{\Glsentryuseriii}[1]{%
4802   \@Gls@entry@field{#1}{useriii}%
4803 }

\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.
4804 \newcommand*{\glsentryuseriv}[1]{%
4805   \@gls@entry@field{#1}{useriv}%
4806 }

```

```

\Glsentryuseriv
4807 \newrobustcmd*{\Glsentryuseriv}[1]{%
4808   \@Gls@entry@field{#1}{useriv}%
4809 }

\glsentryuserv  Get the fifth user key (as specified by the user5 when the entry was defined). The argument is
                  the label associated with the entry.
4810 \newcommand*{\glsentryuserv}[1]{%
4811   \@Gls@entry@field{#1}{userv}%
4812 }

\Glsentryuserv
4813 \newrobustcmd*{\Glsentryuserv}[1]{%
4814   \@Gls@entry@field{#1}{userv}%
4815 }

\glsentryuservi  Get the sixth user key (as specified by the user6 when the entry was defined). The argument
                  is the label associated with the entry.
4816 \newcommand*{\glsentryuservi}[1]{%
4817   \@Gls@entry@field{#1}{uservi}%
4818 }

\Glsentryuservi
4819 \newrobustcmd*{\Glsentryuservi}[1]{%
4820   \@Gls@entry@field{#1}{uservi}%
4821 }

\glsentryshort  Get the short key (as specified by the short the entry was defined). The argument is the label
                  associated with the entry.
4822 \newcommand*{\glsentryshort}[1]{\@Gls@entry@field{#1}{short}}

\Glsentryshort
4823 \newrobustcmd*{\Glsentryshort}[1]{%
4824   \@Gls@entry@field{#1}{short}%
4825 }

glsentryshortpl  Get the short plural key (as specified by the shortplural the entry was defined). The argument
                  is the label associated with the entry.
4826 \newcommand*{\glsentryshortpl}[1]{\@Gls@entry@field{#1}{shortpl}}

Glsentryshortpl
4827 \newrobustcmd*{\Glsentryshortpl}[1]{%
4828   \@Gls@entry@field{#1}{shortpl}%
4829 }

\glsentrylong  Get the long key (as specified by the long the entry was defined). The argument is the label
                  associated with the entry.
4830 \newcommand*{\glsentrylong}[1]{\@Gls@entry@field{#1}{long}}

```

```

\Glsentrylong
4831 \newrobustcmd*{\Glsentrylong}[1]{%
4832   \@Gls@entry@field{#1}{long}%
4833 }

\glsentrylongpl  Get the long plural key (as specified by the longplural the entry was defined). The argument is
                  the label associated with the entry.
4834 \newcommand*{\glsentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}

\Glsentrylongpl
4835 \newrobustcmd*{\Glsentrylongpl}[1]{%
4836   \@Gls@entry@field{#1}{longpl}%
4837 }

Short cut macros to access full form:

\glsentryfull
4838 \newcommand*{\glsentryfull}[1]{%
4839   \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
4840 }

\Glsentryfull
4841 \newrobustcmd*{\Glsentryfull}[1]{%
4842   \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
4843 }

\glsentryfullpl
4844 \newcommand*{\glsentryfullpl}[1]{%
4845   \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
4846 }

\Glsentryfullpl
4847 \newrobustcmd*{\Glsentryfullpl}[1]{%
4848   \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
4849 }

entrynumberlist  Displays the number list as is.
4850 \newcommand*{\glsentrynumberlist}[1]{%
4851   \glsdoifexists{#1}%
4852   {%
4853     \@Gls@entry@field{#1}{numberlist}%
4854   }%
4855 }

splaynumberlist  Formats the number list for the given entry label. Doesn't work with hyperref.
4856 \@ifpackageloaded{hyperref} {%
4857   \newcommand*{\glsdisplaynumberlist}[1]{%
4858     \GlossariesWarning

```

```

4859   {%
4860     \string\glsdisplaynumberlist\space
4861     doesn't work with hyperref.^^JUsing
4862     \string\glsentrynumberlist\space instead%
4863   }%
4864   \glsentrynumberlist{#1}%
4865 }%
4866 }%
4867 {%
4868   \newcommand*{\glsdisplaynumberlist}[1]{%
4869     \glsdoifexists{#1}%
4870     {%
4871       \bgroup

4872       \protected@edef\@glo@label{\glsdetoklabel{#1}}%
4873       \let\@org@glssnumberformat\glssnumberformat
4874       \def\glssnumberformat##1{##1}%
4875       \protected@edef\the@numberlist{%
4876         \csname glo@\@glo@label @numberlist\endcsname}%
4877       \def\@gls@numlist@sep{%
4878       \def\@gls@numlist@nextsep{%
4879       \def\@gls@numlist@lastsep{%
4880       \def\@gls@thislist{%
4881       \def\@gls@donext@def{%
4882       \renewcommand\do[1]{%
4883         \protected@edef\@gls@thislist{%
4884           \@gls@thislist
4885           \noexpand\@gls@numlist@sep
4886           ##1%
4887         }%
4888         \let\@gls@numlist@sep\@gls@numlist@nextsep
4889         \def\@gls@numlist@nextsep{\glssnumlistsep}%
4890         \@gls@donext@def
4891         \def\@gls@donext@def{%
4892           \def\@gls@numlist@lastsep{\glssnumlistlastsep}%
4893         }%
4894       }%
4895       \expandafter \glssnumlistparser \expandafter{\the@numberlist}%
4896       \let\@gls@numlist@sep\@gls@numlist@lastsep
4897       \@gls@thislist
4898     \egroup
4899   }%
4900 }
4901 }

```

\glssnumlistsep

```

4902 \newcommand*{\glssnumlistsep}{, }

```

\glssnumlistlastsep

```

4903 \newcommand*{\glssnumlistlastsep}{ \& }

```

`\glshyperlink` Provide a hyperlink to a glossary entry without adding information to the glossary file. The entry needs to be added using a command like `\glslink` or `\glsadd` to ensure that the target is defined. The first (optional) argument specifies the link text. The entry name is used by default. The second argument is the entry label.

```
4904 \newcommand*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
4905   \def\@glo@label{#2}%
4906   \@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
```

1.12 Adding an entry to the glossary without generating text

The following keys are provided for `\glsadd` and `\glsaddall`:

```
4907 \define@key{glossadd}{counter}{\def\@gls@counter{#1}}
4908 \define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
```

This key is only used by `\glsaddall`:

```
4909 \define@key{glossadd}{types}{\def\@glo@type{#1}}
```

`\glsadd[<options>]{<label>}`

Add a term to the glossary without generating any link text. The optional argument indicates which counter to use, and how to format it (using a key-value list) the second argument is the entry label. Note that *<options>* only has two keys: counter and format (the types key will be ignored).

`\glsadd`

```
4910 \newrobustcmd*{\glsadd}[2][]{%
```

Need to move to horizontal mode if not already in it, but only if not in preamble.

```
4911   \@gls@adjustmode
4912   \glsdoifexists{#2}%
4913   {%
4914     \def\@glsnumberformat{glsnumberformat}%
4915     \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
4916     \setkeys{glossadd}{#1}%
```

Store the entry's counter in `\theglsentrycounter`

```
4917   \@gls@saveentrycounter
```

Define sort key if necessary:

```
4918   \@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.

```
4919   \@do@wrglossary{#2}%
4920   }%
4921 }
```

@gls@adjustmode

```
4922 \newcommand*{\@gls@adjustmode}{}
4923 \AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
```

\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

```
4924 \newrobustcmd*{\glsaddall}[1] [] {%
4925   \protected@edef\@glo@type{\@glo@types}%
4926   \setkeys{glossadd}{#1}%
4927   \forallglsentries[\@glo@type]{\@glo@entry}{%
4928     \glsadd[#1]{\@glo@entry}%
4929   }%
4930 }
```

glsaddallunused

\glsaddallunused[*<glossary type>*]

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.

```
4931 \newrobustcmd*{\glsaddallunused}[1] [\@glo@types] {%
4932   \forallglsentries[#1]{\@glo@entry}%
4933   {%
4934     \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
4935   }%
4936 }
```

\glsignore

```
4937 \newcommand*{\glsignore}[1] {}
```

1.13 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 glssymbols and glsnumbers, the group titles can be translated (so that \glssymbolsgroupname replaces glssymbols and \glsnumbersgroupname replaces glsnumbers) using the command \glsetgrouptitle which is defined in . This is done to prevent any problem characters in \glssymbolsgroupname and \glsnumbersgroupname from breaking hyperlinks.

```
\glsopenbrace  Define \glsopenbrace to make it easier to write an opening brace to a file.
4938 \edef\glsopenbrace{\expandafter\@gobble\string\{}}

\glsclosebrace  Define \glsclosebrace to make it easier to write an opening brace to a file.
4939 \edef\glsclosebrace{\expandafter\@gobble\string\}}

\glsbackslash  Define \glsbackslash to make it easier to write a backslash to a file.
4940 \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.
4941 \edef\glsquote#1{\string"#1\string"}

\glpercentchar  Define \glpercentchar to make it easier to write a percent character to a file.
4942 \edef\glpercentchar{\expandafter\@gobble\string\%}

\glstildechar  Define \glstildechar to make it easier to write a tilde character to a file.
4943 \edef\glstildechar{\string~}

@glfirstletter  Define the first letter to come after the digits 0,...,9. Only required for xindy.
4944 \ifglxindy
4945   \newcommand*{\@glfirstletter}{A}
4946 \fi

tterAfterDigits  Sets the first letter to come after the digits 0,...,9. The starred version sanitizes.
4947 \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
4948   \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
4949 \ifglxindy
4950   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
4951     \renewcommand*{\@glfirstletter}{#1}}
4952   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
4953     \renewcommand*{\@glfirstletter}{#1}%
4954     \@onelevel@sanitize\@glfirstletter
4955   }
4956 \else
4957   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
4958     \glsnxindywarning\GlsSetXdyFirstLetterAfterDigits}
4959   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
4960     \@GlsSetXdyFirstLetterAfterDigits
4961   }
4962 \fi
```


numbergrouporder Specifies the order of the number group.

```
4963 \ifglxindy
4964 \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
4965 \fi
```

numberGroupOrder Sets the relative location of the number group. The starred version sanitizes.

```
4966 \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
4967 \ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
4968 }
4969 \ifglxindy
4970 \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
4971 \renewcommand*{\@xdynumbergrouporder}{#1}%
4972 }
4973 \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
4974 \renewcommand*{\@xdynumbergrouporder}{#1}%
4975 \@onelevel@sanitize\@xdynumbergrouporder
4976 }
4977 \else
4978 \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
4979 \glsnoxywarning\GlsSetXdyNumberGroupOrder}
4980 \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
4981 \@GlsSetXdyNumberGroupOrder}
4982 \fi
```

\@glsminrange Define the minimum number of successive location references to merge into a range.

```
4983 \newcommand*{\@glsminrange}{2}
```

xyMinRangeLength 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.

```
4984 \ifglxindy
4985 \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
4986 \renewcommand*{\@glsminrange}{#1}}
4987 \else
4988 \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
4989 \glsnoxywarning\GlsSetXdyMinRangeLength}
4990 \fi
```

\writeist

```
4991 \ifglxindy
    Code to use if xindy is required.
4992 \def\writeist{%
    Define write register if not already defined
4993 \ifundef{\glswrite}{\newwrite\glswrite}{}%
    Update attributes list
4994 \@gls@addpredefinedattributes
```

Open the file.

```
4995 \openout\glswrite=\istfilename
```

Write header comment at the start of the file

```
4996 \write\glswrite{;; xindy style file created by the glossaries
4997   package}%
4998 \write\glswrite{;; for document '\jobname' on
4999   \the\year-\the\month-\the\day}%
```

Specify the required styles

```
5000 \write\glswrite{^^J; required styles^^J}
5001 \@for\@xdystyle:=\@xdyrequiredstyles\do{%
5002   \ifx\@xdystyle\@empty
5003   \else
5004     \protected@write\glswrite{{(require
5005       \string"\@xdystyle.xdy\string")}}%
5006   \fi
5007 }%
```

List the allowed attributes (possible values used by the format key)

```
5008 \write\glswrite{^^J%
5009   ; list of allowed attributes (number formats)^^J}%
5010 \write\glswrite{(define-attributes ((\@xdyattributes)))}%
```

Define any additional alphabets

```
5011 \write\glswrite{^^J; user defined alphabets^^J}%
5012 \write\glswrite{\@xdyuseralphabets}%
```

Define location classes.

```
5013 \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.

```
5014 \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
```

Case where $\langle Hprefix \rangle$ is empty:

```
5015 \protected@write\glswrite{{(define-location-class
5016   \string"\@gls@classI\string"^^J\space\space\space
5017   (
5018     :sep "{}{"
5019     \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
5020     :sep "}"
5021   )
5022   ^^J\space\space\space
5023   :min-range-length \@glsminrange^^J%
5024   )
5025 }%
```

Nested iteration over all classes:

```
5026 {%
5027   \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
5028     \protected@write\glswrite{{(define-location-class
```

```

5029         \string"\@gls@classII-\@gls@classI\string"
5030         ^^J\space\space\space
5031     (
5032         :sep "{"
5033         \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
5034         :sep "{{"
5035         \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
5036         :sep "}"
5037     )
5038     ^^J\space\space\space
5039     :min-range-length \@glsminrange^^J%
5040 )
5041 }%
5042 }%
5043 }%
5044 }%

```

User defined location classes (needs checking for new location format).

```

5045 \write\glswrite{^^J; user defined location classes}%
5046 \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.)

```

5047 \write\glswrite{^^J; define cross-reference class^^J}%
5048 \write\glswrite{(define-crossref-class \string"see\string"
5049                 :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.)

```

5050 \write\glswrite{(markup-crossref-list
5051                 :class \string"see\string"^^J\space\space\space
5052                 :open \string"\string\glsseeformat\string"
5053                 :close \string"{}\string")}%

```

Provide hook to write extra material here (used by `glossaries-extra` to define a `seealso` class).

```

5054 \@xdycrossrefhook

```

List the order to sort the classes.

```

5055 \write\glswrite{^^J; define the order of the location classes}%
5056 \write\glswrite{(define-location-class-order
5057                 (\@xdylocationclassorder))}%

```

Specify what to write to the start and end of the glossary file.

```

5058 \write\glswrite{^^J; define the glossary markup^^J}%

5059 \write\glswrite{(markup-index^^J\space\space\space
5060                 :open \string"\string
5061                 \glossarysection[\string\glossarytoctitle]{\string
5062                 \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)

```

5063 \@for\@this@ctr:=\@xdycounters\do{%
5064   {%
5065     \@for\@this@attr:=\@xdyattributelist\do{%
5066       \protected@write\glswrite{}\string\providecommand*%
5067       \expandafter\string
5068       \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
5069       {%
5070         \string\setentrycounter
5071         [\expandafter@gobble\string\#1]{\@this@ctr}%
5072         \expandafter\string
5073         \csname\@this@attr\endcsname
5074         {\expandafter@gobble\string\#2}%
5075       }%
5076     }%
5077   }%
5078 }%
5079 }%

```

Add the end part of the open tag and the rest of the markup-index information:

```

5080 \write\glswrite{%
5081   \string\begin
5082   {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
5083   \space\space:close \string"\glspercentchar\glstildechar n\string
5084   \end{theglossary}\string\glossarypostamble
5085   \glstildechar n\string" ^^J\space\space\space
5086   :tree)}}%

```

Specify what to put between letter groups

```

5087 \write\glswrite{(markup-letter-group-list
5088   :sep \string"\string\glsgroupskip\glstildechar n\string")}%

```

Specify what to put between entries

```

5089 \write\glswrite{(markup-indexentry
5090   :open \string"\string\relax \string\glsresetentrylist
5091   \glstildechar n\string")}%

```

Specify how to format entries

```

5092 \write\glswrite{(markup-locclass-list :open
5093   \string"\glsopenbrace\string\glossaryentrynumbers
5094   \glsopenbrace\string\relax\space \string"^^J\space\space\space
5095   :sep \string", \string"
5096   :close \string"\glsclosebrace\glsclosebrace\string")}%

```

Specify how to separate location numbers

```

5097 \write\glswrite{(markup-locref-list
5098   :sep \string"\string\delimN\space\string")}%

```

Specify how to indicate location ranges

```

5099 \write\glswrite{(markup-range
5100   :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.

```

5101 \@onelevel@sanitize\gls@suffixF
5102 \@onelevel@sanitize\gls@suffixFF
5103 \ifx\gls@suffixF\@empty
5104 \else
5105 \write\glswrite{(markup-range
5106 :close "\gls@suffixF" :length 1 :ignore-end)}}%
5107 \fi
5108 \ifx\gls@suffixFF\@empty
5109 \else
5110 \write\glswrite{(markup-range
5111 :close "\gls@suffixFF" :length 2 :ignore-end)}}%
5112 \fi

```

Specify how to format locations.

```

5113 \write\glswrite{^^J; define format to use for locations^^J}%
5114 \write\glswrite{\@xdylocref}%

```

Specify how to separate letter groups.

```

5115 \write\glswrite{^^J; define letter group list format^^J}%
5116 \write\glswrite{(markup-letter-group-list
5117 :sep \string"\string\glsgroupskip\glstildechar n\string")}%

```

Define letter group headings.

```

5118 \write\glswrite{^^J; letter group headings^^J}%
5119 \write\glswrite{(markup-letter-group
5120 :open-head \string"\string\glsgroupheading
5121 \glsopenbrace\string"^^J\space\space\space
5122 :close-head \string"\glsclosebrace\string")}%

```

Define additional letter groups.

```

5123 \write\glswrite{^^J; additional letter groups^^J}%
5124 \write\glswrite{\@xdylettergroups}%

```

Define additional sort rules

```

5125 \write\glswrite{^^J; additional sort rules^^J}
5126 \write\glswrite{\@xdysortrules}%

```

Hook for any additional information:

```

5127 \@gls@writeisthook

```

Close the style file

```

5128 \closeout\glswrite

```

Suppress any further calls.

```

5129 \let\writeist\relax
5130 }
5131 \else

```

Code to use if makeindex is required.

```

5132 \edef\@gls@actualchar{\string?}
5133 \edef\@gls@encapchar{\string|}
5134 \edef\@gls@levelchar{\string!}
5135 \edef\@gls@quotechar{\string"}%
5136 \let\GlsSetQuote\gls@nosetquote
5137 \def\writeist{\relax
5138   \ifundef{\glswrite}{\newwrite\glswrite}{}\relax
5139   \openout\glswrite=\istfilename
5140   \write\glswrite{\glspersentchar\space makeindex style file
5141     created by the glossaries package}
5142   \write\glswrite{\glspersentchar\space for document
5143     '\jobname' on \the\year-\the\month-\the\day}
5144   \write\glswrite{actual '\@gls@actualchar'}
5145   \write\glswrite{encap '\@gls@encapchar'}
5146   \write\glswrite{level '\@gls@levelchar'}
5147   \write\glswrite{quote '\@gls@quotechar'}
5148   \write\glswrite{keyword \string\string\glossaryentry\string}
5149   \write\glswrite{preamble \string\string\glossarysection[\string
5150     \glossarytoctitle]{\string\glossarytitle}\string
5151     \glossarypreamble\string\n\string\begin{theglossary}\string
5152     \glossaryheader\string\n\string}
5153   \write\glswrite{postamble \string\string%\string\n\string
5154     \end{theglossary}\string\glossarypostamble\string\n
5155     \string}
5156   \write\glswrite{group_skip \string\string\glsgroupskip\string\n
5157     \string}
5158   \write\glswrite{item_0 \string\string%\string\n\string}
5159   \write\glswrite{item_1 \string\string%\string\n\string}
5160   \write\glswrite{item_2 \string\string%\string\n\string}
5161   \write\glswrite{item_01 \string\string%\string\n\string}
5162   \write\glswrite{item_x1
5163     \string\string\relax \string\glsresetentrylist\string\n
5164     \string}
5165   \write\glswrite{item_12 \string\string%\string\n\string}
5166   \write\glswrite{item_x2
5167     \string\string\relax \string\glsresetentrylist\string\n
5168     \string}

5169   \write\glswrite{delim_0 \string\string\{\string
5170     \glossaryentrynumbers\string\{\string\relax \string}
5171   \write\glswrite{delim_1 \string\string\{\string
5172     \glossaryentrynumbers\string\{\string\relax \string}
5173   \write\glswrite{delim_2 \string\string\{\string
5174     \glossaryentrynumbers\string\{\string\relax \string}
5175   \write\glswrite{delim_t \string\string\}\string\}\string}
5176   \write\glswrite{delim_n \string\string\delimN \string}
5177   \write\glswrite{delim_r \string\string\delimR \string}
5178   \write\glswrite{headings_flag 1}
5179   \write\glswrite{heading_prefix

```

```

5180     \string\string\glsgroupheading\string\{\string"}
5181 \write\glswrite{heading_suffix
5182     \string\string\}\string\relax
5183     \string\glsgroupheading\string"}
5184 \write\glswrite{symhead_positive \string"glssymbols\string"}
5185 \write\glswrite{numhead_positive \string"glsgroupheading\string"}
5186 \write\glswrite{page_compositor \string"glsgroupheading\string"}
5187 \@gls@escbsdq\gls@suffixF
5188 \@gls@escbsdq\gls@suffixFF
5189 \ifx\gls@suffixF\@empty
5190 \else
5191     \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
5192 \fi
5193 \ifx\gls@suffixFF\@empty
5194 \else
5195     \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
5196 \fi

```

Hook for any additional information:

```

5197 \@gls@writeisthook

```

Close the file and disable \writeist.

```

5198 \closeout\glswrite
5199 \let\writeist\relax
5200 }
5201 \fi

```

SetWriteIstHook Allow user to append information to the style file.

```

5202 \newcommand*\GlsSetWriteIstHook[1]{\renewcommand*\@gls@writeisthook{#1}}
5203 \@onlypremake\GlsSetWriteIstHook

```

ls@writeisthook

```

5204 \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.

```

5205 \ifglxindy
5206 \newcommand*\GlsSetQuote[1]{\glsgroupheading\GlsSetQuote}
5207 \newcommand*\GlsSetQuote[1]{\glsgroupheading\GlsSetQuote}
5208 \else
5209 \newcommand*\GlsSetQuote[1]{\edef\@gls@quotechar{\string#1}}

```

If German is in use, set the extra makeindex option so makeglossaries can pick it up.

```

5210 \@ifpackageloaded{tracklang}%
5211 {%
5212     \IfTrackedLanguage{german}%
5213     {%
5214         \def\@gls@extramakeindexopts{-g}%
5215     }%
5216 }%

```

```

5217 }%
5218 {}%

Need to redefine \@gls@checkquote
5219 \edef\@gls@docheckquotedef{%
5220   \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
5221     \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
5222     \noexpand\toks@={####1}%
5223     \noexpand\ifx\noexpand\null####2\noexpand\null
5224     \noexpand\ifx\noexpand\null####3\noexpand\null
5225     \noexpand\edef\noexpand\@gls@checkedmkidx{%
5226       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
5227     \noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
5228     \noexpand\else
5229     \noexpand\edef\noexpand\@gls@checkedmkidx{%
5230       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
5231       \noexpand\@gls@quotechar\noexpand\@gls@quotechar
5232       \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
5233     \noexpand\def\noexpand\@gls@checkquote{%
5234       \noexpand\@gls@checkquote####3\noexpand\null}%
5235     \noexpand\fi
5236     \noexpand\else
5237     \noexpand\edef\noexpand\@gls@checkedmkidx{%
5238       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
5239       \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
5240     \noexpand\ifx\noexpand\null####3\noexpand\null
5241     \noexpand\def\noexpand\@gls@checkquote{%
5242       \noexpand\@gls@checkquote####2#1#1\noexpand\null}%
5243     \noexpand\else
5244     \noexpand\def\noexpand\@gls@checkquote{%
5245       \noexpand\@gls@checkquote####2#1####3\noexpand\null}%
5246     \noexpand\fi
5247     \noexpand\fi
5248     \noexpand\@gls@checkquote
5249   }%
5250 }%
5251 \@gls@docheckquotedef
5252 \edef\@gls@docheckquotedef{%
5253   \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
5254     \noexpand\def\noexpand\@gls@checkedmkidx{%
5255       \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
5256       #1#1\noexpand\null
5257       \noexpand\expandafter\noexpand\@gls@updatechecked
5258       \noexpand\@gls@checkedmkidx{####1}%
5259       \noexpand\def\noexpand\@gls@checkedmkidx{%
5260         \noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
5261         \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
5262         \noexpand\null
5263         \noexpand\expandafter\noexpand\@gls@updatechecked
5264         \noexpand\@gls@checkedmkidx{####1}%

```



```

5265 \noexpand\def\noexpand\@gls@checkedmkidx{%
5266 \noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
5267 \noexpand\?\noexpand\?\noexpand\null
5268 \noexpand\expandafter\noexpand\@gls@updatechecked
5269 \noexpand\@gls@checkedmkidx{####1}%
5270 \noexpand\def\noexpand\@gls@checkedmkidx{%
5271 \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
5272 \noexpand?\noexpand?\noexpand\null
5273 \noexpand\expandafter\noexpand\@gls@updatechecked
5274 \noexpand\@gls@checkedmkidx{####1}%
5275 \noexpand\def\noexpand\@gls@checkedmkidx{%
5276 \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
5277 \noexpand|\noexpand|\noexpand\null
5278 \noexpand\expandafter\noexpand\@gls@updatechecked
5279 \noexpand\@gls@checkedmkidx{####1}%
5280 \noexpand\def\noexpand\@gls@checkedmkidx{%
5281 \noexpand\expandafter\noexpand\@gls@checkescbar####1\noexpand\@nil
5282 \noexpand||\noexpand||\noexpand\null
5283 \noexpand\expandafter\noexpand\@gls@updatechecked
5284 \noexpand\@gls@checkedmkidx{####1}%
5285 \noexpand\def\noexpand\@gls@checkedmkidx{%
5286 \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
5287 \noexpand!\noexpand!\noexpand\null
5288 \noexpand\expandafter\noexpand\@gls@updatechecked
5289 \noexpand\@gls@checkedmkidx{####1}%

5290 \noexpand\def\noexpand\@gls@checkedmkidx{%
5291 \noexpand\expandafter\noexpand\@gls@checkesclevel####1\noexpand\@nil
5292 \noexpand\!\noexpand\!\noexpand\null
5293 \noexpand\expandafter\noexpand\@gls@updatechecked
5294 \noexpand\@gls@checkedmkidx{####1}%

5295 \noexpand\def\noexpand\@gls@checkedmkidx{%
5296 \noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
5297 \@backslashchar\@backslashchar\noexpand\null
5298 \noexpand\expandafter\noexpand\@gls@updatechecked
5299 \noexpand\@gls@checkedmkidx{####1}%
5300 }%
5301 }%
5302 \@gls@docheckquotedef
5303 \edef\@gls@docheckquotedef{%
5304 \noexpand\def\noexpand\@gls@checkescquote####1%
5305 \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
5306 ####3\noexpand\null{%
5307 \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
5308 \noexpand\toks@={####1}%
5309 \noexpand\ifx\noexpand\null####2\noexpand\null
5310 \noexpand\ifx\noexpand\null####3\noexpand\null
5311 \noexpand\edef\noexpand\@gls@checkedmkidx{%
5312 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%

```

```

5313 \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
5314 \noexpand\else
5315 \noexpand\edef\noexpand\@gls@checkedmkidx{%
5316 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
5317 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
5318 \csname#1\endcsname}\noexpand\@gls@quotechar
5319 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
5320 \csname#1\endcsname}\noexpand\@gls@quotechar}%
5321 \noexpand\def\noexpand\@gls@checkescquote{%
5322 \noexpand\@gls@checkescquote####3\noexpand\null}%
5323 \noexpand\fi
5324 \noexpand\else
5325 \noexpand\edef\noexpand\@gls@checkedmkidx{%
5326 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
5327 \noexpand\@gls@quotechar\noexpand\string
5328 \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
5329 \noexpand\ifx\noexpand\null####3\noexpand\null
5330 \noexpand\def\noexpand\@gls@checkescquote{%
5331 \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
5332 \expandonce{\csname#1\endcsname}\noexpand\null}%
5333 \noexpand\else
5334 \noexpand\def\noexpand\@gls@checkescquote{%
5335 \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
5336 ####3\noexpand\null}%
5337 \noexpand\fi
5338 \noexpand\fi
5339 \noexpand\@gls@checkescquote
5340 }%
5341 }%
5342 \@gls@docheckquotedef
5343 }
5344 \newcommand*{\@gls@nosetquote}[1]{\PackageError{glossaries}%
5345 {\string\GlsSetQuote\space not permitted here}%
5346 {Move \string\GlsSetQuote\space earlier in the preamble, as
5347 soon as possible after glossaries.sty has been loaded}}
5348 \fi

```

ramakeindexopts

```

5349 \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`

```

5350 \newcommand{\noist}{}

```

Update attributes list

```

5351 \@gls@addpredefinedattributes
5352 \let\writeist\relax
5353 }

```

`\@makeglossary` is an internal command that takes an argument indicating the glossary type. This command will create the glossary file required by `makeindex` for the given glossary type, using the extension supplied by the `<out-ext>` parameter used in `\newglossary` (and it will also activate the `\glossary` command, and create the customized .ist `makeindex` style file).

Note that you can't use `\@makeglossary` for only some of the defined glossaries (with just the base glossaries package). You either need to have a `\makeglossaries` for all glossaries or none (otherwise you will end up with a situation where \TeX is trying to write to a non-existent file). The relevant glossary must be defined prior to using `\@makeglossary`. `glossaries-extra` allows for a hybrid approach.

`\@makeglossary` Unstarred form of `\ifglossaryexists` is used as `\@makeglossary` can't be used with an ignored glossary.

```
5354 \newcommand*\@makeglossary}[1]{%
5355   \ifglossaryexists{#1}%
5356   {%
```

Only create a new write if `savewrites=false` otherwise create a token to collect the information.

```
5357   \ifglssavewrites
5358     \expandafter\newtoks\csname glo@#1@filetok\endcsname
5359   \else
5360     \expandafter\newwrite\csname glo@#1@file\endcsname
5361     \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
5362   \fi
5363   \@gls@renewglossary
5364   \writeist
5365 }%
5366 {%
5367   \PackageError{glossaries}%
5368   {Glossary type ‘#1’ not defined}%
5369   {New glossaries must be defined before using \string\makeglossaries}%
5370 }%
5371 }
```

`\@glsopenfile` Open write file associated with the given glossary.

```
5372 \newcommand*\@glsopenfile}[2]{%
5373   \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
5374   \PackageInfo{glossaries}{Writing glossary file
5375     \jobname.\csname @glotype@#2@out\endcsname}%
5376 }
```

`\@closegls`

```
5377 \newcommand*\@closegls}[1]{%
5378   \closeout\csname glo@#1@file\endcsname
5379 }
```

\@gls@automake Unstarred form of \ifglossaryexists is used as \@gls@automake can't be used with an ignored glossary.

```

5380 \ifglsxindy
5381 \newcommand*{\@gls@automake}[1]{%
5382   \ifglossaryexists{#1}
5383   {%
5384     \@closegls{#1}%
5385     \ifdefstring{\glsorder}{letter}%
5386     {\def\@gls@order{-M ord/letorder }}%
5387     {\let\@gls@order\@empty}%
5388     \ifcsundef{xdy@#1@language}%
5389     {\let\@gls@langmod\@xdy@main@language}%
5390     {\letcs\@gls@langmod{xdy@#1@language}}%
5391     \edef\@gls@dothiswrite{\noexpand\write18{xindy
5392       -I xindy
5393       \@gls@order
5394       -L \@gls@langmod\space
5395       -M \@gls@istfilebase\space
5396       -C \@gls@codepage\space
5397       -t \jobname.\csuse{@glotype@#1@log}
5398       -o \jobname.\csuse{@glotype@#1@in}
5399       \jobname.\csuse{@glotype@#1@out}}}%
5400     }%
5401     \@gls@dothiswrite
5402   }%
5403   {%
5404     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5405   }%
5406 }
5407 \else
5408 \newcommand*{\@gls@automake}[1]{%
5409   \ifglossaryexists{#1}
5410   {%
5411     \@closegls{#1}%
5412     \ifdefstring{\glsorder}{letter}%
5413     {\def\@gls@order{-l }}%
5414     {\let\@gls@order\@empty}%
5415     \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
5416       -s \istfilename\space
5417       -t \jobname.\csuse{@glotype@#1@log}
5418       -o \jobname.\csuse{@glotype@#1@in}
5419       \jobname.\csuse{@glotype@#1@out}}}%
5420     }%
5421     \@gls@dothiswrite
5422   }%
5423   {%
5424     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5425   }%
5426 }

```

5427 \fi

omake@immediate Unstarred form of \ifglossaryexists is used as \@gls@automake@immediate can't be used with an ignored glossary.

```
5428 \ifglxindy
5429 \newcommand*{\@gls@automake@immediate}[1]{%
5430   \ifglossaryexists{#1}
5431   {%
5432     \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
5433       {%
5434         \ifdefstring{\glsorder}{letter}%
5435         {\def\@gls@order{-M ord/letorder }}%
5436         {\let\@gls@order\empty}%
5437         \ifcsundef{xdy@#1@language}%
5438         {\let\@gls@langmod\@xdy@main@language}%
5439         {\letcs\@gls@langmod{xdy@#1@language}}%
5440         \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
5441           -I xindy
5442           \@gls@order
5443           -L \@gls@langmod\space
5444           -M \@gls@istfilebase\space
5445           -C \@gls@codepage\space
5446           -t \jobname.\csuse{@glotype@#1@log}
5447           -o \jobname.\csuse{@glotype@#1@in}
5448           \jobname.\csuse{@glotype@#1@out}}}%
5449         }%
5450         \@gls@dothiswrite
5451       }%
5452       {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
5453         doesn't exist. Rerun may be required}}%
5454     }%
5455   }%
5456   \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5457 }%
5458 }
5459 \else
5460 \newcommand*{\@gls@automake@immediate}[1]{%
5461   \ifglossaryexists{#1}
5462   {%
5463     \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
5464       {%
5465         \ifdefstring{\glsorder}{letter}%
5466         {\def\@gls@order{-l }}%
5467         {\let\@gls@order\empty}%
5468         \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
5469           -s \istfilename\space
5470           -t \jobname.\csuse{@glotype@#1@log}
5471           -o \jobname.\csuse{@glotype@#1@in}
5472           \jobname.\csuse{@glotype@#1@out}}}%
5473       }%
5474     }%
5475   }%
5476 }
```

```

5473     }%
5474     \@gls@dothiswrite
5475 }%
5476 {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gls@type@#1@out}
5477     doesn't exist. Rerun may be required}}%
5478 }%
5479 {%
5480     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5481 }%
5482 }
5483 \fi

```

`\makeglossaries` Issue warning that `\makeglossaries` hasn't been used.

```

5484 \newcommand*{\@warn@nomakeglossaries}{}

    Only use this if warning if \printglossary has been used without \makeglossaries
5485 \newcommand*{\@warn@nomakeglossaries}{\@warn@nomakeglossaries}

```

`\make@immediate`

```

5486 \newcommand{\@gls@@automake@immediate}{%
5487   \ifnum\gls@automake@nr=2\relax
5488     \@for\@gls@type:=\@gls@types\do{%
5489       \ifdefempty{\@gls@type}{}%
5490       {\@gls@automake@immediate{\@gls@type}}%
5491     }%
5492     \glsautomakefalse
5493     \renewcommand*{\@gls@doautomake}{}%
5494 \fi
5495 }

```

`\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`

```

5496 \newcommand*{\makeglossaries}{%
5497   \@domakeglossaries
5498   {%

    If automake=immediate setting is on, use the shell escape now.
5499     \@gls@@automake@immediate

    Define the write used for style file also used for all other output files if savewrites=true.
5500     \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.
5501     \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
5502     \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }

```

If `\@@gls@extramakeindexopts` has been defined, write it:

```
5503 \ifundef\@@gls@extramakeindexopts
5504 {}%
5505 {%
5506   \protected@write\@auxout{}\string\providecommand
5507     \string\@gls@extramakeindexopts[1]{}
5508   \protected@write\@auxout{}\string\@gls@extramakeindexopts
5509     {\@@gls@extramakeindexopts}}%
5510 }%
```

Write the name of the style file to the aux file (needed by `makeglossaries`)

```
5511 \protected@write\@auxout{}\string\@istfilename{\istfilename}}%
5512 \protected@write\@auxout{}\string\@glsorder{\glsorder}}
```

Iterate through each glossary type and activate it.

```
5513 \@for\@glo@type:=\@glo@types\do{%
5514   \ifthenelse{equal{\@glo@type}{}}{}{}%
5515   \@makeglossary{\@glo@type}}%
5516 }%
```

New glossaries must be created before `\makeglossaries` so disable `\newglossary`.

```
5517 \renewcommand*\newglossary[4][]{%
5518   \PackageError{glossaries}{New glossaries
5519     must be created before \string\makeglossaries}{You need
5520     to move \string\makeglossaries\space after all your
5521     \string\newglossary\space commands}}%
```

Any subsequence 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).

```
5522 \let\@makeglossary\@gobble
5523 \let\makeglossaries\relax
```

Disable all commands that have no effect after `\makeglossaries`

```
5524 \@disable@onlypremakeg
```

Allow see key:

```
5525 \let\gls@checkseeallowed\relax
```

Suppress warning about no `\makeglossaries`

```
5526 \let\warn@nomakeglossaries\relax
```

Activate warning about missing `\printglossary`

```
5527 \let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
```

Declare list parser for `\glsdisplaynumberlist`

```
5528 \ifglssavenumberlist
5529   \edef\@gls@dodolistparser{\noexpand\DeclareListParser
5530     {\noexpand\glsnumlistparser}{\delimN}}%
5531   \@gls@dodolistparser
5532 \fi
```

Prevent user from also using \makenoidxglossaries

```
5533 \let\makenoidxglossaries\@no@makeglossaries
```

Prohibit sort key in printgloss family:

```
5534 \renewcommand*{\@printgloss@setsort}{%
5535   \let\@glo@assign@sortkey\@glo@no@assign@sortkey
5536 }%
```

Check the automake setting:

```
5537 \ifglsautomake
5538   \renewcommand*{\@gls@doautomake}{%
5539     \@for\@gls@type:=\@glo@types\do{%
5540       \ifdefempty{\@gls@type}{}%
5541       {\@gls@automake{\@gls@type}}%
5542     }%
5543   }%
5544 \fi
```

Check the sort setting:

```
5545 \@glo@check@sortallowed\makeglossaries
5546 }%
5547 }
```

Must occur in the preamble:

```
5548 \@onlypreamble{\makeglossaries}
```

noprintglossary

```
5549 \newcommand\@makeglossaries@warn@noprintglossary{%
5550   \ifdefstring{\@glo@types}{,}%
5551   {%
5552     \GlossariesWarningNoLine{No glossaries have been defined}%
5553   }%
5554   {%
5555     \GlossariesWarningNoLine{No \string\printglossary\space
5556       or \string\printglossaries\space
5557       found. ^^J(Remove \string\makeglossaries\space if you
5558       don't want any glossaries.) ^^JThis document will not
5559       have a glossary}%
5560   }%
5561 }%
```

\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.

```
5562 \AtEndDocument{%
5563   \warn@nomakeglossaries
5564   \warn@noprintglossary
5565 }
```


noidxglossaries Analogous to \makeglossaries this activates the commands needed for \printnoidxglossary

```
5566 \newcommand*{\makenoidxglossaries}{%
5567 \domakeglossaries
5568 {%
```

Redefine empty glossary warning:

```
5569 \renewcommand{\@gls@noref@warn}[1]{%
5570 \GlossariesWarning{Empty glossary for
5571 \string\printnoidxglossary[type={##1}].
5572 Rerun may be required (or you may have forgotten to use
5573 commands like \string\gls)}}%
5574 }%
```

Don't escape makeindex/xindy characters:

```
5575 \let\@gls@checkmkidxchars\@gobble
```

Don't escape locations:

```
5576 \glscsclocationsfalse
```

Write glossary information to aux instead of glossary files

```
5577 \let\@do@wrglossary\gls@noidxglossary
```

Switch on group headings that use the character code:

```
5578 \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
```

Allow see key:

```
5579 \let\gls@checkseeallowed\relax
```

Redefine cross-referencing macro:

```
5580 \renewcommand{\@do@seeglossary}[2]{%
5581 \protected@edef\@gls@label{\glsdetoklabel{##1}}%
5582 \protected@write\@auxout{}{%
5583 \string\@gls@reference
5584 {\csname glo@\@gls@label @type\endcsname}%
5585 {\@gls@label}%
5586 {%
5587 \string\glsseeformat##2}%
5588 }%
5589 }%
5590 }%
```

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.

```
5591 \AtBeginDocument
5592 {%
5593 \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
5594 }%
```

Change warning about no glossaries

```
5595 \def\warn@noprntglossary{%
5596 \GlossariesWarningNoLine{No \string\printnoidxglossary\space
5597 or \string\printnoidxglossaries ^^J
5598 found. (Remove \string\makenoidxglossaries\space if you
```

```

5599      don't want any glossaries.)^^JThis document will not have a glossary}%
5600 }%

  Suppress warning about no \makeglossaries
5601 \let\warn@nomakeglossaries\relax

  Prevent user from also using \makeglossaries
5602 \let\makeglossaries\@no@makeglossaries

  Allow sort key in printgloss family:
5603 \renewcommand*{\@printgloss@setsort}{%
5604   \let\@glo@assign@sortkey\@glo@assign@sortkey

  Initialise default sort order:
5605   \def\@glo@sorttype{\@glo@default@sorttype}%
5606 }%

  All entries must be defined in the preamble:
5607 \renewcommand*\new@glossaryentry[2]{%
5608   \PackageError{glossaries}{Glossary entries must be
5609     defined in the preamble^^Jwhen you use
5610     \string\makenoidxglossaries}%
5611   {Either move your definitions to the preamble or use
5612     \string\makeglossaries}%
5613 }%

  Redefine \glsentrynumberlist
5614 \renewcommand*{\glsentrynumberlist}[1]{%
5615   \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
5616   \ifdef\@gls@loclist
5617   {%
5618     \glsnoidxloclist{\@gls@loclist}%
5619   }%
5620   {%
5621     ??\glsdoifexists{##1}%
5622     {%
5623       \GlossariesWarning{Missing location list for '##1'. Either
5624         a rerun is required or you haven't referenced the entry}%
5625     }%
5626   }%
5627 }%

  Redefine \glsdisplaynumberlist
5628 \renewcommand*{\glsdisplaynumberlist}[1]{%
5629   \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
5630   \ifdef\@gls@loclist
5631   {%
5632     \def\@gls@noidxloclist@sep{%
5633       \def\@gls@noidxloclist@sep{%
5634         \def\@gls@noidxloclist@sep{%
5635           \glsnumlistsep
5636         }%

```

```

5637         \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
5638     }%
5639 }%
5640 \def\@gls@noidxloclist@finalsep{}%
5641 \def\@gls@noidxloclist@prev{}%
5642 \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
5643 \@gls@noidxloclist@finalsep
5644 \@gls@noidxloclist@prev
5645 }%
5646 {%
5647 ??\glsdoifexists{##1}%
5648 {%
5649     \GlossariesWarning{Missing location list for ‘##1’. Either
5650         a rerun is required or you haven’t referenced the entry}%
5651 }%
5652 }%
5653 }%

```

Provide a generic way of iterating through the number list:

```

5654 \renewcommand*\@glsnumberlistloop}[3]{%
5655     \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
5656     \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
5657     \let\@gls@org@glsseeformat\glsseeformat
5658     \let\glsnoidxdisplayloc##2\relax
5659     \let\glsseeformat##3\relax
5660     \ifdef\@gls@loclist
5661     {%
5662         \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
5663     }%
5664     {%
5665         ??\glsdoifexists{##1}%
5666         {%
5667             \GlossariesWarning{Missing location list for ‘##1’. Either
5668                 a rerun is required or you haven’t referenced the entry}%
5669         }%
5670     }%
5671     \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
5672     \let\glsseeformat\@gls@org@glsseeformat
5673 }%

```

Modify sanitize sort function

```

5674 \let\@gls@sanitizesort\@gls@noidx@sanitizesort
5675 \let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
5676 \@gls@noidx@setsanitizesort

```

Check sort option allowed.

```

5677 \@glo@check@sortallowed\makenoidxglossaries
5678 }%
5679 }

```

Preamble-only command:

```
5680 \@onlypreamble{\makenoidxglossaries}
```

```
snumberlistloop \glsnumberlistloop{<label>}{<handler>}
```

```
5681 \newcommand*{\glsnumberlistloop}[2]{%
5682   \PackageError{glossaries}{\string\glsnumberlistloop\space
5683     only works with \string\makenoidxglossaries}{}%
5684 }
```

```
listloophandler Handler macro for \glsnumberlistloop. (The argument should be in the form \glsnoidxdisplayloc
  {<prefix>}{<counter>}{<format>}{<n>})
```

```
5685 \newcommand*{\glsnoidxnumberlistloophandler}[1]{%
5686   #1%
5687 }
```

```
@makeglossaries Can't use both \makeglossaries and \makenoidxglossaries
```

```
5688 \newcommand*{\@no@makeglossaries}{%
5689   \PackageError{glossaries}{You can't use both
5690     \string\makeglossaries\space and \string\makenoidxglossaries}%
5691   {Either use one or other (or none) of those commands but not both
5692     together.}%
5693 }
```

```
@gls@noref@warn Warning when no instances of \@gls@reference found.
```

```
5694 \newcommand{\@gls@noref@warn}[1]{%
5695   \GlossariesWarning{\string\makenoidxglossaries\space
5696     is required to make \string\printnoidxglossary[type={#1}] work}%
5697 }
```

1.14 Writing information to associated files

```
s@noidxglossary Write the glossary information to the aux file (for the 'noidx' method):
```

```
5698 \newcommand*{\gls@noidxglossary}{%
5699   \protected@write\@auxout{}{%
5700     \string\@gls@reference
5701     {\csname glo@\@gls@label @type\endcsname}%
5702     {\@gls@label}%
5703     {\string\glsnoidxdisplayloc
5704       {\@glo@counterprefix}%
5705       {\@gls@counter}%
5706       {\@glsnumberformat}%
5707       {\@glslocref}%
5708     }%
5709   }%
5710 }
```

`\istfile` Deprecated.

```
5711 \providecommand\istfile{\glswrite}
```

At the end of the document, the files should be created if `savewrites=true`.

```
5712 \AtEndDocument{%
5713   \glswritefiles
5714 }
```

`\@glswritefiles` Only write the files if `savewrites=true`.

```
5715 \newcommand*{\@glswritefiles}{%
```

Iterate through all the glossaries.

```
5716   \forallglossaries{\@glo@type}{%
```

Check for empty glossaries (patch provided by Patrick Häcker)

```
5717     \ifcsundef{glo@\@glo@type @filetok}%
5718     {%
5719       \def\gls@tmp{}%
5720     }%
5721     {%
5722       \edef\gls@tmp{\expandafter\the
5723         \csname glo@\@glo@type @filetok\endcsname}%
5724     }%
5725     \ifx\gls@tmp\@empty
5726       \ifx\@glo@type\glsdefaulttype
5727         \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
5728           entries.^^JRemember to use package option ‘nomain’ if
5729 you
5730           don’t want to^^Juse the main glossary}%
5731       \else
5732         \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
5733           entries}%
5734       \fi
5735     \else
5736       \@glsopenfile{\glswrite}{\@glo@type}%
5737       \immediate\write\glswrite{%
5738         \expandafter\the
5739         \csname glo@\@glo@type @filetok\endcsname}%
5740       \immediate\closeout\glswrite
5741     \fi
5742   }%
5743 }
```

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 `\theglstrycounter` before using `\gls@glossary`.

`\gls@glossary`

```
5744 \newcommand*{\gls@glossary}[1]{%
5745   \gls@glossary{#1}%
5746 }
```

`\@gls@glossary`

`\@gls@glossary{<type>}{<indexing info>}`

(In v4.10, `\@glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\@makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as memoir changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\@makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within `<indexing info>` (so you can do, for example, `\index{%@%}`), and the original design of `\@glossary` here was actually a legacy from the old glossary package. With the glossaries package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the sort key), the actual value (given by `\glossentry{<label>}` or `\subglossentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glspercentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```
5747 \newcommand*{\@gls@glossary}[2]{%
5748   \if@gls@debug
5749     \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
5750   \fi
5751 }
```

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.

`s@renewglossary`

```
5752 \newcommand{\@gls@renewglossary}{%
5753   \gdef\@gls@glossary##1{\@bsphack\beginingroup\gls@wrglossary{##1}}%
5754   \let\@gls@renewglossary\@empty
5755 }
```

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`

```
5756 \newcommand*{\gls@wrglossary}[2]{%
```

```

5757 \ifglssavewrites
5758   \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
5759   \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
5760     \expandafter{\@gls@tmp^^J}%
5761 \else
5762   \ifcsdef{glo@#1@file}%
5763   {%
5764     \expandafter\protected@write\csname glo@#1@file\endcsname{%
5765       \gls@disablepagerefexpansion}{#2}%
5766   }%
5767   {%
5768     \ifignoredglossary{#1}{}%
5769     {%
5770       \GlossariesWarning{No file defined for glossary ‘#1’}%
5771     }%
5772   }%
5773 \fi
5774 \endgroup\@esphack
5775 }

```

\@do@wrglossary

```

5776 \newcommand*{\@do@wrglossary}[1]{%
5777   \glswriteentry{#1}{\@do@wrglossary{#1}}%
5778 }

```

\glswriteentry Provide a user level command so the user can customize whether or not a line should be added to the glossary. The arguments are the label and the code that writes to the glossary file.

```

5779 \newcommand*{\glswriteentry}[2]{%
5780   \ifglindexonlyfirst
5781     \ifglused{#1}{#2}%
5782   \else
5783     #2%
5784   \fi
5785 }

```

protected@pagefmts List of page formats to be protected against expansion.

```

5786 \newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
5787   \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

```

agerefexpansion

```

5788 \newcommand*{\gls@disablepagerefexpansion}{%
5789   \@for\@gls@this:=\gls@protected@pagefmts\do
5790   {%
5791     \expandafter\let\@gls@this\relax
5792   }%
5793 }

```

```

\gls@alphpage
5794 \newcommand*{\gls@alphpage}{\@alph\c@page}

\gls@Alphpage
5795 \newcommand*{\gls@Alphpage}{\@Alph\c@page}

\gls@numberpage
5796 \newcommand*{\gls@numberpage}{\number\c@page}

\gls@arabicpage
5797 \newcommand*{\gls@arabicpage}{\@arabic\c@page}

\gls@romanpage
5798 \newcommand*{\gls@romanpage}{\romannumeral\c@page}

\gls@Romanpage
5799 \newcommand*{\gls@Romanpage}{\@Roman\c@page}

```

```

protectedpagefmt \glsaddprotectedpagefmt{\csname}

```

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 T_EX register as the argument (`\csname`\c@page must be valid).

```

5800 \newcommand*{\glsaddprotectedpagefmt}[1]{%
5801   \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}}%
5802   \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
5803   \protected@eappto\@wrglossarynumberhook{%
5804     \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
5805     \expandonce{\csname#1\endcsname}%
5806     \noexpand\def\expandonce{\csname#1\endcsname}{%
5807       \noexpand\@wrglossary@pageformat
5808       \expandonce{\csname gls#1page\endcsname}%
5809       \expandonce{\csname org@gls#1\endcsname}%
5810     }%
5811   }%
5812 }

```

`ssarynumberhook` Hook used by `\@do@wrglossary`

```

5813 \newcommand*\@wrglossarynumberhook{}

```

`sary@pageformat`

```

5814 \newcommand{\@wrglossary@pageformat}[3]{%
5815   \ifx#3\c@page #1\else #2#3\fi
5816 }

```


`@@do@wrglossary` Write the glossary entry in the appropriate format.

```
5817 \newcommand*{\@@do@wrglossary}[1]{%
5818   \ifglseclocations
5819     \@@do@esc@wrglossary{#1}%
5820   \else
5821     \@@do@noesc@wrglossary{#1}%
5822   \fi
5823 }
```

`oesc@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.

```
5824 \newcommand*{\@@do@noesc@wrglossary}[1]{%
```

Don't fully expand yet.

```
5825   \expandafter\def\expandafter\@glslocdef\expandafter{\theglsentrycounter}%
5826   \expandafter\def\expandafter\@glsHlocdef\expandafter{\theHglentrycounter}%
```

Find the prefix if `\@glsHlocdef` and `\@glslocdef` aren't the same.

```
5827   \ifx\@glsHlocdef\@glslocdef
5828     \def\@glo@counterprefix{}%
5829   \else
```

The value of the counter isn't important here as it's the prefix that's of interest. (`\c@page` will have the same value in both `\theglsentrycounter` and `\theHglentrycounter` at this point, even if it hasn't been updated yet. The page number is not expected to occur in the prefix.)

```
5830     \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
5831       {\@glslocdef}{\@glsHlocdef}}%
5832     }%
5833     \@do@gls@getcounterprefix
5834   \fi
```

De-tok label if required.

```
5835   \protected@edef\@gls@label{\glsdetoklabel{#1}}%
```

Write the information to file:

```
5836   \@@do@@wrglossary
5837 }
```

`owprimitivemods` Conditional to determine whether or not `\@@do@esc@wrglossary` should be allowed to temporarily redefine `\the` and `\number`.

```
5838 \newif\ifglswrallowprimitivemods
5839 \glswrallowprimitivemodstrue
```

`@esc@wrglossary` Write the glossary entry in the appropriate format. (Need to set `\@glsnumberformat` and `\@gls@counter` prior to use.) The argument is the entry's label. This is far more complicated with xindy than with other indexing methods. There are two necessary but conflicting requirements with xindy:

1. all backslashes in the location must be escaped;
2. `\c@page` can't be prematurely expanded.

(With `makeindex` there's the remote possibility that the page compositor is a `makeindex` special character, so that would also need to be escaped.)

For example, suppose `\thepage` is defined as

```
\renewcommand{\thepage}{\tally{page}}
\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
```

where `\tallynum` is a robust command that takes a number as its argument. With all indexing methods other than `xindy`, a deferred write with `\thepage` as the location will expand to `\tallynum{<n>}` where `<n>` is the page number. Since the write is deferred, the page number is correct. (`makeindex` won't accept this location format, but `\makenoidxglossaries` and `bib2gls` are quite happy with it.) Unfortunately, this fails with `xindy` because `xindy` interprets this location as `\tallynum{<n>}` because `\t` represents a the character "t". The location must be written as `\\tallynum{<n>}`.

This means that the location `\tally{page}` must be expanded and then the backslashes must be doubled. Unfortunately `\c@page` mustn't be expanded until the deferred write is performed, so the location actually needs to be expanded to `\tallynum{\the\c@page}` but the backslashes in `\the\c@page` mustn't be escaped. All other backslashes must be escaped. (In this case, only the backslash in `\tallynum` but the location format may include other control sequences.) The code below works on the assumption that commands like `\tally` are defined in the form

```
\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
```

(note the use of `\expandafter` and `\name`) or in the form

```
\newcommand{\tally}[1]{\tallynum{\arabic{#1}}}
```

In the second case, `\arabic` is one of the known commands that's temporarily adjusted to prevent `\c@page` from being prematurely expanded. In the first case, `\the` is temporarily modified (unless `\glswrallowprimitivemodsfalse`) to check if it's followed by `\c@page`. The `\expandafter` ensures that it is. If `\tally` is defined in another way that hides `\c@page` for example using `\the\value{#1}` then the process fails.

With `makeindex`, `\tallynum` needs to expand to just the decimal number while writing the location to the glossary file, otherwise `makeindex` will reject it. This can be done by defining `\glstallypage` so that `\tally` can locally be set to `\arabic` while expansion is occurring. Again, `\c@page` must be protected from expansion until the deferred write occurs.

The expansion before the write occurs also allows the hyper prefix to be determined where `\theH{<counter>}` is defined in the form `<prefix>.\the{<counter>}`. It's possible (although again unlikely) that a `makeindex` character might occur in the prefix, which therefore needs escaping. The prefix is passed as the optional argument of `\setentrycounter` which is needed by commands like `\glshypernumber` to create a hyperlink for a given counter (like `\hyperpage` but for an arbitrary counter).

```
5840 \newcommand*{\@@do@esc@wrglossary}[1]{% please read documented code!
5841 \begingroup
```

First a bit of hackery to prevent premature expansion of `\c@page`. Store original definitions (scoped):

```
5842 \let\gls@orgthe\the
5843 \let\gls@orgnumber\number
5844 \let\gls@orgarabic\@arabic
5845 \let\gls@orgromannumeral\romannumeral
5846 \let\gls@orgalph\@alph
5847 \let\gls@orgAlph\@Alph
5848 \let\gls@orgRoman\@Roman
```

Redefine:

```
5849 \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.

```
5850 \def\gls@the##1{%
5851   \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
5852 \def\the{\expandafter\gls@the}%
5853 \def\gls@number##1{%
5854   \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
5855 \def\number{\expandafter\gls@number}%
5856 \fi
5857 \def\@arabic##1{%
5858   \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
5859 \def\romannumeral##1{%
5860   \ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
5861 \def\@Roman##1{%
5862   \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
5863 \def\@alph##1{%
5864   \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
5865 \def\@Alph##1{%
5866   \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%

```

Add hook to allow for other number formats:

```
5867 \@wrglossarynumberhook
```

Prevent expansion:

```
5868 \gls@disablepagerefexpansion
```

Now store location in `\@glslocref`:

```
5869 \protected@xdef\@glslocref{\theglentrycounter}%
5870 \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.

```
5871 \@gls@checkmkidxchars\@glslocref
```

Check if the hyper-location is the same as the location and set the hyper prefix.

```
5872 \expandafter\ifx\theglentrycounter\theglentrycounter\relax
5873 \def\@glo@counterprefix{}%

```

```

5874 \else
5875   \protected@edef\@glsHlocref{\theHglentrycounter}%
5876   \@gls@checkmkidxchars\@glsHlocref
5877   \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
5878     {\@glslocref}{\@glsHlocref}%
5879   }%
5880   \@do@gls@getcounterprefix
5881 \fi

```

De-tok label if required

```

5882 \protected@edef\@gls@label{\glsdetoklabel{#1}}%

```

Write the information to file:

```

5883 \@do@@wrglossary
5884 }

```

@do@@wrglossary

```

5885 \newcommand*{\@do@@wrglossary}{%

```

Determine whether to use xindy or makeindex syntax

```

5886 \ifglxindy

```

Need to determine if the formatting information starts with a (or) indicating a range.

```

5887   \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
5888   \def\@glo@range{}%
5889   \expandafter\if\@glo@prefix(\relax
5890     \def\@glo@range{:open-range}%
5891   \else
5892     \expandafter\if\@glo@prefix)\relax
5893     \def\@glo@range{:close-range}%
5894   \fi
5895 \fi

```

Write to the glossary file using xindy syntax.

```

5896   \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
5897     (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
5898       :locref \string"{\@glo@counterprefix}{\@glslocref}\string" %
5899       :attr \string"\@gls@counter\@glo@suffix\string"
5900       \@glo@range
5901     )
5902   }%
5903 \else

```

Convert the format information into the format required for makeindex

```

5904   \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
5905   {\@glo@counterprefix}%

```

Write to the glossary file using makeindex syntax.

```

5906   \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
5907     \string\glossaryentry{\csname glo@\@gls@label @index\endcsname
5908       \@gls@encapchar\@glo@numfmt}{\@glslocref}}%

```

```

5909 \fi
5910 }

```

`\etcounterprefix` Get the prefix that needs to be prepended to counter in order to get the hyper counter. (For example, with the standard article class and hyperref, `\theequation` needs to be prefixed with `\section num` to get the equivalent `\theHequation`.) NB this assumes that the prefix ends with a dot, which is the standard. (Otherwise it makes the xindy location classes more complicated.)

```

5911 \newcommand*\@gls@getcounterprefix[2]{%
5912   \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
5913   \ifx\@gls@thisloc\@gls@thisHloc
5914     \def\@glo@counterprefix{}%
5915   \else
5916     \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
5917       \def\@glo@tmp{##2}%
5918       \ifx\@glo@tmp\@empty
5919         \def\@glo@counterprefix{}%
5920       \else
5921         \def\@glo@counterprefix{##1}%
5922       \fi
5923     }%
5924     \@gls@get@counterprefix#2.#1\end@getprefix

```

Warn if no prefix can be formed.

```

5925   \ifx\@glo@counterprefix\@empty
5926     \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
5927       prefixing~Jlocation ‘#1’. You need to modify the
5928       definition of \string\theH\@gls@counter~Jotherwise you
5929       will get the warning: “name{\@gls@counter.#1}’ has been~J
5930       referenced but does not exist”}%
5931   \fi
5932 \fi
5933 }

```

1.15 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.

```

5934 \newcommand{\@do@seeglossary}[2]{%
5935   \def\@gls@xref{#2}%
5936   \@onelevel@sanitize\@gls@xref
5937   \@gls@checkmkidxchars\@gls@xref
5938   \ifglxindy
5939     \gls@glossary{\csname glo@#1@type\endcsname}{%
5940       (indexentry
5941         :tkey (\csname glo@#1@index\endcsname)
5942         :xref (\string"\@gls@xref\string")

```

```

5943      :attr \string"see\string"
5944    )
5945  }%
5946 \else
5947   \gls@glossary{\csname glo@#1@type\endcsname}{%
5948   \string\glossaryentry{\csname glo@#1@index\endcsname
5949   \@gls@encapchar glsseeformat\@gls@xref}{Z}}}%
5950 \fi
5951 }

```

`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```

5952 \def\@gls@fixbraces#1#2#3\@nil{%
5953   \ifx#2[\relax
5954     \@gls@fixbraces#1#2#3\@end@fixbraces
5955   \else
5956     \def#1{{#2#3}}%
5957   \fi
5958 }

```

`@@gls@fixbraces`

```

5959 \def@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
5960   \def#1{[#2]{#3}}%
5961 }

```

`\glssee` `\glssee{<label>}{<cross-ref list>}`

```

5962 \newrobustcmd*{\glssee}[3][\seename]{%
5963   \@do@seeglossary{#2}{#1}{#3}}
5964 \newcommand*{\@glssee}[3][\seename]{%
5965   \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.

```

5966 \newrobustcmd*{\glsseeformat}[3][\seename]{%
5967   \emph{#1} \glsseelist{#2}}

```

`\glsseelist` `\glsseelist{<list>}` formats list of entry labels.

```

5968 \newrobustcmd*{\glsseelist}[1]{%

```

If there is only one item in the list, set the last separator to do nothing.

```

5969   \let\@gls@dolast\relax

```

Don’t display separator on the first iteration of the loop

```

5970   \let\@gls@donext\relax

```

Iterate through the labels

```

5971   \@for\@gls@thislabel:=#1\do{%

```

Check if on last iteration of loop

```

5972     \ifx\@xfor@nextelement\@nnil
5973       \@gls@dolast

```

```

5974 \else
5975 \@gls@donext
5976 \fi

```

Display the entry for this label. (Expanding label as it's a temporary control sequence that's used elsewhere.)

```

5977 \expandafter\glsseeitem\expandafter{\@gls@thislabel}%

```

Update separators

```

5978 \let\@gls@dolast\glsseelastsep
5979 \let\@gls@donext\glsseesep
5980 }%
5981 }

```

`\glsseelastsep` Separator to use between penultimate and ultimate entries in a cross-referencing list.

```

5982 \newcommand*{\glsseelastsep}{\space\andname\space}

```

`\glsseesep` Separator to use between entries in a cross-referencing list.

```

5983 \newcommand*{\glsseesep}{, }

```

`\glsseeitem` `\glsseeitem{<label>}` formats individual entry in a cross-referencing list.

```

5984 \newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}

```

`\glsseeitemformat` As from v3.0, default is to use `\glsentrytext` instead of `\glsentryname`. (To avoid problems with the name key being sanitized, although this is no longer a problem now.)

```

5985 \newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}

```

1.16 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.

`\save@numberlist` Provide command to store number list.

```

5986 \newcommand*{\gls@save@numberlist}[1]{%
5987 \ifglssavenumberlist
5988 \toks@{#1}%

5989 \protected@edef\@do@writeaux@info{%
5990 \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
5991 }%
5992 \@onelevel@sanitize\@do@writeaux@info
5993 \protected@write\@auxout{}{\@do@writeaux@info}%
5994 \fi
5995 }

```

`\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.)

```
5996 \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.

```
5997 \ifcsundef{printglossary}{}%
```

```
5998 {%
```

If `\printglossary` is already defined, issue a warning and undefine it.

```
5999 \@gls@warnonglossdefined
```

```
6000 \undef\printglossary
```

```
6001 }
```

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.

`exists@noignored`

```
6002 \newcommand*{\@printgloss@checkexists@noignored}[2]{%
```

```
6003 \ifglossaryexists{#1}%
```

```
6004 {#2}%
```

```
6005 {%
```

```
6006 \ifignoredglossary{#1}%
```

```
6007 {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
```

```
6008 {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
```

```
6009 }%
```

```
6010 }
```

`ts@allowignored` For use with `\printunsrtglossary`.

```
6011 \newcommand*{\@printgloss@checkexists@allowignored}[2]{%
```

```
6012 \s@ifglossaryexists{#1}%
```

```
6013 {#2}%
```

```
6014 {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
```

```
6015 }
```

`oss@checkexists`

```
6016 \let\@printgloss@checkexists\@printgloss@checkexists@noignored
```

`\printglossary` has an optional argument. The default value is to set the glossary type to the main glossary.

```
6017 \newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
```

```
6018 \let\@printgloss@checkexists\@printgloss@checkexists@noignored
```

```
6019 \@printglossary{#1}{\@printglossary}%
```

```
6020 }
```


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`

```
6021 \newcommand*{\printglossaries}{%
6022   \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
6023 }
```

`\printnoidxglossary` Provide an alternative to `\printglossary` that doesn't require an external indexing application. Entries won't be sorted and the location list will be empty.

```
6024 \newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
6025   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
6026   \@printglossary{#1}{\@printnoidx@glossary}%
6027 }
```

`\printnoidxglossaries` Analogous to `\printglossaries`

```
6028 \newcommand*{\printnoidxglossaries}{%
6029   \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
6030 }
```

`\printgloss@setsort` Initialise to do nothing.

```
6031 \newcommand*{\@printgloss@setsort}{}%
```

`\preglossaryhook`

```
6032 \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.

```
6033 \newcommand{\@printglossary}[2]{%
6034   Set up defaults.
```

```
6034   \def\@glo@type{\glsdefaulttype}%
6035   \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
6036   \def\glossarytoctitle{\glossarytitle}%
6037   \let\org@glossarytitle\glossarytitle
```

```
6038   \def\@glossarystyle{%
6039     \ifx\@glossary@default@style\relax
6040       \GlossariesWarning{No default glossary style provided \MessageBreak
6041         for the glossary '@glo@type'. \MessageBreak
```

```

6042      Using deprecated fallback. \MessageBreak
6043      To fix this set the style with \MessageBreak
6044      \string\setglossarystyle\space or use the \MessageBreak
6045      style key=value option}%
6046  \fi
6047 }%
6048 \def\gls@dotocitle{\glssettocitle{\@glo@type}}%

Store current value of \glossaryentrynumbers. (This may be changed via the optional ar-
gument)
6049 \let\org@glossaryentrynumbers\glossaryentrynumbers

Localise the effects of the optional argument
6050 \bgroup

Activate or deactivate sort key:
6051 \@printgloss@setsort

Determine settings specified in the optional argument.
6052 \setkeys{printgloss}{#1}%

Does the glossary exist?
6053 \@printgloss@checkexists{\@glo@type}%
6054 {%

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)
6055 \ifx\glossarytitle\org@glossarytitle
6056 \else
6057 \expandafter\let\csname @glo@type\@glo@type @title\endcsname
6058 \glossarytitle
6059 \fi

Allow a high-level user command to indicate the current glossary
6060 \let\currentglossary\@glo@type

Enable individual number lists to be suppressed.
6061 \let\org@glossaryentrynumbers\glossaryentrynumbers
6062 \let\glsnonextpages\@glsnonextpages

Enable individual number list to be activated:
6063 \let\glsnextpages\@glsnextpages

Enable suppression of description terminators.
6064 \let\nopostdesc\@nopostdesc

Set up the entry for the TOC
6065 \gls@dotocitle

Set the glossary style
6066 \@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):

```

6067 \let\gls@org@glossaryentryfield\glossentry
6068 \let\gls@org@glossarysubentryfield\subglossentry
6069 \renewcommand{\glossentry}[1]{%
6070   \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
6071   \gls@org@glossaryentryfield{##1}%
6072 }%
6073 \renewcommand{\subglossentry}[2]{%
6074   \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
6075   \gls@org@glossarysubentryfield{##1}{##2}%
6076 }%
6077 \@gls@preglossaryhook

```

Now do the handler macro that deals with the actual glossary:

```

6078   #2%
6079 }%

End the current scope
6080 \egroup

Reset \glossaryentrynumbers
6081 \global\let\glossaryentrynumbers\@org@glossaryentrynumbers

Suppress warning about no \printglossary
6082 \global\let\warn@noprintglossary\relax
6083 }

```

`@print@glossary` Internal workings of `\printglossary` dealing with reading the external file.

```

6084 \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.)

```

```

6085 \makeatletter

```

Input the glossary file, if it exists.

```

6086 \@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%

If the glossary file doesn't exist, do \null. (This ensures that the page is shipped out and all
write commands are done.) This might produce an empty page, but at this point the docu-
ment isn't complete, so it shouldn't matter.

```

```

6087 \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
6088 {}%
6089 {\null}%

```

If `xindy` is being used, need to write the language dependent information to the `.aux` file for `makeglossaries`.

```

6090 \ifglsxindy
6091   \ifcsundef{@xdy@\@glo@type @language}%
6092   {%
6093     \edef\@do@auxoutstuff{%
6094       \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.

```

6095      \noexpand\immediate\noexpand\write\@auxout{%
6096      \string\providecommand\string\@xdylanguage[2]{}}%
6097      \noexpand\immediate\noexpand\write\@auxout{%
6098      \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
6099      }%
6100    }%
6101  }%
6102  {%
6103    \edef\@do@auxoutstuff{%
6104      \noexpand\AtEndDocument{%
6105        \noexpand\immediate\noexpand\write\@auxout{%
6106        \string\providecommand\string\@xdylanguage[2]{}}%
6107        \noexpand\immediate\noexpand\write\@auxout{%
6108        \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
6109        @language\endcsname}}%
6110      }%
6111    }%
6112  }%
6113    \@do@auxoutstuff
6114    \edef\@do@auxoutstuff{%
6115      \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.

```

6116      \noexpand\immediate\noexpand\write\@auxout{%
6117      \string\providecommand\string\@gls@codepage[2]{}}%
6118      \noexpand\immediate\noexpand\write\@auxout{%
6119      \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
6120    }%
6121  }%
6122    \@do@auxoutstuff
6123  \fi

```

Activate warning if \makeglossaries hasn't been used.

```

6124  \renewcommand*{\@warn@nomakeglossaries}{%
6125    \GlossariesWarningNoLine{\string\makeglossaries\space
6126    hasn't been used,^^Jthe glossaries will not be updated}%
6127  }%
6128 }

```

The sort macros all have the syntax:

`\@glo@sortmacro@<order>{<type>}`

where *<order>* is the sort order as specified by the sort key and *<type>* is the glossary type. (The referenced entry list is stored in \@glsref@<type>. The actual sorting is done by \@glo@sortentries{<handler>}{<type>}).

glo@sortentries

```
6129 \newcommand*{\@glo@sortentries}[2]{%
6130   \glosortentrieswarning
6131   \def\@glo@sortinglist{}%
6132   \def\@glo@sortinghandler{#1}%

6133   \protected@edef\@glo@type{#2}%
6134   \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
6135   \csdef{\@glsref@#2}{}%
6136   \@for\@this@label:=\@glo@sortinglist\do{%
```

Has this entry already been added?

```
6137     \xifinlistcs{\@this@label}{\@glsref@#2}%
6138     {}%
6139     {%
6140       \listcsxadd{\@glsref@#2}{\@this@label}%
6141     }%
6142     \ifcsdef{\@glo@sortingchildren@\@this@label}%
6143     {%
6144       \@glo@addchildren{#2}{\@this@label}%
6145     }%
6146     {}%
6147   }%
6148 }
```

glo@addchildren

`\@glo@addchildren{<type>}{<parent>}`

```
6149 \newcommand*{\@glo@addchildren}[2]{%
```

Scope to allow nesting.

```
6150   \bgroup
6151     \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
6152     \@for\@this@childlabel:=\@glo@childlist\do
6153     {%
```

Check this label hasn't already been added.

```
6154       \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
6155       {}%
6156       {%
6157         \listcsxadd{\@glsref@#1}{\@this@childlabel}%
6158       }%
```

Does this child have children?

```
6159       \ifcsdef{\@glo@sortingchildren@\@this@childlabel}%
6160       {%
6161         \@glo@addchildren{#1}{\@this@childlabel}%
6162       }%
6163       {%
6164       }%
6165     }%
```

```

6166 \egroup
6167 }

```

@do@sortentries

```

6168 \newcommand*{\@glo@do@sortentries}[1]{%
6169   \ifglshasparent{#1}%
6170   {%
      This entry has a parent, so add it to the child list
6171     \protected@edef\@glo@parent{\csuse{glo@glstdetoklabel{#1}@parent}}%
6172     \ifcsundef{glo@sortingchildren@\@glo@parent}%
6173     {%
6174       \csdef{glo@sortingchildren@\@glo@parent}{}%
6175     }%
6176     {}%
6177     \expandafter\@glo@sortedinsert
6178     \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%

```

Has the parent been added?

```

6179   \xifinlistcs{\@glo@parent}{@glstoklabel{#1}@parent}%
6180   {%

```

Yes, it has so do nothing.

```

6181   }%
6182   {%

```

No, it hasn't so add it now.

```

6183     \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
6184     }%
6185   }%
6186   {%
6187     \@glo@sortedinsert{\@glo@parent}{#1}%
6188   }%
6189 }

```

lo@sortedinsert

```
\@glo@sortedinsert{\@glo@parent}{#1}
```

Insert into list.

```

6190 \newcommand*{\@glo@sortedinsert}[2]{%
6191   \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
6192 }%

```

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).

orthandler@word

```

6193 \newcommand*{\@glo@sorthandler@word}[2]{%
6194   \letcs\@gls@sort@A{glo@glstdetoklabel{#1}@sort}%

```

```

6195 \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
6196 \edef\glo@do@compare{%
6197   \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
6198   {\expandonce\@gls@sort@B}%
6199   {\expandonce\@gls@sort@A}%
6200 }%
6201 \glo@do@compare
6202 }

```

thandler@letter

```

6203 \newcommand*{\@glo@sorthandler@letter}[2]{%
6204   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
6205   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
6206   \edef\glo@do@compare{%
6207     \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
6208     {\expandonce\@gls@sort@B}%
6209     {\expandonce\@gls@sort@A}%
6210   }%
6211   \glo@do@compare
6212 }

```

orthandler@case Case-sensitive sort.

```

6213 \newcommand*{\@glo@sorthandler@case}[2]{%
6214   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
6215   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
6216   \edef\glo@do@compare{%
6217     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
6218     {\expandonce\@gls@sort@B}%
6219     {\expandonce\@gls@sort@A}%
6220   }%
6221   \glo@do@compare
6222 }

```

thandler@nocase Case-insensitive sort.

```

6223 \newcommand*{\@glo@sorthandler@nocase}[2]{%
6224   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
6225   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
6226   \edef\glo@do@compare{%
6227     \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
6228     {\expandonce\@gls@sort@B}%
6229     {\expandonce\@gls@sort@A}%
6230   }%
6231   \glo@do@compare
6232 }

```

@sortmacro@word Sort macro for ‘word’

```

6233 \newcommand*{\@glo@sortmacro@word}[1]{%
6234   \ifdefstring{\@glo@default@sorttype}{standard}%
6235   {%

```

```

6236 \glo@sortentries{\glo@sorthandler@word}{#1}%
6237 }%
6238 {%
6239 \PackageError{glossaries}{Conflicting sort options:^^J
6240 \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
6241 \string\printnoidxglossary[sort=word]}{}%
6242 }%
6243 }

```

ortmacro@letter Sort macro for ‘letter’

```

6244 \newcommand*{\glo@sortmacro@letter}[1]{%
6245 \ifdefstring{\glo@default@sorttype}{standard}%
6246 {%
6247 \glo@sortentries{\glo@sorthandler@letter}{#1}%
6248 }%
6249 {%
6250 \PackageError{glossaries}{Conflicting sort options:^^J
6251 \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
6252 \string\printnoidxglossary[sort=letter]}{}%
6253 }%
6254 }

```

tmacro@standard Sort macro for ‘standard’. (Use either ‘word’ or ‘letter’ order.)

```

6255 \newcommand*{\glo@sortmacro@standard}[1]{%
6256 \ifdefstring{\glo@default@sorttype}{standard}%
6257 {%
6258 \ifcsdef{\glo@sorthandler@glorder}%
6259 {%
6260 \glo@sortentries{\csuse{\glo@sorthandler@glorder}}{#1}%
6261 }%
6262 {%
6263 \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}%
6264 }%
6265 }%
6266 {%
6267 \PackageError{glossaries}{Conflicting sort options:^^J
6268 \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
6269 \string\printnoidxglossary[sort=standard]}{}%
6270 }%
6271 }

```

@sortmacro@case Sort macro for ‘case’

```

6272 \newcommand*{\glo@sortmacro@case}[1]{%
6273 \ifdefstring{\glo@default@sorttype}{standard}%
6274 {%
6275 \glo@sortentries{\glo@sorthandler@case}{#1}%
6276 }%
6277 {%
6278 \PackageError{glossaries}{Conflicting sort options:^^J

```



```

6279     \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
6280     \string\printnoidxglossary[sort=case]}{}%
6281 }%
6282 }

```

ortmacro@nocase Sort macro for ‘nocase’

```

6283 \newcommand*{\@glo@sortmacro@nocase}[1]{%
6284   \ifdefstring{\@glo@default@sorttype}{standard}%
6285   {%
6286     \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
6287   }%
6288   {%
6289     \PackageError{glossaries}{Conflicting sort options:^^J
6290       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
6291       \string\printnoidxglossary[sort=nocase]}{}%
6292   }%
6293 }

```

o@sortmacro@def Sort macro for ‘def’. The order of definition is given in \glo@list@*type*.

```

6294 \newcommand*{\@glo@sortmacro@def}[1]{%
6295   \def\@glo@sortinglist{%
6296     \for@gl@sentries[#1]{\@gl@s@thislabel}%
6297     {%
6298       \xifinlistcs{\@gl@s@thislabel}{\@gl@sref@#1}%
6299     }%
6300     \listcsadd{\@glo@sortinglist}{\@gl@s@thislabel}%
6301   }%
6302   {%

```

Hasn't been referenced.

```

6303   }%
6304 }%
6305 \cslet{\@gl@sref@#1}{\@glo@sortinglist}%
6306 }

```

ortmacro@def@do This won't include parent entries that haven't been referenced.

```

6307 \newcommand*{\@glo@sortmacro@def@do}[1]{%
6308   \ifinlistcs{#1}{\@gl@sref@\@glo@type}%
6309   {}%
6310   {%
6311     \listcsadd{\@gl@sref@\@glo@type}{#1}%
6312   }%
6313   \ifcsdef{\@glo@sortingchildren@#1}%
6314   {%
6315     \@glo@addchildren{\@glo@type}{#1}%
6316   }%
6317   {}%
6318 }

```

o@sortmacro@use Sort macro for ‘use’. (No sorting is required, as the entries are already in order of use, so do nothing.)

```
6319 \newcommand*{\@glo@sortmacro@use}[1]{}%
```

@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.

```
6320 \newcommand*{\@print@noidx@glossary}{%
```

```
6321   \ifcsdef{@gls@ref@{\@glo@type}}%
```

```
6322   {%
```

Sort the entries:

```
6323   \ifcsdef{@glo@sortmacro@{\@glo@sorttype}}%
```

```
6324   {%
```

```
6325     \csuse{@glo@sortmacro@{\@glo@sorttype}}{\@glo@type}}%
```

```
6326   }%
```

```
6327   {%
```

```
6328     \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
```

```
6329   }%
```

Do the glossary heading and preamble

```
6330   \glossarysection[\glossarytoctitle]{\glossarytitle}}%
```

```
6331   \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.

```
6332   \def\@gls@currentlettergroup{}%
```

```
6333   \begin{theglossary}}%
```

```
6334   \glossaryheader
```

```
6335   \glsresetentrylist
```

Iterate through the entries.

```
6336   \forlistcsloop{\@gls@noidx@do}{@gls@ref@{\@glo@type}}%
```

Finally end the glossary and do the postamble:

```
6337   \end{theglossary}}%
```

```
6338   \glossarypostamble
```

```
6339   }%
```

```
6340   {%
```

```
6341     \@gls@noref@warn{\@glo@type}}%
```

```
6342   }%
```

```
6343 }
```

\glo@grabfirst This won’t work for UTF-8 and inputenc.

```
6344 \def\glo@grabfirst#1#2\@nil{%
```

```
6345   \def\@gls@firsttok{#1}}%
```

```

6346 \ifdefempty\@gls@firsttok
6347 {%
6348   \def\@glo@thislettergrp{0}%
6349 }%
6350 {%

```

Sanitize it:

```

6351   \@onelevel@sanitize\@gls@firsttok

```

Fetch the first letter:

```

6352   \expandafter\@glo@grabfirst\@gls@firsttok{}\{}\@nil
6353 }%
6354 }

```

\@glo@grabfirst

```

6355 \def\@glo@grabfirst#1#2\@nil{%
6356   \ifdefempty\@glo@thislettergrp
6357   {%
6358     \def\@glo@thislettergrp{glssymbols}%
6359   }%
6360   {%
6361     \count@=\uccode'#1\relax
6362     \ifnum\count@=0\relax
6363       \def\@glo@thislettergrp{glssymbols}%
6364     \else
6365       \ifdefstring\@glo@sorttype{case}%
6366       {%
6367         \count@='#1\relax
6368       }%
6369       {%
6370       }%
6371       \edef\@glo@thislettergrp{\the\count@}%
6372     \fi
6373   }%
6374 }

```

\@gls@noidx@do Handler for list iteration used by \@print@noidx@glossary. The argument is the entry label.
This only allows one sublevel.

```

6375 \newcommand{\@gls@noidx@do}[1]{%

```

Get this entry's location list

```

6376   \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%

```

Does this entry have a parent?

```

6377   \ifglshasparent{#1}%
6378   {%

```

Has a parent.

```

6379   \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
6380   \ifdefvoid{\@gls@loclist}
6381   {%

```

```

6382     \subglossentry{\gls@level}{#1}{}%
6383 }%
6384 {%
6385     \subglossentry{\gls@level}{#1}%
6386     {%
6387         \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
6388     }%
6389 }%
6390 }%
6391 {%

```

Doesn't have a parent Get this entry's sort key

```

6392     \letcs{\@gls@sort}{\glo@\glsdetoklabel{#1}@sort}%

```

Fetch the first letter:

```

6393     \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
6394     \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
6395     {}%
6396     {%

```

Do the group header:

```

6397     \ifdefempty{\@gls@currentlettergroup}{}%
6398     {%

```

The group skip may start a new scope, so make a global assignment.

```

6399         \global\let\@glo@thislettergrp\@glo@thislettergrp
6400         \glsgroupskip
6401     }%
6402     \glsgroupheading{\@glo@thislettergrp}%
6403 }%

6404     \global\let\@gls@currentlettergroup\@glo@thislettergrp

```

Do this entry:

```

6405     \ifdefvoid{\@gls@loclist}
6406     {%
6407         \glossentry{#1}{}%
6408     }%
6409     {%
6410         \glossentry{#1}%
6411         {%
6412             \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
6413         }%
6414     }%
6415 }%
6416 }

```

`\glsnoidxloclist` `\glsnoidxloclist{\langle list cs \rangle}`

Display location list.

```

6417 \newcommand*{\glsnoidxloclist}[1]{%
6418   \def\@gls@noidxloclist@sep{}%
6419   \def\@gls@noidxloclist@prev{}%
6420   \forlistloop{\glsnoidxloclisthandler}{#1}%
6421 }

```

`xloclisthandler` Handler for location list iterator.

```

6422 \newcommand*{\glsnoidxloclisthandler}[1]{%
6423   \ifdefstring{\@gls@noidxloclist@prev}{#1}%
6424   {%
        Same as previous location so skip.
6425   }%
6426   {%
6427     \@gls@noidxloclist@sep
6428     #1%
6429     \def\@gls@noidxloclist@sep{\delimN}%
6430     \def\@gls@noidxloclist@prev{#1}%
6431   }%
6432 }

```

`yloclisthandler` Handler for location list iterator when used with `\glsdisplaynumberlist`.

```

6433 \newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
6434   \ifdefstring{\@gls@noidxloclist@prev}{#1}%
6435   {%
        Same as previous location so skip.
6436   }%
6437   {%
6438     \@gls@noidxloclist@sep
6439     \@gls@noidxloclist@prev
6440     \def\@gls@noidxloclist@prev{#1}%
6441   }%
6442 }

```

`noidxdisplayloc` `\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<location>}`

Display a location in the location list.

```

6443 \newcommand*\glsnoidxdisplayloc[4]{%
6444   \setentrycounter[#1]{#2}%
6445   \csuse{#3}{#4}%
6446 }

```

`\@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.

```

6447 \newcommand*\@gls@reference[3]{%

```

Add to label list

```
6448 \glsdoifexistsorwarn{#2}%  
6449 {%  
6450 \ifcsundef{@glsref@#1}{\csgdef{@glsref@#1}{}}{}%  
6451 \ifinlistcs{#2}{@glsref@#1}%  
6452 {}%  
6453 {\listcsgadd{@glsref@#1}{#2}}%
```

Add to location list

```
6454 \ifcsundef{glo@glstdetoklabel{#2}@loclist}%  
6455 {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%  
6456 {}%  
6457 \listcsgadd{glo@glstdetoklabel{#2}@loclist}{#3}%  
6458 }%  
6459 }
```

The keys that can be used in the optional argument to `\printglossary` or `\printnoidxglossary` are as follows: The type key sets the glossary type.

```
6460 \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`.

```
6461 \define@key{printgloss}{title}{%  
6462 \def@glossarytitle{#1}%  
6463 \let@gls@dotoc@title\relax  
6464 }
```

The toctitle sets the text used for the relevant entry in the table of contents.

```
6465 \define@key{printgloss}{toctitle}{%  
6466 \def@glossarytoctitle{#1}%  
6467 \let@gls@dotoc@title\relax  
6468 }
```

The style key sets the glossary style (but only for the given glossary).

```
6469 \define@key{printgloss}{style}{%  
6470 \ifcsundef{@glsstyle@#1}%  
6471 {%  
6472 \PackageError{glossaries}%  
6473 {Glossary style ‘#1’ undefined}{}%  
6474 }%  
6475 {%  
6476 \def@glossarystyle{\setglossentrycompatibility  
6477 \csname @glsstyle@#1\endcsname}%  
6478 }%  
6479 }
```

The numberedsection key determines if this glossary should be in a numbered section.

```
6480 \define@choicekey{printgloss}{numberedsection}%  
6481 [\gls@numberedsection@val\gls@numberedsection@nr]%  
6482 {false,nolabel,autolabel,nameref}[nolabel]%  
6483 {%
```

```

6484 \ifcase\gls@numberedsection@nr\relax
6485   \renewcommand*{\@@glossarysecstar}{*}%
6486   \renewcommand*{\@@glossaryseclabel}{}%
6487 \or
6488   \renewcommand*{\@@glossarysecstar}{}%
6489   \renewcommand*{\@@glossaryseclabel}{}%
6490 \or
6491   \renewcommand*{\@@glossarysecstar}{}%
6492   \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
6493 \or
6494   \renewcommand*{\@@glossarysecstar}{*}%
6495   \renewcommand*{\@@glossaryseclabel}{%
6496     \protected@edef\@currentlabelname{\glossarytoctitle}%
6497     \label{\glsautoprefix\@glo@type}}%
6498 \fi
6499 }

```

The `nogroupskip` key determines whether or not there should be a vertical gap between glossary groups.

```

6500 \define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
6501   \csuse{glsnogroupskip#1}%
6502 }

```

The `nopostdot` key has the same effect as the package option of the same name.

```

6503 \define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
6504   \csuse{glsnopostdot#1}%
6505 }

```

`CounterLabelPrefix` Make it easier to redefine the label prefix.

```

6506 \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.

```

6507 \define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
6508   \csuse{glssentrycounter#1}%
6509   \@gls@define@glossaryentrycounter
6510 }

```

The `subentrycounter` key is the same as the package option but localised to the current glossary. Note that this doesn't affect the master/slave counter attributes, which occurs if `subentrycounter` and `entrycounter` package options are set to true.

```

6511 \define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
6512   \csuse{glssubentrycounter#1}%
6513   \@gls@define@glossarysubentrycounter
6514 }

```

The nonnumberlist key determines if this glossary should have a number list.

```
6515 \define@boolkey{printgloss}[gls]{nonnumberlist}[true]{%
6516 \ifglslnonnumberlist
6517   \def\glossaryentrynumbers##1{%
6518 \else
6519   \def\glossaryentrynumbers##1{##1}%
6520 \fi}
```

The sort key sets the glossary sort handler (\printnoidxglossary only).

```
6521 \define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
```

@assign@sortkey Issue error if used with \printglossary

```
6522 \newcommand*{\@glo@no@assign@sortkey}[1]{%
6523   \PackageError{glossaries}{‘sort’ key not permitted with
6524   \string\printglossary}%
6525   {The ‘sort’ key may only be used with \string\printnoidxglossary}%
6526 }
```

@assign@sortkey For use with \printnoidxglossary

```
6527 \newcommand*{\@glo@assign@sortkey}[1]{%
6528   \def\@glo@sorttype{#1}%
6529 }
```

\glslnonetpages 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 \glslnonetpages 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 re-defined.

```
6530 \newcommand*{\@glslnonetpages}{%
6531   \gdef\glossaryentrynumbers##1{%
6532     \glsresetentrylist
6533   }%
6534 }
```

\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 re-defined.

```
6535 \newcommand*{\@glsnextpages}{%
6536   \gdef\glossaryentrynumbers##1{%
6537     ##1\glsresetentrylist}}
```

glsresetentrylist Resets \glossaryentrynumbers

```
6538 \newcommand*{\glsresetentrylist}{%
6539   \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
```


`\glsnonextpages` Outside of `\printglossary` this does nothing.

```
6540 \newcommand*{\glsnonextpages}{}
```

`\glsnextpages` Outside of `\printglossary` this does nothing.

```
6541 \newcommand*{\glsnextpages}{}


```

Process `entrycounter` and then `subentrycounter` options (this ensures the sub-counter can pick up the main counter as the master if required):

```
6542 \@gls@define@glossaryentrycounter
```

```
6543 \@gls@define@glossarysubentrycounter


```

`subentrycounter` Resets the `glossarysubentry` counter.

```
6544 \newcommand*{\glsresetsubentrycounter}{%
6545   \ifglssubentrycounter
6546     \setcounter{glossarysubentry}{0}%
6547   \fi
6548 }
```

`subentrycounter` Resets the `glossaryentry` counter.

```
6549 \newcommand*{\glsresetentrycounter}{%
6550   \ifglsentrycounter
6551     \setcounter{glossaryentry}{0}%
6552   \fi
6553 }
```

`\glsstepentry` Advance the `glossaryentry` counter if in use. The argument is the label associated with the entry.

```
6554 \newcommand*{\glsstepentry}[1]{%
6555   \ifglsentrycounter
6556     \refstepcounter{glossaryentry}%
6557     \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
6558   \fi
6559 }
```

`glsstepsubentry` Advance the `glossarysubentry` counter if in use. The argument is the label associated with the subentry.

```
6560 \newcommand*{\glsstepsubentry}[1]{%
6561   \ifglssubentrycounter

6562     \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
6563     \refstepcounter{glossarysubentry}%
6564     \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
6565   \fi
6566 }
```

`\glsrefentry` Reference the entry or sub-entry counter if in use, otherwise just do `\gls`.

```
6567 \newcommand*{\glsrefentry}[1]{%
6568   \ifglsentrycounter
```

```

6569   \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
6570 \else
6571   \ifglssubentrycounter
6572     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
6573   \else
6574     \gls{#1}%
6575   \fi
6576 \fi
6577 }

```

trycounterlabel Defines how to display the glossaryentry counter.

```

6578 \newcommand*{\glssentrycounterlabel}{%
6579   \ifglssentrycounter
6580     \theglossaryentry.\space
6581   \fi
6582 }

```

trycounterlabel Defines how to display the glossarysubentry counter.

```

6583 \newcommand*{\glsssubentrycounterlabel}{%
6584   \ifglsssubentrycounter
6585     \theglossarysubentry)\space
6586   \fi
6587 }

```

\glssentryitem Step and display glossaryentry counter, if appropriate.

```

6588 \newcommand*{\glssentryitem}[1]{%
6589   \ifglssentrycounter
6590     \glsstepentry{#1}\glssentrycounterlabel
6591   \else
6592     \glsresetsubentrycounter
6593   \fi
6594 }

```

glsssubentryitem Step and display glossarysubentry counter, if appropriate.

```

6595 \newcommand*{\glsssubentryitem}[1]{%
6596   \ifglsssubentrycounter
6597     \glsstepsubentry{#1}\glsssubentrycounterlabel
6598   \fi
6599 }

```

theglossary If the theglossary environment has already been defined, a warning will be issued. This environment should be redefined by glossary styles.

```

6600 \ifcsundef{theglossary}%
6601 {%
6602   \newenvironment{theglossary}{}{}%
6603 }%
6604 {%
6605   \@gls@warnontheGLOSSdefined

```

```
6606 \renewenvironment{theglossary}{-}{-}%
6607 }
```

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`

```
6608 \newcommand*{\glossaryheader}{-}
```

`\glstarget`

```
\glstarget{<label>}{<name>}
```

Provide user interface to `\@glstarget` to make it easier to modify the glossary style in the document.

```
6609 \newcommand*{\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
```

As from version 3.08, glossary information is now written to the external files using `\glossentry` and `\subglossentry` instead of `\glossaryentryfield` and `\glossarysubentryfield`. The default definition provides backward compatibility for glossary styles that use the old forms.

`\compatibleglossentry`

```
\glossentry{<label>}{<page-list>}
```

```
6610 \providecommand*{\compatibleglossentry}[2]{%
6611   \toks0{#2}%
6612   \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
6613     {\noexpand\glsnamefont
6614       {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
6615     {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
6616     {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
6617     {\the\toks0}}%
6618   }%
6619   \@do@glossentry
6620 }
```

`\glossentryname`

```
6621 \newcommand*{\glossentryname}[1]{%
6622   \glsdoifexistsorwarn{#1}%
6623   {%
6624     \letcs{\glo@name}{\glo@glsdetoklabel{#1}@name}%
6625     \expandafter\glsnamefont\expandafter{\glo@name}%
6626   }%
6627 }
```

\Glossentryname

```
6628 \newcommand*{\Glossentryname}[1]{%
6629   \glsdoifexistsorwarn{#1}%
6630   {%
6631     \glsnamefont{\Glsentryname{#1}}%
6632   }%
6633 }
```

\glossentrydesc

```
6634 \newcommand*{\glossentrydesc}[1]{%
6635   \glsdoifexistsorwarn{#1}%
6636   {%
6637     \glsentrydesc{#1}%
6638   }%
6639 }
```

\Glossentrydesc

```
6640 \newcommand*{\Glossentrydesc}[1]{%
6641   \glsdoifexistsorwarn{#1}%
6642   {%
6643     \Glsentrydesc{#1}%
6644   }%
6645 }
```

\glossentrysymbol

```
6646 \newcommand*{\glossentrysymbol}[1]{%
6647   \glsdoifexistsorwarn{#1}%
6648   {%
6649     \glsentrysymbol{#1}%
6650   }%
6651 }
```

\Glossentrysymbol

```
6652 \newcommand*{\Glossentrysymbol}[1]{%
6653   \glsdoifexistsorwarn{#1}%
6654   {%
6655     \Glsentrysymbol{#1}%
6656   }%
6657 }
```

\lesubglossentry

`\subglossentry{<level>}{<label>}{<page-list>}`

```
6658 \providecommand*{\compatiblesubglossentry}[3]{%
6659   \toks@{#3}%
6660   \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
6661     {#2}%
6662     {\noexpand\glsnamefont
```

```

6663      {\expandafter\expandonce\csname glo@#2@name\endcsname}}}%
6664      {\expandafter\expandonce\csname glo@#2@desc\endcsname}}}%
6665      {\expandafter\expandonce\csname glo@#2@symbol\endcsname}}}%
6666      {\the\toks@}%
6667    }%
6668    \@do@subglossentry
6669  }

```

rycompatibility

```

6670 \newcommand*{\setglossentrycompatibility}{%
6671   \let\glossentry\compatibleglossentry
6672   \let\subglossentry\compatiblesubglossentry
6673 }
6674 \setglossentrycompatibility

```

ssaryentryfield

```

\glossaryentryfield{<label>}{<name>}{<description>}{<symbol>}{<page-list>}

```

This command formerly governed how each entry row should be formatted in the glossary. Now deprecated.

```

6675 \newcommand{\glossaryentryfield}[5]{%
6676   \GlossariesWarning
6677   {Deprecated use of \string\glossaryentryfield.^^J
6678     I recommend you change to \string\glossentry.^^J
6679     If you've just upgraded, try removing your gls auxiliary
6680     files^^J and recompile}%
6681   \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}

```

rysubentryfield

```

\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.)

```

6682 \newcommand*{\glossarysubentryfield}[6]{%
6683   \GlossariesWarning
6684   {Deprecated use of \string\glossarysubentryfield.^^J
6685     I recommend you change to \string\subglossentry.^^J
6686     If you've just upgraded, try removing your gls auxiliary
6687     files^^J and recompile}%
6688   \glstarget{#2}{\strut}#4. #6\par}

```

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`

```
6689 \newcommand*{\glsgroupskip}{}

```

Each of the 28 glossary groups described above is preceded by a group heading. This is formatted by the command `\glsgroupheading` which takes one argument which is the *label* assigned to that group (not the title). The corresponding labels are: `glssymbols`, `glsnumbers`, `A`, ..., `Z`. Glossary styles must redefine this command. (In between groups, `\glsgroupheading` comes immediately after `\glsgroupskip`.)

`\glsgroupheading`

```
6690 \newcommand*{\glsgroupheading}[1]{}

```

It is possible to “trick” `makeindex` into treating entries as though they belong to the same group, even if the terms don’t start with the same letter, by modifying the sort key. For example, all entries belonging to one group could be defined so that the sort key starts with an `a`, while entries belonging to another group could be defined so that the sort key starts with a `b`, and so on. If you want each group to have a heading, you would then need to modify the translation control sequences `\glsgetgrouptitle` and `\glsgetgrouplabel` so that the label is translated into the required title (and vice-versa).

```
\glsgetgrouptitle{<label>}
```

This command produces the title for the glossary group whose label is given by *<label>*. By default, the group labelled `glssymbols` produces `\glssymbolsgroupname`, the group labelled `glsnumbers` produces `\glsnumbersgroupname` and all the other groups simply produce their label. As mentioned above, the group labels are: `glssymbols`, `glsnumbers`, `A`, ..., `Z`. If you want to redefine the group titles, you will need to redefine this command. Languages other than English may produce labels that are non-expandable, so we need to check for that otherwise it will create a “missing `\endcsname` inserted” error.

`\glsgetgrouptitle`

```
6691 \newcommand*{\glsgetgrouptitle}[1]{%
6692   \@gls@getgrouptitle{#1}{\@gls@grptitle}%
6693   \@gls@grptitle
6694 }
```

`\@gls@getgrouptitle`

Gets the group title specified by the label (first argument) and stores in the second argument, which must be a control sequence.

```
6695 \newcommand*{\@gls@getgrouptitle}[2]{%

```

Even if the argument appears to be a single letter, it won’t be considered a single letter by `\dtl@ifsingle` if it’s an active character.

```

6696 \dtl@ifsingle{#1}%
6697 {%
6698   \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
6699 }%
6700 {%
6701   \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
6702               or test{\ifstrequal{#1}{glsnumbers}}}%
6703   {%
6704     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
6705   }%
6706   {%
6707     \def#2{#1}%
6708   }%
6709 }%
6710 }

```

`x@getgrouptitle` Version for the no-indexing app option:

```

6711 \newcommand*{\@gls@noidx@getgrouptitle}[2]{%
6712   \DTLifint{#1}%
6713   {\edef#2{\char#1\relax}}%
6714   {%
6715     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
6716   }%
6717 }

```

`\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`

```

6718 \newcommand*{\glsgetgrouplabel}[1]{%
6719 \ifthenelse{equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
6720 \ifthenelse{equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}

```

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`

```

6721 \newcommand*{\setentrycounter}[2] []{%
6722   \def\@glo@counterprefix{#1}%
6723   \ifx\@glo@counterprefix\@empty
6724     \def\@glo@counterprefix{.}%
6725   \else
6726     \def\@glo@counterprefix{.#1.}%
6727   \fi

```

```

6728 \def\glentrycounter{#2}%
6729 }

```

The current glossary style can be set using `\setglossarystyle{<style>}`.

`\setglossarystyle`

```

6730 \newcommand*{\setglossarystyle}[1]{%
6731   \ifcsundef{@glsstyle@#1}%
6732   {%
6733     \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
6734   }%
6735   {%
6736     \csname @glsstyle@#1\endcsname
6737   }%

```

Set the default style if it's not already set.

```

6738   \ifx\@glossary@default@style\relax
6739     \protected@edef\@glossary@default@style{#1}%
6740   \fi
6741 }

```

`\glossarystyle`

```

6742 \newcommand*{\glossarystyle}[1]{%
6743   \ifcsundef{@glsstyle@#1}%
6744   {%
6745     \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
6746   }%
6747   {%
6748     \GlossariesWarning
6749     {Deprecated command \string\glossarystyle.^^J
6750     I recommend you switch to \string\setglossarystyle\space unless
6751     you want to maintain backward compatibility}%
6752     \setglossentrycompatibility
6753     \csname @glsstyle@#1\endcsname

6754     \ifcsdef{@glscompstyle@#1}%
6755     {\setglossentrycompatibility\csuse{@glscompstyle@#1}}%
6756     {}%
6757   }%

```

Set the default style if it isn't already set so that `\printglossary` can warn if the fallback style is in use.

```

6758   \ifx\@glossary@default@style\relax
6759     \protected@edef\@glossary@default@style{#1}%
6760   \fi
6761 }

```

`\newglossarystyle` New glossary styles can be defined using:

`\newglossarystyle{<name>}{<definition>}`

The *<definition>* argument should redefine `\theglossary`, `\glossaryheader`, `\glsgroupheading`, `\glossaryentryfield` and `\glsgroupskip` (see [section 1.19](#) 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.

```
6762 \newcommand{\newglossarystyle}[2]{%
6763   \ifcsundef{@glsstyle@#1}%
6764   {%
6765     \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
6766   }%
6767   {%
6768     \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
6769   }%
6770 }
```

`\ewglossarystyle` Code for this macro supplied by Marco Daniel.

```
6771 \newcommand{\renewglossarystyle}[2]{%
6772   \ifcsundef{@glsstyle@#1}%
6773   {%
6774     \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
6775   }%
6776   {%
6777     \csdef{@glsstyle@#1}{#2}%
6778   }%
6779 }
```

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`

```
6780 \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 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 package.

`\glshypernumber`

```
6781 \ifcsundef{hyperlink}%
6782 {%
6783   \def\glshypernumber#1{#1}%
6784 }%
6785 {%
6786   \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
6787 }
```

`@glshypernumber` This code was provided by Heiko Oberdiek to allow material to be attached to the location.

```
6788 \def\@glshypernumber#1\nohyperpage#2#3\@nil{%
6789   \ifx\#1\%
6790     \else
6791       \@delimR#1\delimR\delimR\%
6792     \fi
6793     \ifx\#2\%
6794       \else
6795         #2%
6796       \fi
6797       \ifx\#3\%
6798         \else
6799           \@glshypernumber#3\@nil
6800         \fi
6801 }
```

`\@delimR` displays a range of numbers for the counter whose name is given by `\@gls@counter` (which must be set prior to using `\glshypernumber`).

`\@delimR`

```
6802 \def\@delimR#1\delimR #2\delimR #3\%
6803 \ifx\#2\%
6804   \@delimN{#1}%
6805 \else
6806   \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
6807 \fi}
```

`\@delimN` displays a list of individual numbers, instead of a range:

`\@delimN`

```
6808 \def\@delimN#1{\@delimN#1\delimN \delimN\%
6809 \def\@delimN#1\delimN #2\delimN#3\%
6810 \ifx\#3\%
6811   \@gls@numberlink{#1}%
6812 \else
6813   \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
6814 \fi
6815 }
```

The following code is modified from hyperref's \HyInd@pagelink where the name of the counter being used is given by \@gls@counter.

```

6816 \def\@gls@numberlink#1{%
6817 \begingroup
6818 \toks@={}%
6819 \@gls@removespaces#1 \@nil
6820 \endgroup}

6821 \def\@gls@removespaces#1 #2\@nil{%
6822 \toks@=\expandafter{\the\toks@#1}%
6823 \ifx\#2\%

6824 \edef\@gls@x{\the\toks@}%
6825 \ifx\@gls@x\empty
6826 \else

6827 \hyperlink{\glstrycounter\@glo@counterprefix\the\toks@}%
6828 {\the\toks@}%
6829 \fi
6830 \else
6831 \@gls@ReturnAfterFi{%
6832 \@gls@removespaces#2\@nil
6833 }%
6834 \fi
6835 }
6836 \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

```
6837 \newcommand*\hyperrm[1]{\textrm{\glshypernumber{#1}}}
```

\hypersf

```
6838 \newcommand*\hypersf[1]{\textsf{\glshypernumber{#1}}}
```

\hypertt

```
6839 \newcommand*\hypertt[1]{\texttt{\glshypernumber{#1}}}
```

\hyperbf

```
6840 \newcommand*\hyperbf[1]{\textbf{\glshypernumber{#1}}}
```

\hypermd

```
6841 \newcommand*\hypermd[1]{\textmd{\glshypernumber{#1}}}
```

\hyperit

```
6842 \newcommand*\hyperit[1]{\textit{\glshypernumber{#1}}}
```

`\hypersl`

```
6843 \newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
```

`\hyperup`

```
6844 \newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
```

`\hypersc`

```
6845 \newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
```

`\hyperemph`

```
6846 \newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}
```

1.17 Acronyms

`\oldacronym`

```
\oldacronym[⟨label⟩]{⟨abbrv⟩}{⟨long⟩}{⟨key-val list⟩}
```

This emulates the way the old 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 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 package is loaded. If 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 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 if desired.

```
6847 \newcommand{\oldacronym}[4][\gls@label]{%
6848   \def\gls@label{#2}%
6849   \newacronym[#4]{#1}{#2}{#3}%
6850   \ifcsundef{xspace}%
6851     {%
6852       \expandafter\edef\csname#1\endcsname{%
6853         \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
6854     }%
6855   }%
6856   {%
6857     \expandafter\edef\csname#1\endcsname{%
6858       \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
6859         \noexpand\gls{#1}\noexpand\xspace}%
6860     }%
6861   }%
6862 }
```

```
\newacronym[⟨key-val list⟩]{⟨label⟩}{⟨abbrev⟩}{⟨long⟩}
```

This is a quick way of defining acronyms, using `\newglossaryentry` with the appropriate values. It sets the glossary type to `\acronymtype` which will be acronym if the package option `acronym` has been used, otherwise it will be the default glossary. Since `\newacronym` merely calls `\newglossaryentry`, the acronym is treated like any other glossary entry.

If you prefer a different format, you can redefine `\newacronym` as required. The optional argument can be used to override any of the settings.

This is just a stub. It's redefined by commands like `\SetDefaultAcronymStyle`.

`\newacronym`

```
6863 \newcommand{\newacronym}[4] [] {}
```

Set up some convenient short cuts. These need to be changed if `\newacronym` is changed (or if the description key is changed).

`\acrpluralsuffix` Plural suffix used by `\newacronym`. This just defaults to `\glspluralsuffix` but is changed to include `\textup` if the `smallcaps` option is used, so that the suffix doesn't appear in small caps as it doesn't look right. For example, ABCS looks as though the "s" is part of the acronym, but ABCs looks as though the "s" is a plural suffix. Since the entire text `abcs` is set in `\textsc`, `\textup` is needed to cancel it out.

```
6864 \newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
```

If `garamondx` has been loaded, need to use `\textulc` instead of `\textup`.

`\glstextup`

```
6865 \newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
```

The following are defined for compatibility with version 2.07 and earlier.

`\glsshortkey`

```
6866 \newcommand*{\glsshortkey}{short}
```

`\glsshortpluralkey`

```
6867 \newcommand*{\glsshortpluralkey}{shortplural}
```

`\glslongkey`

```
6868 \newcommand*{\glslongkey}{long}
```

`\glslongpluralkey`

```
6869 \newcommand*{\glslongpluralkey}{longplural}
```

`\acrfull` Full form of the acronym.

```
6870 \newrobustcmd*{\acrfull}{\@gls@hyp@opt\@ns@acrfull}
```

```
6871 \newcommand*\@ns@acrfull[2] [] {%
```

```
6872 \new@ifnextchar[{\@acrfull{#1}{#2}}%
```

```
6873 \@acrfull{#1}{#2} []}%
```

```
6874 }
```

`\@acrfull` Low-level macro:

```
6875 \def\@acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
6876   \acrfullfmt{#1}{#2}{#3}%
6877 }
```

Using `\acrlinkfullformat` and `\acrfullformat` is now deprecated as it can cause complications with the first letter upper case variants, but the package needs to provide backward compatibility support.

`\acrfullfmt` No case change full format.

```
6878 \newcommand*\acrfullfmt[3]{%
6879   \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
6880 }
```

`\acrlinkfullformat` Format for full links like `\acrfull`. Syntax: `\acrlinkfullformat{<long cs>}{<short cs>}{<options>}{<label>}{<insert>}`

```
6881 \newcommand*\acrlinkfullformat[5]{%
6882   \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
6883 }
```

`\acrfullformat` Default full form is *<long>* (*<short>*).

```
6884 \newcommand*\acrfullformat[2]{#1\glsspace(#2)}
```

`\glsspace` Robust space to ensure it's written to the `.glsdefs` file.

```
6885 \newrobustcmd{\glsspace}{\space}
```

Default format for full acronym

`\Acrfull`

```
6886 \newrobustcmd*\Acrfull{\@gls@hyp@opt\ns@Acrfull}
6887 \newcommand*\ns@Acrfull[2][]{%
6888   \new@ifnextchar[\@Acrfull{#1}{#2}}%
6889   {\@Acrfull{#1}{#2}[]}%
6890 }
```

Low-level macro:

```
6891 \def\@Acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
6892   \Acrfullfmt{#1}{#2}{#3}%
6893 }
```

`\Acrfullfmt` First letter upper case full format.

```
6894 \newcommand*\Acrfullfmt[3]{%
6895   \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
6896 }
```

`\ACRfull`

```
6897 \newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
```

```
6898 \newcommand*\ns@ACRfull[2] [] {%
```

```
6899   \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
```

```
6900       {\@ACRfull{#1}{#2} []}%
```

```
6901 }
```

Low-level macro:

```
6902 \def\@ACRfull#1#2[#3] {%
```

Make it easier for acronym styles to change this:

```
6903   \ACRfullfmt{#1}{#2}{#3}%
```

```
6904 }
```

`\ACRfullfmt` All upper case full format.

```
6905 \newcommand*{\ACRfullfmt}[3] {%
```

```
6906   \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
```

```
6907 }
```

Plural:

`\acrfullpl`

```
6908 \newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
```

```
6909 \newcommand*\ns@acrfullpl[2] [] {%
```

```
6910   \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
```

```
6911       {\@acrfullpl{#1}{#2} []}%
```

```
6912 }
```

Low-level macro:

```
6913 \def\@acrfullpl#1#2[#3] {%
```

Make it easier for acronym styles to change this:

```
6914   \acrfullplfmt{#1}{#2}{#3}%
```

```
6915 }
```

`\acrfullplfmt` No case change plural full format.

```
6916 \newcommand*{\acrfullplfmt}[3] {%
```

```
6917   \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
```

```
6918 }
```

`\Acrfullpl`

```
6919 \newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
```

```
6920 \newcommand*\ns@Acrfullpl[2] [] {%
```

```
6921   \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
```

```
6922       {\@Acrfullpl{#1}{#2} []}%
```

```
6923 }
```

Low-level macro:

```
6924 \def\@Acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
6925 \Acrfullplfmt{#1}{#2}{#3}%
6926 }
```

`\Acrfullplfmt` First letter upper case plural full format.

```
6927 \newcommand*\@Acrfullplfmt}[3]{%
6928 \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
6929 }
```

`\ACRfullpl`

```
6930 \newrobustcmd*\@ACRfullpl{\@gls@hyp@opt\@ns@ACRfullpl}

6931 \newcommand*\@ns@ACRfullpl[2][ ]{%
6932 \new@ifnextchar[{\@ACRfullpl{#1}{#2}}]{%
6933 {\@ACRfullpl{#1}{#2}[ ]}%
6934 }
```

Low-level macro:

```
6935 \def\@ACRfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
6936 \ACRfullplfmt{#1}{#2}{#3}%
6937 }
```

`\ACRfullplfmt` All upper case plural full format.

```
6938 \newcommand*\@ACRfullplfmt}[3]{%
6939 \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
6940 }
```

1.18 Predefined acronym styles

`\acronymfont` This is only used with the additional acronym styles:

```
6941 \newcommand{\acronymfont}[1]{#1}
```

`\firstacronymfont` This is only used with the additional acronym styles:

```
6942 \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.

```
6943 \newcommand*\acrnameformat[2]{\acronymfont{#1}}
```

Define some tokens used by `\newacronym`:

`\glskeylisttok`

```
6944 \newtoks\glskeylisttok
```



```

\glslabeltok
6945 \newtoks\glslabeltok

\glsshorttok
6946 \newtoks\glsshorttok

\gslongtok
6947 \newtoks\gslongtok

\newacronymhook  Provide a hook for \newacronym:
6948 \newcommand*{\newacronymhook}{}

\genericNewAcronym  New improved version of setting the acronym style.
6949 \newcommand*{\SetGenericNewAcronym}{%
    Change the behaviour of \Glsentryname to workaround expansion issues that cause a prob-
    lem for \makefirstuc
6950 \let\@Gls@entryname\@Gls@acentryname
    Change the way acronyms are defined:
6951 \renewcommand{\newacronym}[4][\{%
6952 \ifdefempty{\@glsacronymlists}%
6953 {%
6954 \def\@glo@type{\acronymtype}%
6955 \setkeys{glossentry}{##1}%
6956 \DeclareAcronymList{\@glo@type}%
6957 }%
6958 }%
6959 \glskeylisttok{##1}%
6960 \glslabeltok{##2}%
6961 \glsshorttok{##3}%
6962 \gslongtok{##4}%
6963 \newacronymhook
6964 \protected@edef\@do@newglossaryentry{%
6965 \noexpand\newglossaryentry{\the\glslabeltok}%
6966 {%
6967 type=\acronymtype,%
6968 name={\expandonce{\acronymentry{##2}}},%
6969 sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
6970 text={\the\glsshorttok},%
6971 short={\the\glsshorttok},%
6972 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
6973 long={\the\gslongtok},%
6974 longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
6975 \GenericAcronymFields,%
6976 \the\glskeylisttok
6977 }%
6978 }%
6979 \@do@newglossaryentry
6980 }%

```

Make sure that `\acrfull` etc reflects the new style:

```

6981 \renewcommand*{\acrfullfmt}[3]{%
6982   \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
6983 \renewcommand*{\Acrfullfmt}[3]{%
6984   \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
6985 \renewcommand*{\ACRfullfmt}[3]{%
6986   \glslink[##1]{##2}{%
6987     \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}}}%
6988 \renewcommand*{\acrfullplfmt}[3]{%
6989   \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
6990 \renewcommand*{\Acrfullplfmt}[3]{%
6991   \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
6992 \renewcommand*{\ACRfullplfmt}[3]{%
6993   \glslink[##1]{##2}{%
6994     \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}}}%

```

Make sure that `\glstentryfull` etc reflects the new style:

```

6995 \renewcommand*{\glstentryfull}[1]{\genacrfullformat{##1}{}}%
6996 \renewcommand*{\Glstentryfull}[1]{\Genacrfullformat{##1}{}}%
6997 \renewcommand*{\glstentryfullpl}[1]{\genplacrfullformat{##1}{}}%
6998 \renewcommand*{\Glstentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
6999 }

```

`\GenericAcronymFields` Fields used by `\SetGenericNewAcronym` that can be changed by the acronym style.

```

7000 \newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}

```

`\acronymentry` `\acronymentry{<label>}`

Display style for the name field in the list of acronyms.

```

7001 \newcommand*{\acronymentry}[1]{\acronymfont{\glstentryshort{#1}}}

```

`\acronymsort` `\acronymsort{<short>}{<long>}`

Default sort format for acronyms.

```

7002 \newcommand*{\acronymsort}[2]{#1}

```

`\setacronymstyle` `\setacronymstyle{<style name>}`

```

7003 \newcommand*{\setacronymstyle}[1]{%
7004   \ifcsundef{@glsacr@dispstyle@#1}%
7005   {%
7006     \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
7007   }%
7008   {%
7009     \ifdefempty{\@glsacronymlists}%

```

```

7010   {%
7011     \DeclareAcronymList{\acronymtype}%
7012   }%
7013   {}%
7014   \SetGenericNewAcronym
7015   \GlsUseAcrStyleDefs{#1}%
7016   \@for\@gls@type:=\@glsacronymlists\do{%
7017     \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
7018   }%
7019   }%
7020 }

```

`\newacronymstyle`

`\newacronymstyle{<style name>}{<entry format definition>}{<display definitions>}`

Defines a new acronym style called *<style name>*.

```

7021 \newcommand*{\newacronymstyle}[3]{%
7022   \ifcsdef{@glsacr@dispstyle@#1}%
7023   {%
7024     \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
7025   }%
7026   {%
7027     \csdef{@glsacr@dispstyle@#1}{#2}%
7028     \csdef{@glsacr@styledefs@#1}{#3}%
7029   }%
7030 }

```

`\renewacronymstyle`

Redefines the given acronym style.

```

7031 \newcommand*{\renewacronymstyle}[3]{%
7032   \ifcsdef{@glsacr@dispstyle@#1}%
7033   {%
7034     \csdef{@glsacr@dispstyle@#1}{#2}%
7035     \csdef{@glsacr@styledefs@#1}{#3}%
7036   }%
7037   {%
7038     \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
7039   }%
7040 }

```

`\rEntryDispStyle`

```

7041 \newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}

```

`\UseAcrStyleDefs`

```

7042 \newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}

```

Predefined acronym styles:

long-short *<long>* (*<short>*) acronym style.

```
7043 \newacronymstyle{long-short}%
7044 {%
    Check for long form in case this is a mixed glossary.
7045   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
7046 }%
7047 {%
7048   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
7049   \renewcommand*{\genacrfullformat}[2]{%
7050     \gl Sentrylong{##1}##2\space
7051     (\protect\firstacronymfont{\gl Sentryshort{##1}})%
7052   }%
7053   \renewcommand*{\Genacrfullformat}[2]{%
7054     \Gl Sentrylong{##1}##2\space
7055     (\protect\firstacronymfont{\gl Sentryshort{##1}})%
7056   }%
7057   \renewcommand*{\genplacrfullformat}[2]{%
7058     \gl Sentrylongpl{##1}##2\space
7059     (\protect\firstacronymfont{\gl Sentryshortpl{##1}})%
7060   }%
7061   \renewcommand*{\Genplacrfullformat}[2]{%
7062     \Gl Sentrylongpl{##1}##2\space
7063     (\protect\firstacronymfont{\gl Sentryshortpl{##1}})%
7064   }%
7065   \renewcommand*{\acronymentry}[1]{\acronymfont{\gl Sentryshort{##1}}}%
7066   \renewcommand*{\acronymsort}[2]{##1}%
7067   \renewcommand*{\acronymfont}[1]{##1}%
7068   \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
7069   \renewcommand*{\acrpluralsuffix}{\gl spluralsuffix}%
7070 }
```

long-sp-short Similar to the previous style but allows the space between the long and short form to be customized.

```
7071 \newacronymstyle{long-sp-short}%
7072 {%
    Check for long form in case this is a mixed glossary.
7073   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
7074 }%
7075 {%
7076   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
7077   \renewcommand*{\genacrfullformat}[2]{%
7078     \gl Sentrylong{##1}##2\gl sacspace{##1}%
7079     (\protect\firstacronymfont{\gl Sentryshort{##1}})%
7080   }%
7081   \renewcommand*{\Genacrfullformat}[2]{%
7082     \Gl Sentrylong{##1}##2\gl sacspace{##1}%
7083     (\protect\firstacronymfont{\gl Sentryshort{##1}})%
7084   }%
7085 }
```

```

7085 \renewcommand*{\genplacrfullformat}[2]{%
7086   \glentrylongpl{##1}##2\glsacspace{##1}%
7087   (\protect\firstacronymfont{\glentryshortpl{##1}})%
7088 }%
7089 \renewcommand*{\Genplacrfullformat}[2]{%
7090   \Glsentrylongpl{##1}##2\glsacspace{##1}%
7091   (\protect\firstacronymfont{\glentryshortpl{##1}})%
7092 }%
7093 \renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
7094 \renewcommand*{\acronymsort}[2]{##1}%
7095 \renewcommand*{\acronymfont}[1]{##1}%
7096 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
7097 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
7098 }

```

`\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.

```

7099 \newcommand*{\glsacspace}[1]{%
7100   \settowidth{\dimen@}{(\firstacronymfont{\glentryshort{##1}})}%
7101   \ifdim\dimen@<3em~\else\space\fi
7102 }

```

`short-long` (*short*) (*long*) acronym style.

```

7103 \newacronymstyle{short-long}%
7104 {%
  Check for long form in case this is a mixed glossary.
7105   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
7106 }%
7107 {%
7108   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
7109   \renewcommand*{\genacrfullformat}[2]{%
7110     \protect\firstacronymfont{\glentryshort{##1}}##2\space
7111     (\glentrylong{##1})%
7112   }%
7113   \renewcommand*{\Genacrfullformat}[2]{%
7114     \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
7115     (\glentrylong{##1})%
7116   }%
7117   \renewcommand*{\genplacrfullformat}[2]{%
7118     \protect\firstacronymfont{\glentryshortpl{##1}}##2\space
7119     (\glentrylongpl{##1})%
7120   }%
7121   \renewcommand*{\Genplacrfullformat}[2]{%
7122     \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
7123     (\glentrylongpl{##1})%
7124   }%
7125   \renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
7126   \renewcommand*{\acronymsort}[2]{##1}%

```

```

7127 \renewcommand*{\acronymfont}[1]{##1}%
7128 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
7129 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
7130 }

```

long-sc-short *<long>* (\textsc{<short>}) acronym style.

```

7131 \newacronymstyle{long-sc-short}%
7132 {%
7133   \GlsUseAcrEntryDisplayStyle{long-short}%
7134 }%
7135 {%
7136   \GlsUseAcrStyleDefs{long-short}%
7137   \renewcommand{\acronymfont}[1]{\textsc{##1}}%
7138   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7139 }

```

long-sm-short *<long>* (\textsmaller{<short>}) acronym style.

```

7140 \newacronymstyle{long-sm-short}%
7141 {%
7142   \GlsUseAcrEntryDisplayStyle{long-short}%
7143 }%
7144 {%
7145   \GlsUseAcrStyleDefs{long-short}%
7146   \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
7147   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7148 }

```

sc-short-long *<short>* (\textsc{<long>}) acronym style.

```

7149 \newacronymstyle{sc-short-long}%
7150 {%
7151   \GlsUseAcrEntryDisplayStyle{short-long}%
7152 }%
7153 {%
7154   \GlsUseAcrStyleDefs{short-long}%
7155   \renewcommand{\acronymfont}[1]{\textsc{##1}}%
7156   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7157 }

```

sm-short-long *<short>* (\textsmaller{<long>}) acronym style.

```

7158 \newacronymstyle{sm-short-long}%
7159 {%
7160   \GlsUseAcrEntryDisplayStyle{short-long}%
7161 }%
7162 {%
7163   \GlsUseAcrStyleDefs{short-long}%
7164   \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
7165   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7166 }

```

long-short-desc *<long>* (*{<short>}*) acronym style that has an accompanying description (which the user needs to supply).

```

7167 \newacronymstyle{long-short-desc}%
7168 {%
7169   \GlsUseAcrEntryDisplayStyle{long-short}%
7170 }%
7171 {%
7172   \GlsUseAcrStyleDefs{long-short}%
7173   \renewcommand*{\GenericAcronymFields}{}%
7174   \renewcommand*{\acronymsort}[2]{##2}%
7175   \renewcommand*{\acronymentry}[1]{%
7176     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7177 }

```

g-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.

```

7178 \newacronymstyle{long-sp-short-desc}%
7179 {%
7180   \GlsUseAcrEntryDisplayStyle{long-sp-short}%
7181 }%
7182 {%
7183   \GlsUseAcrStyleDefs{long-sp-short}%
7184   \renewcommand*{\GenericAcronymFields}{}%
7185   \renewcommand*{\acronymsort}[2]{##2}%
7186   \renewcommand*{\acronymentry}[1]{%
7187     \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##1}})}%
7188 }

```

g-sc-short-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7189 \newacronymstyle{long-sc-short-desc}%
7190 {%
7191   \GlsUseAcrEntryDisplayStyle{long-sc-short}%
7192 }%
7193 {%
7194   \GlsUseAcrStyleDefs{long-sc-short}%
7195   \renewcommand*{\GenericAcronymFields}{}%
7196   \renewcommand*{\acronymsort}[2]{##2}%
7197   \renewcommand*{\acronymentry}[1]{%
7198     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7199 }

```

g-sm-short-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7200 \newacronymstyle{long-sm-short-desc}%
7201 {%
7202   \GlsUseAcrEntryDisplayStyle{long-sm-short}%
7203 }%

```

```

7204 {%
7205   \GlsUseAcrStyleDefs{long-sm-short}%
7206   \renewcommand*{\GenericAcronymFields}{}%
7207   \renewcommand*{\acronymsort}[2]{##2}%
7208   \renewcommand*{\acronymentry}[1]{%
7209     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7210 }

```

short-long-desc *<short>* (*<long>*) acronym style that has an accompanying description (which the user needs to supply).

```

7211 \newacronymstyle{short-long-desc}%
7212 {%
7213   \GlsUseAcrEntryDispStyle{short-long}%
7214 }%
7215 {%
7216   \GlsUseAcrStyleDefs{short-long}%
7217   \renewcommand*{\GenericAcronymFields}{}%
7218   \renewcommand*{\acronymsort}[2]{##2}%
7219   \renewcommand*{\acronymentry}[1]{%
7220     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7221 }

```

short-long-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7222 \newacronymstyle{sc-short-long-desc}%
7223 {%
7224   \GlsUseAcrEntryDispStyle{sc-short-long}%
7225 }%
7226 {%
7227   \GlsUseAcrStyleDefs{sc-short-long}%
7228   \renewcommand*{\GenericAcronymFields}{}%
7229   \renewcommand*{\acronymsort}[2]{##2}%
7230   \renewcommand*{\acronymentry}[1]{%
7231     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7232 }

```

short-long-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7233 \newacronymstyle{sm-short-long-desc}%
7234 {%
7235   \GlsUseAcrEntryDispStyle{sm-short-long}%
7236 }%
7237 {%
7238   \GlsUseAcrStyleDefs{sm-short-long}%
7239   \renewcommand*{\GenericAcronymFields}{}%
7240   \renewcommand*{\acronymsort}[2]{##2}%
7241   \renewcommand*{\acronymentry}[1]{%
7242     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7243 }

```


dua *<long>* only acronym style.

```
7244 \newacronymstyle{dua}%  
7245 {%
```

Check for long form in case this is a mixed glossary.

```
7246 \ifdefempty\glscustomtext  
7247 {%  
7248 \ifglshaslong{\glslabel}%  
7249 {%  
7250 \glsifplural  
7251 {%
```

Plural form:

```
7252 \glscapscase  
7253 {%
```

Plural form, don't adjust case:

```
7254 \glentrylongpl{\glslabel}\glsinsert  
7255 }%  
7256 {%
```

Plural form, make first letter upper case:

```
7257 \Glentrylongpl{\glslabel}\glsinsert  
7258 }%  
7259 {%
```

Plural form, all caps:

```
7260 \mfirstucMakeUppercase  
7261 {\glentrylongpl{\glslabel}\glsinsert}%  
7262 }%  
7263 }%  
7264 {%
```

Singular form

```
7265 \glscapscase  
7266 {%
```

Singular form, don't adjust case:

```
7267 \glentrylong{\glslabel}\glsinsert  
7268 }%  
7269 {%
```

Subsequent singular form, make first letter upper case:

```
7270 \Glentrylong{\glslabel}\glsinsert  
7271 }%  
7272 {%
```

Subsequent singular form, all caps:

```
7273 \mfirstucMakeUppercase  
7274 {\glentrylong{\glslabel}\glsinsert}%  
7275 }%  
7276 }%  
7277 }%  
7278 {%
```

Not an acronym:

```
7279     \glsgenentryfmt
7280     }%
7281     }%
7282     {\glscustomtext\glsinsert}%
7283 }%
7284 {%
7285   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

7286   \renewcommand*{\acrfullfmt}[3]{%
7287     \glslink[##1]{##2}{\glsentrylong{##2}##3\space
7288       (\acronymfont{\glsentryshort{##2}})}}%
7289   \renewcommand*{\Acrfullfmt}[3]{%
7290     \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
7291       (\acronymfont{\glsentryshort{##2}})}}%
7292   \renewcommand*{\ACRfullfmt}[3]{%
7293     \glslink[##1]{##2}{%
7294       \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
7295         (\acronymfont{\glsentryshort{##2}})}}}%

7296   \renewcommand*{\acrfullplfmt}[3]{%
7297     \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
7298       (\acronymfont{\glsentryshortpl{##2}})}}%

7299   \renewcommand*{\Acrfullplfmt}[3]{%
7300     \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
7301       (\acronymfont{\glsentryshortpl{##2}})}}%
7302   \renewcommand*{\ACRfullplfmt}[3]{%
7303     \glslink[##1]{##2}{%
7304       \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
7305         (\acronymfont{\glsentryshortpl{##2}})}}}%
7306   \renewcommand*{\glsentryfull}[1]{%
7307     \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
7308   }%
7309   \renewcommand*{\Glsentryfull}[1]{%
7310     \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
7311   }%
7312   \renewcommand*{\glsentryfullpl}[1]{%
7313     \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
7314   }%
7315   \renewcommand*{\Glsentryfullpl}[1]{%
7316     \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
7317   }%
7318   \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
7319   \renewcommand*{\acronymsort}[2]{##1}%
7320   \renewcommand*{\acronymfont}[1]{##1}%
7321   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7322 }
```

dua-desc *<long>* only acronym style with user-supplied description.

```
7323 \newacronymstyle{dua-desc}%
7324 {%
7325   \GlsUseAcrEntryDispStyle{dua}%
7326 }%
7327 {%
7328   \GlsUseAcrStyleDefs{dua}%
7329   \renewcommand*{\GenericAcronymFields}{}%

7330   \renewcommand*{\acronymentry}[1]{\acronymfont{\glentrylong{##1}}}%
7331   \renewcommand*{\acronymsort}[2]{##2}%
7332 }%
```

footnote *<short>*\footnote{*<long>*} acronym style.

```
7333 \newacronymstyle{footnote}%
7334 {%

  Check for long form in case this is a mixed glossary.

7335   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
7336 }%
7337 {%
7338   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

Need to ensure hyperlinks are switched off on first use:

```
7339 \glshyperfirstfalse
7340 \renewcommand*{\genacrfullformat}[2]{%
7341   \protect\firstacronymfont{\glentryshort{##1}}##2%
7342   \protect\footnote{\glentrylong{##1}}%
7343 }%
7344 \renewcommand*{\Genacrfullformat}[2]{%
7345   \firstacronymfont{\Glentryshort{##1}}##2%
7346   \protect\footnote{\glentrylong{##1}}%
7347 }%
7348 \renewcommand*{\genplacrfullformat}[2]{%
7349   \protect\firstacronymfont{\glentryshortpl{##1}}##2%
7350   \protect\footnote{\glentrylongpl{##1}}%
7351 }%
7352 \renewcommand*{\Genplacrfullformat}[2]{%
7353   \protect\firstacronymfont{\Glentryshortpl{##1}}##2%
7354   \protect\footnote{\glentrylongpl{##1}}%
7355 }%
7356 \renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
7357 \renewcommand*{\acronymsort}[2]{##1}%
7358 \renewcommand*{\acronymfont}[1]{##1}%
7359 \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
```

Don't use footnotes for \acrfull:

```
7360 \renewcommand*{\acrfullfmt}[3]{%
7361   \glslink[##1]{##2}{\acronymfont{\glentryshort{##2}}##3\space
7362   (\glentrylong{##2})}%
```

```

7363 \renewcommand*{\Acrfullfmt}[3]{%
7364   \glslink{##1}{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
7365     (\glsentrylong{##2})}%
7366 \renewcommand*{\ACRfullfmt}[3]{%
7367   \glslink{##1}{##2}{%
7368     \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
7369       (\glsentrylong{##2})}}}%
7370 \renewcommand*{\acrfullplfmt}[3]{%
7371   \glslink{##1}{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
7372     (\glsentrylongpl{##2})}%
7373 \renewcommand*{\Acrfullplfmt}[3]{%
7374   \glslink{##1}{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
7375     (\glsentrylongpl{##2})}%
7376 \renewcommand*{\ACRfullplfmt}[3]{%
7377   \glslink{##1}{##2}{%
7378     \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
7379       (\glsentrylongpl{##2})}}}%

```

Similarly for \glsentryfull etc:

```

7380 \renewcommand*{\glsentryfull}[1]{%
7381   \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
7382 \renewcommand*{\Glsentryfull}[1]{%
7383   \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
7384 \renewcommand*{\glsentryfullpl}[1]{%
7385   \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
7386 \renewcommand*{\Glsentryfullpl}[1]{%
7387   \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
7388 }

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

7389 \newacronymstyle{footnote-sc}%
7390 {%
7391   \GlsUseAcrEntryDisplayStyle{footnote}%
7392 }%
7393 {%
7394   \GlsUseAcrStyleDefs{footnote}%
7395   \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
7396   \renewcommand{\acronymfont}[1]{\textsc{##1}}%
7397   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7398 }%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```

7399 \newacronymstyle{footnote-sm}%
7400 {%
7401   \GlsUseAcrEntryDisplayStyle{footnote}%
7402 }%
7403 {%
7404   \GlsUseAcrStyleDefs{footnote}%
7405   \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
7406   \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%

```

```

7407 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7408 }%

```

footnote-desc *<short>*\footnote{*<long>*} acronym style that has an accompanying description (which the user needs to supply).

```

7409 \newacronymstyle{footnote-desc}%
7410 {%
7411 \GlsUseAcrEntryDisplayStyle{footnote}%
7412 }%
7413 {%
7414 \GlsUseAcrStyleDefs{footnote}%
7415 \renewcommand*{\GenericAcronymFields}{}%
7416 \renewcommand*{\acronymsort}[2]{##2}%
7417 \renewcommand*{\acronymentry}[1]{%
7418 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
7419 }

```

ootnote-sc-desc \textsc{*<short>*}\footnote{*<long>*} acronym style that has an accompanying description (which the user needs to supply).

```

7420 \newacronymstyle{footnote-sc-desc}%
7421 {%
7422 \GlsUseAcrEntryDisplayStyle{footnote-sc}%
7423 }%
7424 {%
7425 \GlsUseAcrStyleDefs{footnote-sc}%
7426 \renewcommand*{\GenericAcronymFields}{}%
7427 \renewcommand*{\acronymsort}[2]{##2}%
7428 \renewcommand*{\acronymentry}[1]{%
7429 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
7430 }

```

ootnote-sm-desc \textsmaller{*<short>*}\footnote{*<long>*} acronym style that has an accompanying description (which the user needs to supply).

```

7431 \newacronymstyle{footnote-sm-desc}%
7432 {%
7433 \GlsUseAcrEntryDisplayStyle{footnote-sm}%
7434 }%
7435 {%
7436 \GlsUseAcrStyleDefs{footnote-sm}%
7437 \renewcommand*{\GenericAcronymFields}{}%
7438 \renewcommand*{\acronymsort}[2]{##2}%
7439 \renewcommand*{\acronymentry}[1]{%
7440 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
7441 }

```

AcronymSynonyms

```

7442 \newcommand*{\DefineAcronymSynonyms}{%

```

Short form

`\acs`
 7443 `\let\acs\acrshort`
 First letter uppercase short form

`\Acs`
 7444 `\let\Acs\Acrshort`
 Plural short form

`\acsp`
 7445 `\let\acsp\acrshortpl`
 First letter uppercase plural short form

`\Acsp`
 7446 `\let\Acsp\Acrshortpl`
 Long form

`\acl`
 7447 `\let\acl\acrlong`
 Plural long form

`\aclp`
 7448 `\let\aclp\acrlongpl`
 First letter upper case long form

`\Acl`
 7449 `\let\Acl\Acrlong`
 First letter upper case plural long form

`\Aclp`
 7450 `\let\Aclp\Acrlongpl`
 Full form

`\acf`
 7451 `\let\acf\acrfull`
 Plural full form

`\acfp`
 7452 `\let\acfp\acrfullpl`
 First letter upper case full form

`\Acf`
 7453 `\let\Acf\Acrfull`

First letter upper case plural full form

`\Acfp`

```
7454 \let\Acfp\Acrfullpl
```

Standard form

`\ac`

```
7455 \let\ac\gls
```

First upper case standard form

`\Ac`

```
7456 \let\Ac\Gls
```

Standard plural form

`\acp`

```
7457 \let\acp\glspl
```

Standard first letter upper case plural form

`\Acp`

```
7458 \let\Acp\Glspl
```

```
7459 }
```

Define synonyms if required

```
7460 \ifglsacrshortcuts
```

```
7461 \DefineAcronymSynonyms
```

```
7462 \fi
```

These commands for setting the style are now deprecated but are kept for backward compatibility.

`\glsDisplayStyle` Sets the default acronym display style for given glossary.

```
7463 \newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
```

```
7464 \defglsentryfmt[#1]{\glsentryfmt}%
```

```
7465 }
```

`\glsNewAcronymDef` Sets up the acronym definition for the default style. The information is provided by the tokens `\glslabeltok`, `\glsshorttok`, `\gslongtok` and `\glskeylisttok`.

```
7466 \newcommand*{\DefaultNewAcronymDef}{%
```

```
7467 \edef\@do@newglossaryentry{%
```

```
7468 \noexpand\newglossaryentry{\the\glslabeltok}%
```

```
7469 {%
```

```
7470 type=\acronymtype,%
```

```
7471 name={\the\glsshorttok},%
```

```
7472 sort={\the\glsshorttok},%
```

```
7473 text={\the\glsshorttok},%
```

```
7474 first={\acrfullformat{\the\gslongtok}{\the\glsshorttok}},%
```

```
7475 plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
```

```

7476 firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
7477                {\noexpand\expandonce\noexpand\@glo@shortpl}},%
7478 short={\the\glsshorttok},%
7479 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7480 long={\the\glslongtok},%
7481 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7482 description={\the\glslongtok},%
7483 descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```

Remaining options specified by the user:

```

7484 \the\glskeylisttok
7485 }%
7486 }%
7487 \let\@org@gls@assign@firstpl\gls@assign@firstpl
7488 \let\@org@gls@assign@plural\gls@assign@plural
7489 \let\@org@gls@assign@descplural\gls@assign@descplural
7490 \def\gls@assign@firstpl##1##2{%
7491   \@@gls@expand@field{##1}{firstpl}{##2}%
7492 }%
7493 \def\gls@assign@plural##1##2{%
7494   \@@gls@expand@field{##1}{plural}{##2}%
7495 }%
7496 \def\gls@assign@descplural##1##2{%
7497   \@@gls@expand@field{##1}{descplural}{##2}%
7498 }%
7499 \do@newglossaryentry
7500 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7501 \let\gls@assign@plural\@org@gls@assign@plural
7502 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural

```

If expansion is switched off, the default plurals won't be properly set.

```

7503 \@@gls@noexpand@assign@default@plurals
7504 }

```

default@plurals

```

7505 \newcommand*{\@@gls@noexpand@assign@default@plurals}{%
7506   \expandafter\ifx\csname glo@\the\glslabeltok @plural\endcsname\@@gls@@default@acr@plural
7507   \csxdef{glo@\the\glslabeltok @plural}{\expandonce\@glo@shortpl}%
7508   \fi
7509   \expandafter\ifx\csname glo@\the\glslabeltok @firstpl\endcsname\@@gls@@default@acr@firstplura
7510   \csxdef{glo@\the\glslabeltok @firstpl}{%
7511     \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}}%
7512   \fi
7513   \expandafter\ifx\csname glo@\the\glslabeltok @descplural\endcsname\@@gls@@default@acr@descplu
7514   \csxdef{glo@\the\glslabeltok @descplural}{\expandonce\@glo@longpl}%
7515   \fi
7516 }

```

ault@acr@plural

```

7517 \newcommand*{\@@gls@@default@acr@plural}{\expandonce\@glo@shortpl}

```


acr@firstplural

```
7518 \newcommand*{\@@gls@@default@acr@firstplural}{%
7519 \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
7520 }
```

@acr@descplural

```
7521 \newcommand*{\@@gls@@default@acr@descplural}{\expandonce\@glo@longpl}
```

ultAcronymStyle Set up the default acronym style:

```
7522 \newcommand*{\SetDefaultAcronymStyle}{%
```

Set the display style:

```
7523 \for\@gls@type:=\@glsacronymlists\do{%
7524 \SetDefaultAcronymDisplayStyle{\@gls@type}%
7525 }%
```

Set up the definition of \newacronym:

```
7526 \renewcommand{\newacronym}[4][\]{%
```

If user is just using the main glossary and hasn't identified it as a list of acronyms, then update.
(This is done to ensure backwards compatibility with versions prior to 2.04).

```
7527 \ifx\@glsacronymlists\@empty
7528 \def\@glo@type{\acronymtype}%
7529 \setkeys{glossentry}{##1}%
7530 \DeclareAcronymList{\@glo@type}%
7531 \SetDefaultAcronymDisplayStyle{\@glo@type}%
7532 \fi
7533 \glskeylisttok{##1}%
7534 \glslabeltok{##2}%
7535 \glsshorttok{##3}%
7536 \glslongtok{##4}%
7537 \newacronymhook
7538 \DefaultNewAcronymDef
7539 }%
7540 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7541 }
```

\acrfootnote Used by the footnote acronym styles.

```
7542 \newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
```

acrlinkfootnote

```
7543 \newcommand*{\acrlinkfootnote}[3]{%
7544 \footnote{\glslink[#1]{#2}{#3}}%
7545 }
```

rnolinkfootnote

```
7546 \newcommand*{\acrlinkfootnote}[3]{%
7547 \footnote{#3}%
7548 }
```

`nymDisplayStyle` Sets the acronym display style for given glossary for the description and footnote combination.

```

7549 \newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
7550   \defglentryfmt[#1]{%

7551     \ifdefempty\glscustomtext
7552     {%
7553       \ifglused{\glslabel}%
7554       {%
7555         \acronymfont{\glsgenentryfmt}%
7556       }%
7557     {%
7558       \firstacronymfont{\glsgenentryfmt}%
7559       \ifglshassymbol{\glslabel}%
7560       {%
7561         \expandafter\protect\expandafter\acrfootnote\expandafter
7562         {\@gls@link@opts}{\@gls@link@label}%
7563         {%
7564           \glsifplural
7565             {\glsentrysymbolplural{\glslabel}}%
7566             {\glsentrysymbol{\glslabel}}%
7567           }%
7568         }%
7569       }%
7570     }%
7571     {\glscustomtext\glsinsert}%
7572   }%
7573 }
```

`teNewAcronymDef`

```

7574 \newcommand*{\DescriptionFootnoteNewAcronymDef}{%
7575   \edef\@do@newglossaryentry{%
7576     \noexpand\newglossaryentry{\the\glslabeltok}%
7577     {%
7578       type=\acronymtype,%
7579       name={\noexpand\acronymfont{\the\glsshorttok}},%
7580       sort={\the\glsshorttok},%
7581       first={\the\glsshorttok},%
7582       firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7583       text={\the\glsshorttok},%
7584       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7585       short={\the\glsshorttok},%
7586       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7587       long={\the\glslongtok},%
7588       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7589       symbol={\the\glslongtok},%
7590       symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7591       \the\glskeylisttok
7592     }%
7593   }
```

```

7593 }%
7594 \let\@org@gl@s@assign@firstpl\gl@s@assign@firstpl
7595 \let\@org@gl@s@assign@plural\gl@s@assign@plural
7596 \let\@org@gl@s@assign@symbolplural\gl@s@assign@symbolplural
7597 \def\gl@s@assign@firstpl##1##2{%
7598   \@@gl@s@expand@field{##1}{firstpl}{##2}%
7599 }%
7600 \def\gl@s@assign@plural##1##2{%
7601   \@@gl@s@expand@field{##1}{plural}{##2}%
7602 }%
7603 \def\gl@s@assign@symbolplural##1##2{%
7604   \@@gl@s@expand@field{##1}{symbolplural}{##2}%
7605 }%
7606 \do@newglossaryentry
7607 \let\gl@s@assign@plural\@org@gl@s@assign@plural
7608 \let\gl@s@assign@firstpl\@org@gl@s@assign@firstpl
7609 \let\gl@s@assign@symbolplural\@org@gl@s@assign@symbolplural
7610 }

```

oteAcronymStyle If a description and footnote are both required, store the long form in the symbol key. Store the short form in text key. Note that since the long form is stored in the symbol key, if you want the long form to appear in the list of acronyms, you need to use a glossary style that displays the symbol key. This style was deprecated with the introduction of `\newacronymstyle`.

```

7611 \newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
7612   \ifgl@sacrsmallcaps
7613     \@gl@s@deprecated@acr@warning{footnote,smallcaps,description}%
7614     {footnote-sc-desc}{short-sc-footnote-desc}%
7615   \else
7616     \ifgl@sacrsmaller
7617       \@gl@s@deprecated@acr@warning{footnote,smaller,description}%
7618       {footnote-sm-desc}{short-sm-footnote-desc}%
7619     \else
7620       \@gl@s@deprecated@acr@warning{footnote,description}%
7621       {footnote-desc}{short-footnote-desc}%
7622     \fi
7623   \fi
7624   \renewcommand{\newacronym}[4][\]{%
7625     \ifx\@gl@sacronymlists\@empty
7626       \def\@glo@type{\acronymtype}%
7627       \setkeys{glossentry}{##1}%
7628       \DeclareAcronymList{\@glo@type}%
7629       \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
7630     \fi
7631     \gl@skeylisttok{##1}%
7632     \gl@slabeltok{##2}%
7633     \gl@sshorttok{##3}%
7634     \gl@slongtok{##4}%
7635     \newacronymhook
7636     \DescriptionFootnoteNewAcronymDef

```

7637 }%

If footnote package option is specified, set the first use to append the long form (stored in symbol) as a footnote.

```
7638 \@for\@gls@type:=\@glsacronymlists\do{%
7639   \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
7640 }%
```

Redefine \acronymfont if small caps required. The plural suffix is set in an upright font so that it remains in normal lower case, otherwise it looks as though it's part of the acronym.

```
7641 \ifglsacrsmallcaps
7642   \renewcommand*\acronymfont}[1]{\textsc{##1}}%
7643   \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
7644 \else
7645   \ifglsacrsmaller
7646     \renewcommand*\acronymfont}[1]{\textsmaller{##1}}%
7647   \fi
7648 \fi
```

Check for package option clash

```
7649 \ifglsacrdua
7650   \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
7651     can’t both be set}{}%
7652 \fi
7653 }%
```

nymDisplayStyle Sets the acronym display style for given glossary with description and dua combination.

```
7654 \newcommand*\SetDescriptionDUAAcronymDisplayStyle}[1]{%
7655   \defglsentryfmt[#1]{\glsngenentryfmt}%
7656 }
```

UANewAcronymDef

```
7657 \newcommand*\DescriptionDUANewAcronymDef{%
7658   \edef\@do@newglossaryentry{%
7659     \noexpand\newglossaryentry{\the\glslabeltok}%
7660     {%
7661       type=\acronymtype,%
7662       name={\the\glslongtok},%
7663       sort={\the\glslongtok},%
7664       text={\the\glslongtok},%
7665       first={\the\glslongtok},%
7666       plural={\noexpand\expandonce\noexpand\@glo@longpl},%
7667       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7668       short={\the\glsshorttok},%
7669       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7670       long={\the\glslongtok},%
7671       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7672       symbol={\the\glsshorttok},%
7673       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
```

```

7674     \the\glskeylisttok
7675 }%
7676 }%
7677 \let\@org@gls@assign@firstpl\gls@assign@firstpl
7678 \let\@org@gls@assign@plural\gls@assign@plural
7679 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
7680 \def\gls@assign@firstpl##1##2{%
7681   \@@gls@expand@field{##1}{firstpl}{##2}%
7682 }%
7683 \def\gls@assign@plural##1##2{%
7684   \@@gls@expand@field{##1}{plural}{##2}%
7685 }%
7686 \def\gls@assign@symbolplural##1##2{%
7687   \@@gls@expand@field{##1}{symbolplural}{##2}%
7688 }%
7689 \do@newglossaryentry
7690 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7691 \let\gls@assign@plural\@org@gls@assign@plural
7692 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
7693 }

```

DUAACronymStyle Description, don't use acronym and no footnote. Note that the short form is stored in the symbol key, so if the short form needs to be displayed in the glossary, use a style the displays the symbol. This style was deprecated with the introduction of `\newacronymstyle`.

```

7694 \newcommand*{\SetDescriptionDUAACronymStyle}{%
7695   \ifglsacrsmallcaps
7696     \PackageError{glossaries}{Option clash: 'smallcaps' and 'dua'
7697       can't both be set}{}%
7698   \else
7699     \ifglsacrsmaller
7700       \PackageError{glossaries}{Option clash: 'smaller' and 'dua'
7701         can't both be set}{}%
7702     \fi
7703   \fi
7704   \@gls@deprecated@acr@warning{dua,description}{dua-desc}{long-noshort-desc}%
7705   \renewcommand{\newacronym}[4][]{%
7706     \ifx\@glsacronymlists\@empty
7707       \def\@glo@type{\acronymtype}%
7708       \setkeys{glossentry}{##1}%
7709       \DeclareAcronymList{\@glo@type}%
7710       \SetDescriptionDUAACronymDisplayStyle{\@glo@type}%
7711     \fi
7712     \glskeylisttok{##1}%
7713     \glslabeltok{##2}%
7714     \glsshorttok{##3}%
7715     \glslongtok{##4}%
7716     \newacronymhook
7717     \DescriptionDUANewAcronymDef
7718   }%

```

Set display.

```
7719 \for\@gls@type:=\@glsacronymlists\do{%
7720   \SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
7721 }%
7722 }%
```

`\acronymDisplayStyle` Sets the acronym display style for given glossary using the description setting (but not footnote or dua).

```
7723 \newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
7724   \def\glsentryfmt[#1]{%

7725     \ifdefempty\glscustomtext
7726     {%
7727       \ifglsused{\glslabel}%
7728       {%
```

Move the inserted text outside of `\acronymfont`

```
7729       \let\gls@org@insert\glsinsert
7730       \let\glsinsert\@empty
7731       \acronymfont{\glsentryfmt}\gls@org@insert
7732     }%
7733     {%
7734       \glsentryfmt
7735       \ifgls hassymbol{\glslabel}%
7736       {%
7737         \glsifplural
7738         {%
7739           \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
7740         }%
7741         {%
7742           \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
7743         }%
7744         \space(\protect\firstacronymfont
7745           {\glscapscase
7746             {\@glo@symbol}
7747             {\@glo@symbol}
7748             {\mfirstucMakeUppercase{\@glo@symbol}}})%
7749       }%
7750     }%
7751   }%
7752 }%
7753 {\glscustomtext\glsinsert}%
7754 }%
7755 }
```

`\onNewAcronymDef`

```
7756 \newcommand*{\DescriptionNewAcronymDef}{%
7757   \edef\@do@newglossaryentry{%
7758     \noexpand\newglossaryentry{\the\glslabeltok}%
```

```

7759 {%
7760     type=\acronymtype,%
7761     name={\noexpand
7762         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
7763     sort={\the\glsshorttok},%
7764     first={\the\glslongtok},%
7765     firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7766     text={\the\glsshorttok},%
7767     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7768     short={\the\glsshorttok},%
7769     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7770     long={\the\glslongtok},%
7771     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7772     symbol={\noexpand\@glo@text},%
7773     symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7774     \the\glskeylisttok}%
7775 }%
7776 \let\@org@gls@assign@firstpl\gls@assign@firstpl
7777 \let\@org@gls@assign@plural\gls@assign@plural
7778 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
7779 \def\gls@assign@firstpl##1##2{%
7780     \@gls@expand@field{##1}{firstpl}{##2}%
7781 }%
7782 \def\gls@assign@plural##1##2{%
7783     \@gls@expand@field{##1}{plural}{##2}%
7784 }%
7785 \def\gls@assign@symbolplural##1##2{%
7786     \@gls@expand@field{##1}{symbolplural}{##2}%
7787 }%
7788 \do@newglossaryentry
7789 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7790 \let\gls@assign@plural\@org@gls@assign@plural
7791 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
7792 }

```

ionAcronymStyle Option description is used, but not dua or footnote. Store long form in first key and short form in text and symbol key. The name is stored using \acrnameformat to allow the user to override the way the name is displayed in the list of acronyms.

```

7793 \newcommand*{\SetDescriptionAcronymStyle}{%
7794     \ifglsacrsmallcaps
7795         \@gls@deprecated@acr@warning{smallcaps,description}%
7796         {long-sc-short-desc}{long-short-sc-desc}%
7797     \else
7798         \ifglsacrsmaller
7799             \@gls@deprecated@acr@warning{smaller,description}%
7800             {long-sm-short-desc}{long-short-sm-desc}%
7801         \else
7802             \@gls@deprecated@acr@warning{description}%
7803             {long-short-desc}{long-short-desc}%

```

```

7804 \fi
7805 \fi
7806 \renewcommand{\newacronym}[4][\%
7807 \ifx\@glsacronymlists\@empty
7808 \def\@glo@type{\acronymtype}%
7809 \setkeys{glossentry}{##1}%
7810 \DeclareAcronymList{\@glo@type}%
7811 \SetDescriptionAcronymDisplayStyle{\@glo@type}%
7812 \fi
7813 \glskeylisttok{##1}%
7814 \glslabeltok{##2}%
7815 \glsshorttok{##3}%
7816 \glslongtok{##4}%
7817 \newacronymhook
7818 \DescriptionNewAcronymDef
7819 }%

```

Set display.

```

7820 \@for\@gls@type:=\@glsacronymlists\do{%
7821 \SetDescriptionAcronymDisplayStyle{\@gls@type}%
7822 }%

```

Redefine `\acronymfont` if small caps required. The plural suffix is set in an upright font so that it remains in normal lower case, otherwise it looks as though it's part of the acronym.

```

7823 \ifglsacrsmallcaps
7824 \renewcommand{\acronymfont}[1]{\textsc{##1}}
7825 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7826 \else
7827 \ifglsacrsmaller
7828 \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
7829 \fi
7830 \fi
7831 }%

```

`\acronymDisplayStyle` Sets the acronym display style for given glossary with footnote setting (but not description or dua).

```

7832 \newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
7833 \defglsentryfmt[##1]{%
7834 \ifdefempty\glscustomtext
7835 {%

```

Move the inserted text outside of `\acronymfont`

```

7836 \let\gls@org@insert\glsinsert
7837 \let\glsinsert\@empty
7838 \ifglsused{\glslabel}%
7839 {%
7840 \acronymfont{\glsgenentryfmt}\gls@org@insert
7841 }%
7842 {%

```



```

7843     \firstacronymfont{\glsgenentryfmt}\gls@org@insert
7844     \ifglshaslong{\glslabel}%
7845     {%
7846         \expandafter\protect\expandafter\acrfootnote\expandafter
7847         {\@gls@link@opts}{\@gls@link@label}%
7848         {%
7849             \glsifplural
7850             {\glsentrylongpl{\glslabel}}%
7851             {\glsentrylong{\glslabel}}%
7852         }%
7853     }%

7854     {%
7855     }%
7856 }%
7857 {\glscustomtext\glsinsert}%
7858 }%
7859 }

```

teNewAcronymDef

```

7860 \newcommand*{\FootnoteNewAcronymDef}{%
7861     \edef\@do@newglossaryentry{%
7862         \noexpand\newglossaryentry{\the\glslabeltok}%
7863         {%
7864             type=\acronymtype,%
7865             name={\noexpand\acronymfont{\the\glsshorttok}},%
7866             sort={\the\glsshorttok},%
7867             text={\the\glsshorttok},%
7868             plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7869             first={\the\glsshorttok},%
7870             firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7871             short={\the\glsshorttok},%
7872             shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7873             long={\the\glslongtok},%
7874             longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7875             description={\the\glslongtok},%
7876             descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7877             \the\glskeylisttok
7878         }%
7879     }%
7880     \let\@org@gls@assign@plural\gls@assign@plural
7881     \let\@org@gls@assign@firstpl\gls@assign@firstpl
7882     \let\@org@gls@assign@descplural\gls@assign@descplural
7883     \def\gls@assign@firstpl##1##2{%
7884         \@gls@expand@field{##1}{firstpl}{##2}%
7885     }%
7886     \def\gls@assign@plural##1##2{%
7887         \@gls@expand@field{##1}{plural}{##2}%
7888     }%
7889     \def\gls@assign@descplural##1##2{%

```

```

7890 \@@gls@expand@field{##1}{descplural}{##2}%
7891 }%
7892 \do@newglossaryentry
7893 \let\gls@assign@plural\@org@gls@assign@plural
7894 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7895 \let\gls@assign@descplural\@org@gls@assign@descplural
7896 }

```

`oteAcronymStyle` 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`.

```

7897 \newcommand*{\SetFootnoteAcronymStyle}{%
7898 \ifglsacrsmallcaps
7899 \@gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
7900 \else
7901 \ifglsacrsmaller
7902 \@gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
7903 \else
7904 \@gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
7905 \fi
7906 \fi
7907 \renewcommand{\newacronym}[4][\]{%
7908 \ifx\@glsacronymlists\@empty
7909 \def\@glo@type{\acronymtype}%
7910 \setkeys{glossentry}{##1}%
7911 \DeclareAcronymList{\@glo@type}%
7912 \SetFootnoteAcronymDisplayStyle{\@glo@type}%
7913 \fi
7914 \glskeylisttok{##1}%
7915 \glslabeltok{##2}%
7916 \glsshorttok{##3}%
7917 \glslongtok{##4}%
7918 \newacronymhook
7919 \FootnoteNewAcronymDef
7920 }%

```

Set display

```

7921 \@for\@gls@type:=\@glsacronymlists\do{%
7922 \SetFootnoteAcronymDisplayStyle{\@gls@type}%
7923 }%

```

Redefine `\acronymfont` if small caps required. The plural suffix is set in an upright font so that it remains in normal lower case, otherwise it looks as though it's part of the acronym.

```

7924 \ifglsacrsmallcaps
7925 \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
7926 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7927 \else
7928 \ifglsacrsmaller
7929 \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
7930 \fi

```

7931 \fi

Check for option clash

7932 \ifglsacrdua

7933 \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’

7934 can’t both be set}{}%

7935 \fi

7936 }%

parenifnotempty 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.

7937 \DeclareRobustCommand*\glsdoparenifnotempty}[2]{%

7938 \protected@edef\gls@tmp{#1}%

7939 \ifdefempty\gls@tmp

7940 {}%

7941 {%

7942 \ifx\gls@tmp\@gls@default@value

7943 \else

7944 \space (#2{#1})%

7945 \fi

7946 }%

7947 }

nymDisplayStyle Sets the acronym display style for given glossary where neither footnote nor description is required, but smallcaps or smaller specified.

7948 \newcommand*\SetSmallAcronymDisplayStyle}[1]{%

7949 \defglsentryfmt[#1]{%

7950 \ifdefempty\glscustomtext

7951 {%

Move the inserted text outside of \acronymfont

7952 \let\gls@org@insert\glsinsert

7953 \let\glsinsert\@empty

7954 \ifglsused{\glslabel}%

7955 {%

7956 \acronymfont{\glsgenentryfmt}\gls@org@insert

7957 }%

7958 {%

7959 \glsgenentryfmt

7960 \ifgls hassymbol{\glslabel}%

7961 {%

7962 \glsifplural

7963 {%

7964 \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%

7965 }%

7966 {%

7967 \def\@glo@symbol{\glsentrysymbol{\glslabel}}%

7968 }%

7969 \space

```

7970      (\glscapscase
7971      {\firstacronymfont{\@glo@symbol}}}%
7972      {\firstacronymfont{\@glo@symbol}}}%
7973      {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
7974      }%
7975      {}%
7976      }%
7977      }%
7978      {\glscustomtext\glsinsert}%
7979      }%
7980 }

```

1.1NewAcronymDef

```

7981 \newcommand*\SmallNewAcronymDef{%
7982   \edef\@do@newglossaryentry{%
7983     \noexpand\newglossaryentry{\the\glslabeltok}%
7984     {%
7985       type=\acronymtype,%
7986       name={\noexpand\acronymfont{\the\glsshorttok}},%
7987       sort={\the\glsshorttok},%
7988       text={\the\glsshorttok},%
7989       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7990       first={\the\glslongtok},%
7991       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7992       short={\the\glsshorttok},%
7993       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7994       long={\the\glslongtok},%
7995       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7996       description={\noexpand\@glo@first},%
7997       descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7998       symbol={\the\glsshorttok},%
7999       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
8000       \the\glskeylisttok
8001     }%
8002   }%
8003   \let\@org@gls@assign@firstpl\gls@assign@firstpl
8004   \let\@org@gls@assign@plural\gls@assign@plural
8005   \let\@org@gls@assign@descplural\gls@assign@descplural
8006   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
8007   \def\gls@assign@firstpl##1##2{%
8008     \@@gls@expand@field{##1}{firstpl}{##2}%
8009   }%
8010   \def\gls@assign@plural##1##2{%
8011     \@@gls@expand@field{##1}{plural}{##2}%

```

```

8012 }%
8013 \def\gls@assign@descplural##1##2{%
8014   \@gls@expand@field{##1}{descplural}{##2}%
8015 }%
8016 \def\gls@assign@symbolplural##1##2{%
8017   \@gls@expand@field{##1}{symbolplural}{##2}%
8018 }%
8019 \@do@newglossaryentry
8020 \let\gls@assign@firstpl\@org@gls@assign@firstpl
8021 \let\gls@assign@plural\@org@gls@assign@plural
8022 \let\gls@assign@descplural\@org@gls@assign@descplural
8023 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
8024 }

```

`allAcronymStyle` 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`.

```

8025 \newcommand*{\SetSmallAcronymStyle}{%
8026   \ifglsacrsmallcaps
8027     \@gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
8028   \else
8029     \@gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
8030   \fi
8031   \renewcommand{\newacronym}[4][1]{%
8032     \ifx\@glsacronymlists\@empty
8033       \def\@glo@type{\acronymtype}%
8034       \setkeys{glossentry}{##1}%
8035       \DeclareAcronymList{\@glo@type}%
8036       \SetSmallAcronymDisplayStyle{\@glo@type}%
8037     \fi
8038     \glskeylisttok{##1}%
8039     \glslabeltok{##2}%
8040     \glsshorttok{##3}%
8041     \glslongtok{##4}%
8042     \newacronymhook
8043     \SmallNewAcronymDef
8044   }%

```

Change the display since first only contains long form.

```

8045 \@for\@gls@type:=\@glsacronymlists\do{%
8046   \SetSmallAcronymDisplayStyle{\@gls@type}%
8047 }%

```

Redefine `\acronymfont` if small caps required. The plural suffix is set in an upright font so that it remains in normal lower case, otherwise it looks as though it's part of the acronym.

```

8048 \ifglsacrsmallcaps
8049   \renewcommand*{\acronymfont}[1]{\textsc{##1}}
8050   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
8051 \else
8052   \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}

```

```

8053 \fi
      check for option clash
8054 \ifglsacrdua
8055   \ifglsacrsmallcaps
8056     \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
8057       can’t both be set}{}%
8058   \else
8059     \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
8060       can’t both be set}{}%
8061 \fi
8062 \fi
8063 }%

```

DUADisplayStyle Sets the acronym display style for given glossary with dua setting.

```

8064 \newcommand*{\SetDUADisplayStyle}[1]{%
8065   \defglsentryfmt[#1]{\glsentryfmt}%
8066 }

```

UANewAcronymDef

```

8067 \newcommand*{\DUANewAcronymDef}{%
8068   \edef\@do@newglossaryentry{%
8069     \noexpand\newglossaryentry{\the\glslabeltok}%
8070     {%
8071       type=\acronymtype,%
8072       name={\the\glsshorttok},%
8073       text={\the\glslongtok},%
8074       first={\the\glslongtok},%
8075       plural={\noexpand\expandonce\noexpand\@glo@longpl},%
8076       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
8077       short={\the\glsshorttok},%
8078       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
8079       long={\the\glslongtok},%
8080       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
8081       description={\the\glslongtok},%
8082       descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
8083       symbol={\the\glsshorttok},%
8084       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
8085       \the\glskeylisttok
8086     }%
8087   }%
8088   \let\@org@gls@assign@firstpl\gls@assign@firstpl
8089   \let\@org@gls@assign@plural\gls@assign@plural
8090   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
8091   \let\@org@gls@assign@descplural\gls@assign@descplural
8092   \def\gls@assign@firstpl##1##2{%
8093     \@@gls@expand@field{##1}{firstpl}{##2}%
8094   }%
8095   \def\gls@assign@plural##1##2{%
8096     \@@gls@expand@field{##1}{plural}{##2}%

```

```

8097 }%
8098 \def\gls@assign@symbolplural##1##2{%
8099   \@gls@expand@field{##1}{symbolplural}{##2}%
8100 }%
8101 \def\gls@assign@descplural##1##2{%
8102   \@gls@expand@field{##1}{descplural}{##2}%
8103 }%
8104 \do@newglossaryentry
8105 \let\gls@assign@firstpl\@org@gls@assign@firstpl
8106 \let\gls@assign@plural\@org@gls@assign@plural
8107 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
8108 \let\gls@assign@descplural\@org@gls@assign@descplural
8109 }

```

`\SetDUASyle` Always expand acronyms. This style was deprecated with the introduction of `\newacronymstyle`.

```

8110 \newcommand*{\SetDUASyle}{%
8111   \@gls@deprecated@acr@warning{dua}{dua}{long-noshort}%
8112   \renewcommand{\newacronym}[4][]{%
8113     \ifx\@glsacronymlists\@empty
8114       \def\@glo@type{\acronymtype}%
8115       \setkeys{glossentry}{##1}%
8116       \DeclareAcronymList{\@glo@type}%
8117       \SetDUADisplayStyle{\@glo@type}%
8118     \fi
8119     \glskeylisttok{##1}%
8120     \glslabeltok{##2}%
8121     \glsshorttok{##3}%
8122     \glslongtok{##4}%
8123     \newacronymhook
8124     \DUANewAcronymDef
8125   }%

```

Set the display

```

8126   \@for\@gls@type:=\@glsacronymlists\do{%
8127     \SetDUADisplayStyle{\@gls@type}%
8128   }%
8129 }

```

`ted@acr@warning`

```

8130 \newcommand*{\@gls@deprecated@acr@warning}[3]{%
8131   \GlossariesWarning{Acronym style option #1 deprecated.
8132   Use \string\setacronymstyle{#2} with just glossaries.sty or
8133   use \string\setabbreviationstyle[acronym]{#3} with
8134   glossaries-extra.sty}%
8135 }

```

`SetAcronymStyle`

```

8136 \newcommand*{\SetAcronymStyle}{%

```

```

8137 \SetDefaultAcronymStyle
8138 \ifglssacrdescription
8139   \ifglssacrfootnote
8140     \SetDescriptionFootnoteAcronymStyle
8141   \else
8142     \ifglssacrdua
8143       \SetDescriptionDUAAcronymStyle
8144     \else
8145       \SetDescriptionAcronymStyle
8146     \fi
8147   \fi
8148 \else
8149   \ifglssacrfootnote
8150     \SetFootnoteAcronymStyle
8151   \else
8152     \ifthenelse{\boolean{glssacrsmalls}}{\OR
8153       \boolean{glssacrsmalls}}{%
8154       {%
8155         \SetSmallAcronymStyle
8156       }%
8157     {%
8158       \ifglssacrdua
8159         \SetDUASyle
8160       \fi
8161     }%
8162   \fi
8163 \fi
8164 }

```

Set the acronym style according to the package options

```
8165 \SetAcronymStyle
```

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.

`\setacronymdisplaystyle` Sets the acronym display style.

```

8166 \newcommand*{\SetCustomDisplayStyle}[1]{%
8167   \defglssentryfmt[#1]{\glsgenentryfmt}%
8168 }

```

`\setacronymfields`

```

8169 \newcommand*{\CustomAcronymFields}{%
8170   name={\the\glsshorttok},%
8171   description={\the\glslongtok},%
8172   first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
8173   firstplural={\acrfullformat
8174     {\noexpand\glsgenentrylongpl{\the\glslabeltok}}}%
8175     {\noexpand\glsgenentryshortpl{\the\glslabeltok}}},%

```



```

8176 text={\the\glsshorttok},%
8177 plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
8178 }

```

omNewAcronymDef

```

8179 \newcommand*{\CustomNewAcronymDef}{%
8180   \protected@edef\@do@newglossaryentry{%
8181     \noexpand\newglossaryentry{\the\glslabeltok}%
8182     {%
8183       type=\acronymtype,%
8184       short={\the\glsshorttok},%
8185       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
8186       long={\the\glslongtok},%
8187       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
8188       user1={\the\glsshorttok},%
8189       user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
8190       user3={\the\glslongtok},%
8191       user4={\the\glslongtok\noexpand\acrpluralsuffix},%
8192       \CustomAcronymFields,%
8193       \the\glskeylisttok
8194     }%
8195   }%
8196   \@do@newglossaryentry
8197 }

```

\SetCustomStyle

```

8198 \newcommand*{\SetCustomStyle}{%
8199   \renewcommand{\newacronym}[4][\fi]{%
8200     \ifx\@glsacronymlists\@empty
8201       \def\@glo@type{\acronymtype}%
8202       \setkeys{glossentry}{##1}%
8203       \DeclareAcronymList{\@glo@type}%
8204       \SetCustomDisplayStyle{\@glo@type}%
8205     \fi
8206     \glskeylisttok{##1}%
8207     \glslabeltok{##2}%
8208     \glsshorttok{##3}%
8209     \glslongtok{##4}%
8210     \newacronymhook
8211     \CustomNewAcronymDef
8212   }%
8213   Set the display
8214   \@for\@gls@type:=\@glsacronymlists\do{%
8215     \SetCustomDisplayStyle{\@gls@type}%
8216   }%
8217 }

```

1.19 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.

```
8217 \RequirePackage{glossary-hypernav}
```

The styles that use list-like environments. These are not loaded if the nolist option is used:

```
8218 \@gls@loadlist
```

The styles that use the longtable environment. These are not loaded if the nolong package option is used.

```
8219 \@gls@loadlong
```

The styles that use the supertabular environment. These are not loaded if the nosuper package option is used or if the package isn't installed.

```
8220 \@gls@loadsuper
```

The tree-like styles. These are not loaded if the notree package option is used.

```
8221 \@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.

```
8222 \ifx\@glossary@default@style\relax
```

```
8223 \else
```

```
8224   \setglossarystyle{\@glossary@default@style}
```

```
8225 \fi
```

1.20 Debugging Commands

`\showgloparent`

```
\showgloparent{<label>}
```

```
8226 \newcommand*{\showgloparent}[1]{%
```

```
8227   \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
```

```
8228 }
```

`\showglolevel`

```
\showglolevel{<label>}
```

```
8229 \newcommand*{\showglolevel}[1]{%
```

```
8230   \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
```

```
8231 }
```

`\showglotext`

```
\showglotext{<label>}
```

```

8232 \newcommand*{\showglotext}[1]{%
8233   \expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
8234 }

```

\showgloplural \showgloplural{\label{}}

```

8235 \newcommand*{\showgloplural}[1]{%
8236   \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
8237 }

```

\showglofirst \showglofirst{\label{}}

```

8238 \newcommand*{\showglofirst}[1]{%
8239   \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
8240 }

```

\showglofirstpl \showglofirstpl{\label{}}

```

8241 \newcommand*{\showglofirstpl}[1]{%
8242   \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
8243 }

```

\showglotype \showglotype{\label{}}

```

8244 \newcommand*{\showglotype}[1]{%
8245   \expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
8246 }

```

\showglocounter \showglocounter{\label{}}

```

8247 \newcommand*{\showglocounter}[1]{%
8248   \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
8249 }

```

\showglouser \showglouser{\label{}}

```

8250 \newcommand*{\showglouser}[1]{%

```

```
8251 \expandafter\show\csname glo@glstetoklabel{#1}@useri\endcsname
8252 }
```

\showglouserii \showglouserii{<label>}

```
8253 \newcommand*{\showglouserii}[1]{%
8254 \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
8255 }
```

\showglouseriii \showglouseriii{<label>}

```
8256 \newcommand*{\showglouseriii}[1]{%
8257 \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
8258 }
```

\showglouseriv \showglouseriv{<label>}

```
8259 \newcommand*{\showglouseriv}[1]{%
8260 \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
8261 }
```

\showglouserv \showglouserv{<label>}

```
8262 \newcommand*{\showglouserv}[1]{%
8263 \expandafter\show\csname glo@glstetoklabel{#1}@userv\endcsname
8264 }
```

\showglouservi \showglouservi{<label>}

```
8265 \newcommand*{\showglouservi}[1]{%
8266 \expandafter\show\csname glo@glstetoklabel{#1}@uservi\endcsname
8267 }
```

\showgloname \showgloname{<label>}

```
8268 \newcommand*{\showgloname}[1]{%
8269 \expandafter\show\csname glo@glstetoklabel{#1}@name\endcsname
8270 }
```

\showglodesc \showglodesc{<label>}

```
8271 \newcommand*{\showglodesc}[1]{%
8272   \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname
8273 }
```

owglodescplural \showglodescplural{<label>}

```
8274 \newcommand*{\showglodescplural}[1]{%
8275   \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
8276 }
```

\showglosort \showglosort{<label>}

```
8277 \newcommand*{\showglosort}[1]{%
8278   \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
8279 }
```

\showglosymbol \showglosymbol{<label>}

```
8280 \newcommand*{\showglosymbol}[1]{%
8281   \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
8282 }
```

glosymbolplural \showglosymbolplural{<label>}

```
8283 \newcommand*{\showglosymbolplural}[1]{%
8284   \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname
8285 }
```

\showgloshort \showgloshort{<label>}

```
8286 \newcommand*{\showgloshort}[1]{%
8287   \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname
8288 }
```

`\showglolong` `\showglolong{<label>}`

```
8289 \newcommand*{\showglolong}[1]{%
8290   \expandafter\show\csname glo@glstetoklabel{#1}@long\endcsname
8291 }
```

`\showgloindex` `\showgloindex{<label>}`

```
8292 \newcommand*{\showgloindex}[1]{%
8293   \expandafter\show\csname glo@glstetoklabel{#1}@index\endcsname
8294 }
```

`\showgloflag` `\showgloflag{<label>}`

```
8295 \newcommand*{\showgloflag}[1]{%
8296   \expandafter\show\csname ifglo@glstetoklabel{#1}@flag\endcsname
8297 }
```

`\showgloloclist` `\showgloloclist{<label>}`

```
8298 \newcommand*{\showgloloclist}[1]{%
8299   \expandafter\show\csname glo@glstetoklabel{#1}@loclist\endcsname
8300 }
```

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

```
8301 \newcommand*{\showglofield}[2]{%
8302   \csshow{glo@glstetoklabel{#1}@#2}%
8303 }
```

`\showacronymlists` `\showacronymlists`

Show list of glossaries that have been flagged as a list of acronyms.

```
8304 \newcommand*{\showacronymlists}{%
8305   \show\@glsacronymlists
8306 }
```

`\showglossaries` `\showglossaries`

Show list of defined glossaries.

```

8307 \newcommand*{\showglossaries}{%
8308   \show\@glo@types
8309 }

```

`\showglossaryin` `\showglossaryin{<glossary-label>}`

Show the ‘in’ extension for the given glossary.

```

8310 \newcommand*{\showglossaryin}[1]{%
8311   \expandafter\show\csname @glo@types@#1@in\endcsname
8312 }

```

`showglossaryout` `\showglossaryout{<glossary-label>}`

Show the ‘out’ extension for the given glossary.

```

8313 \newcommand*{\showglossaryout}[1]{%
8314   \expandafter\show\csname @glo@types@#1@out\endcsname
8315 }

```

`owglossarytitle` `\showglossarytitle{<glossary-label>}`

Show the title for the given glossary.

```

8316 \newcommand*{\showglossarytitle}[1]{%
8317   \expandafter\show\csname @glo@types@#1@title\endcsname
8318 }

```

`glossarycounter` `\showglossarycounter{<glossary-label>}`

Show the counter for the given glossary.

```

8319 \newcommand*{\showglossarycounter}[1]{%
8320   \expandafter\show\csname @glo@types@#1@counter\endcsname
8321 }

```

`glossaryentries` `\showglossaryentries{<glossary-label>}`

Show the list of entry labels for the given glossary.

```

8322 \newcommand*{\showglossaryentries}[1]{%
8323   \expandafter\show\csname glolist@#1\endcsname
8324 }

```

1.21 Compatibility with version 2.07 and below

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.

```
8325 \csname ifglcompatible-2.07\endcsname
8326   \RequirePackage{glossaries-compatible-207}
8327 \fi
```


2 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.

```
8328 \NeedsTeXFormat{LaTeX2e}
```

```
8329 \ProvidesPackage{glossaries-prefix}[2021/09/20 v4.47 (NLCT)]
```

Pass all options to glossaries:

```
8330 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
8331 \ProcessOptions
```

Load glossaries:

```
8332 \RequirePackage{glossaries}
```

Add the new keys:

```
8333 \define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
```

```
8334 \define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
```

```
8335 \define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
```

```
8336 \define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
```

Add them to `\@gls@keymap`:

```
8337 \appto\@gls@keymap{,%
```

```
8338   {prefixfirst}{prefixfirst},%
```

```
8339   {prefixfirstplural}{prefixfirstplural},%
```

```
8340   {prefix}{prefix},%
```

```
8341   {prefixplural}{prefixplural}}%
```

```
8342 }
```

Set the default values:

```
8343 \appto\@newglossaryentryprehook{%
```

```
8344   \def\@glo@entryprefix{}
```

```
8345   \def\@glo@entryprefixplural{}
```

```
8346   \let\@glo@entryprefixfirst\@gls@default@value
```

```
8347   \let\@glo@entryprefixfirstplural\@gls@default@value
```

```
8348 }
```

Set the assignment code:

```
8349 \appto\@newglossaryentryposthook{%
```

```
8350   \gls@assign@field{}{\@glo@label}{prefix}{\@glo@entryprefix}%
```

```
8351   \gls@assign@field{}{\@glo@label}{prefixplural}{\@glo@entryprefixplural}%
```

If `prefixfirst` has not been supplied, make it the same as `prefix`.

```
8352   \expandafter\gls@assign@field\expandafter
```

```
8353     {\csname glo@\@glo@label @prefix\endcsname}{\@glo@label}{prefixfirst}%
```

```
8354     {\@glo@entryprefixfirst}}%
```

If prefixfirstplural has not been supplied, make it the same as prefixplural.

```

8355 \expandafter\gls@assign@field\expandafter
8356   {\csname glo@\glo@label @prefixplural\endcsname}{\@glo@label}%
8357   {prefixfirstplural}{\@glo@entryprefixfirstplural}%
8358 }

```

Define commands to access these fields:

entryprefixfirst

```

8359 \newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}

```

entryfirstplural

```

8360 \newcommand*{\glsentryprefixfirstplural}[1]{%
8361   \csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}

```

\glsentryprefix

```

8362 \newcommand*{\glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}

```

entryprefixplural

```

8363 \newcommand*{\glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}

```

Now for the initial upper case variants:

entryprefixfirst

```

8364 \newrobustcmd*{\Glsentryprefixfirst}[1]{%
8365   \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
8366   \xmakefirstuc\@glo@text
8367 }

```

entryfirstplural

```

8368 \newrobustcmd*{\Glsentryprefixfirstplural}[1]{%
8369   \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
8370   \xmakefirstuc\@glo@text
8371 }

```

\Glsentryprefix

```

8372 \newrobustcmd*{\Glsentryprefix}[1]{%
8373   \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
8374   \xmakefirstuc\@glo@text
8375 }

```

entryprefixplural

```

8376 \newrobustcmd*{\Glsentryprefixplural}[1]{%
8377   \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
8378   \xmakefirstuc\@glo@text
8379 }

```

Define commands to determine if the prefix keys have been set:

`\ifglshasprefix`

```
8380 \newcommand*{\ifglshasprefix}[3]{%
8381   \ifcempty{glo@glstdetoklabel{#1}@prefix}%
8382   {#3}%
8383   {#2}%
8384 }
```

`hasprefixplural`

```
8385 \newcommand*{\ifglshasprefixplural}[3]{%
8386   \ifcempty{glo@glstdetoklabel{#1}@prefixplural}%
8387   {#3}%
8388   {#2}%
8389 }
```

`shasprefixfirst`

```
8390 \newcommand*{\ifglshasprefixfirst}[3]{%
8391   \ifcempty{glo@glstdetoklabel{#1}@prefixfirst}%
8392   {#3}%
8393   {#2}%
8394 }
```

`efixfirstplural`

```
8395 \newcommand*{\ifglshasprefixfirstplural}[3]{%
8396   \ifcempty{glo@glstdetoklabel{#1}@prefixfirstplural}%
8397   {#3}%
8398   {#2}%
8399 }
```

`fix@record@hook` Need to take into account the possibility that glossaries-extra might be loaded with the record option.

```
8400 \providecommand{\@glsprefix@record@hook}[2]{%
8401   \ifdef\@glsextr@record
8402   {\@glsextr@record{#1}{#2}{glslink}}%
8403   {}%
8404 }
```

`\glsprefixsep` Separator between prefix and term. Does nothing by default.

```
8405 \newcommand{\glsprefixsep}{}%
```

Define commands that insert the prefix before commands like `\gls`:

`\pgls`

```
8406 \newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
```

`\@pgls` Unstarred version.

```
8407 \newcommand*{\@pgls}[2][ ]{%
8408   \new@ifnextchar[%
8409   {\@pgls@{#1}{#2}}%
```

```

8410  {\@pgls@{#1}{#2} []}%
8411 }

```

\@pgls@ Read in the final optional argument:

```

8412 \def\@pgls@#1#2[#3]{%
8413   \@glsprefix@record@hook{#1}{#2}%
8414   \glsdoifexists{#2}%
8415   {%
8416     \ifglsused{#2}%
8417     {%
8418       \ifglshasprefix{#2}{\glstryprefix{#2}\glsprefixsep}{}%
8419     }%
8420     {%
8421       \ifglshasprefixfirst{#2}{\glstryprefixfirst{#2}\glsprefixsep}{}%
8422     }%
8423     \@gls@{#1}{#2}[#3]%
8424   }%
8425 }

```

Similarly for the plural version:

```

\pglsp1
8426 \newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}

```

\@pglsp1 Unstarred version.

```

8427 \newcommand*{\@pglsp1}[2] [] {%
8428   \new@ifnextchar[%
8429     {\@pglsp1@{#1}{#2}}%
8430     {\@pglsp1@{#1}{#2} []}%
8431 }

```

\@pglsp1@ Read in the final optional argument:

```

8432 \def\@pglsp1@#1#2[#3]{%
8433   \@glsprefix@record@hook{#1}{#2}%
8434   \glsdoifexists{#2}%
8435   {%
8436     \ifglsused{#2}%
8437     {%
8438       \ifglshasprefixplural{#2}{\glstryprefixplural{#2}\glsprefixsep}{}%
8439     }%
8440     {%
8441       \ifglshasprefixfirstplural{#2}%
8442       {\glstryprefixfirstplural{#2}\glsprefixsep}{}%
8443     }%
8444     \@glspl@{#1}{#2}[#3]%
8445   }%
8446 }

```

Now for the first letter upper case versions:

```
\Pgl's
8447 \newrobustcmd{\Pgl's}{\@gls@hyp@opt\@Pgl's}
```

\@Pgl's Unstarred version.

```
8448 \newcommand*{\@Pgl's}[2][\%
8449 \new@ifnextchar[%
8450 {\@Pgl's@{#1}{#2}}%
8451 {\@Pgl's@{#1}{#2}[]}%
8452 }
```

\@Pgl's@ Read in the final optional argument:

```
8453 \def\@Pgl's@#1#2[#3]{%
8454 \glsprefix@record@hook{#1}{#2}%
8455 \glsdoifexists{#2}%
8456 {%
8457 \ifglsused{#2}%
8458 {%
8459 \ifglshasprefix{#2}%
8460 {%
8461 \Glsentryprefix{#2}%
8462 \glsprefixsep
8463 \@gls@{#1}{#2}[#3]%
8464 }%
8465 {\@Gls@{#1}{#2}[#3]}%
8466 }%
8467 {%
8468 \ifglshasprefixfirst{#2}%
8469 {%
8470 \Glsentryprefixfirst{#2}%
8471 \glsprefixsep
8472 \@gls@{#1}{#2}[#3]%
8473 }%
8474 {\@Gls@{#1}{#2}[#3]}%
8475 }%
8476 }%
8477 }
```

Similarly for the plural version:

```
\Pgl'spl
8478 \newrobustcmd{\Pgl'spl}{\@gls@hyp@opt\@Pgl'spl}
```

\@Pgl'spl Unstarred version.

```
8479 \newcommand*{\@Pgl'spl}[2][\%
8480 \new@ifnextchar[%
8481 {\@Pgl'spl@{#1}{#2}}%
8482 {\@Pgl'spl@{#1}{#2}[]}%
8483 }
```

\@Pglsp1@ Read in the final optional argument:

```

8484 \def\@Pglsp1@#1#2[#3]{%
8485   \@glsprefix@record@hook{#1}{#2}%
8486   \glsoifexists{#2}%
8487   {%
8488     \ifglused{#2}%
8489     {%
8490       \ifglshasprefixplural{#2}%
8491       {%
8492         \Glsentryprefixplural{#2}%
8493         \glsprefixsep
8494         \@glsp1@{#1}{#2}[#3]%
8495       }%
8496       {\@Glspl@{#1}{#2}[#3]}%
8497     }%
8498     {%
8499       \ifglshasprefixfirstplural{#2}%
8500       {%
8501         \Glsentryprefixfirstplural{#2}%
8502         \glsprefixsep
8503         \@glsp1@{#1}{#2}[#3]%
8504       }%
8505       {\@Glspl@{#1}{#2}[#3]}%
8506     }%
8507   }%
8508 }
```

Finally the all upper case versions:

\PGLS

```

8509 \newrobustcmd{\PGLS}{\@glshyp@opt\@PGLS}
```

\@PGLS Unstarred version.

```

8510 \newcommand*{\@PGLS}[2][ ]{%
8511   \new@ifnextchar[%
8512   {\@PGLS@{#1}{#2}}%
8513   {\@PGLS@{#1}{#2}[ ]}%
8514 }
```

\@PGLS@ Read in the final optional argument:

```

8515 \def\@PGLS@#1#2[#3]{%
8516   \@glsprefix@record@hook{#1}{#2}%
8517   \glsoifexists{#2}%
8518   {%
8519     \ifglused{#2}%
8520     {%
8521       \ifglshasprefix{#2}%
8522       {\mfirstucMakeUppercase{\Glsentryprefix{#2}\glsprefixsep}}{}}%
```

```

8523 }%
8524 {%
8525     \ifglshasprefixfirst{#2}%
8526     {\mfirstucMakeUppercase{\glstryentryprefixfirst{#2}\glsprefixsep}}{}%
8527 }%
8528 \@GLS@{#1}{#2}[#3]%
8529 }%
8530 }

```

Plural version:

\PGLSp1

```

8531 \newrobustcmd{\PGLSp1}{\@glshyp@opt\PGLSp1}

```

\@PGLSp1 Unstarred version.

```

8532 \newcommand*{\@PGLSp1}[2][{}]{%
8533     \new@ifnextchar[%
8534     {\@PGLSp1@{#1}{#2}}%
8535     {\@PGLSp1@{#1}{#2}[]}%
8536 }

```

\@PGLSp1@ Read in the final optional argument:

```

8537 \def\@PGLSp1@#1#2[#3]{%
8538     \@glsprefix@record@hook{#1}{#2}%
8539     \glsdoidexists{#2}%
8540     {%
8541         \ifglused{#2}%
8542         {%
8543             \ifglshasprefixplural{#2}%
8544             {\mfirstucMakeUppercase{\glstryentryprefixplural{#2}\glsprefixsep}}{}%
8545         }%
8546         {%
8547             \ifglshasprefixfirstplural{#2}%
8548             {\mfirstucMakeUppercase{\glstryentryprefixfirstplural{#2}\glsprefixsep}}{}%
8549         }%
8550         \@GLSp1@{#1}{#2}[#3]%
8551     }%
8552 }

```

3 Glossary Styles

3.1 Glossary hyper-navigation definitions (glossary-hypernav package)

Package Definition:

```
8553 \ProvidesPackage{glossary-hypernav}[2021/09/20 v4.47 (NLCT)]
```

The commands defined in this package are provided to help navigate around the groups within a glossary (see [section 1.16](#).) `\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`

```
8554 \newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
```

```
8555   \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
```

```
8556   \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
```

`navhyperlinkname`

Expands to the hypertarget name. The first argument is the glossary type. The second argument is the group label.

```
8557 \newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
```

`navhypertarget`

```
\glsnavhypertarget[⟨type⟩]{⟨label⟩}{⟨text⟩}
```

This command makes `⟨text⟩` a hypertarget for the glossary group whose label is given by `⟨label⟩` in the glossary given by `⟨type⟩`. If `⟨type⟩` is omitted, `\@glo@type` is used which is set by `\printglossary` to the current glossary label.

```
8558 \newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
```

```
8559   \@glsnavhypertarget{#1}{#2}{#3}%
```

```
8560 }
```

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.

snavhypertarget

```
8561 \newcommand*{\@glsnavhypertarget}[3]{%
    Add this group to the aux file for re-run check.
8562   \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
    Add the target.
8563   \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
    Check list of known groups to determine if a re-run is required.
8564   \expandafter\let
8565     \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
    Iterate through list and terminate loop if this group is found.
8566   \@for\@gls@elem:=\@gls@list\do{%
8567     \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
    Check if list terminated prematurely.
8568   \if@endfor
8569   \else
    This group was not included in the list, so issue a warning.
8570     \GlossariesWarningNoLine{Navigation panel
8571       for glossary type ‘#1’~Jmissing group ‘#2’}%
8572     \gdef\gls@hypergroup@rerun{%
8573       \GlossariesWarningNoLine{Navigation panel
8574         has changed. Rerun LaTeX}}%
8575   \fi
8576 }
```

hypergroup@rerun Give a warning at the end if re-run required

```
8577 \let\gls@hypergroup@rerun\relax
8578 \AtEndDocument{\gls@hypergroup@rerun}
```

`\@gls@hypergroup` This adds to (or creates) the command `\@gls@hypergroup@list@<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.

```
8579 \newcommand*{\@gls@hypergroup}[2]{%
8580 \ifundefined{\@gls@hypergroup@list@#1}{%
8581   \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
8582 }{%
8583   \expandafter\let\expandafter\@gls@tmp
8584     \csname @gls@hypergroup@list@#1\endcsname
8585   \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
8586     \@gls@tmp,#2}%
8587 }%
8588 }
```

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`

```

8589 \newcommand*{\glsnavigation}{%
8590   \def\@gls@between{}%
8591   \ifcsundef{\@gls@hypergrouplist@\@glo@type}%
8592   {%
8593     \def\@gls@list{}%
8594   }%
8595   {%
8596     \expandafter\let\expandafter\@gls@list
8597       \csname \@gls@hypergrouplist@\@glo@type\endcsname
8598   }%
8599   \@for\@gls@tmp:=\@gls@list\do{%
8600     \@gls@between

8601     \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
8602     \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
8603     \let\@gls@between\glshypernavsep
8604   }%
8605 }
```

`\glshypernavsep` Separator for the hyper navigation bar.

```

8606 \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`

```

8607 \newcommand*{\glssymbolnav}{%
8608   \glsnavhyperlink{glssymbols}{\@glsgetgrouptitle{glssymbols}}%
8609   \glshypernavsep
8610   \glsnavhyperlink{glsnumbers}{\@glsgetgrouptitle{glsnumbers}}%
8611   \glshypernavsep
8612 }
```

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.

```

8613 \ProvidesPackage{glossary-inline}[2021/09/20 v4.47 (NLCT)]
```

`inline` Define the inline style.

```

8614 \newglossarystyle{inline}{%
```

Start of glossary sets up first empty separator between entries. (This is then changed by `\glossentry`)

```
8615 \renewenvironment{theglossary}%
8616   {%
8617     \def\gls@inlinesep{%
8618       \def\gls@inlinesubsep{%
8619         \def\gls@inlinepostchild{%
8620           }%
8621         {\glspostinline}%
```

No header:

```
8622 \renewcommand*{\glossaryheader}{}%
```

No group headings (if heading is required, add `\glsinlinedopostchild` to start definition in case heading follows a child entry):

```
8623 \renewcommand*{\glsgroupheading}[1]{}%
```

Just display separator followed by name and description:

```
8624 \renewcommand{\glossentry}[2]{%
8625   \glsinlinedopostchild
8626   \gls@inlinesep
8627   \glsentryitem{##1}%
8628   \glsinlinenameformat{##1}{%
8629     \glossentryname{##1}%
8630   }%
8631   \ifglshasdescsuppressed{##1}%
8632   {%
8633     \glsinlineemptydescformat
8634     {%
8635       \glossentrysymbol{##1}%
8636     }%
8637     {%
8638       ##2%
8639     }%
8640   }%
8641   {%
8642     \ifglshasdesc{##1}%
8643     {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
8644     {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
8645   }%
8646   \ifglshaschildren{##1}%
8647   {%
8648     \glsresetsubentrycounter
8649     \glsinlineparentchildseparator
8650     \def\gls@inlinesubsep{%
8651       \def\gls@inlinepostchild{\glsinlinepostchild}%
8652     }%
8653   }%
8654   \def\gls@inlinesep{\glsinlineseparator}%
8655 }%
```

Sub-entries display description:

```
8656 \renewcommand{\subglossentry}[3]{%
8657   \gls@inlinesubsep%
8658   \glsinlinesubnameformat{##2}{%
8659     \glossentryname{##2}}%
8660   \glssubentryitem{##2}%
8661   \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
8662   \def\gls@inlinesubsep{\glsinlinesubseparator}%
8663 }
```

Nothing special between groups:

```
8664 \renewcommand*{\glsgroupskip}{}%
8665 }
```

linedopostchild

```
8666 \newcommand*{\glsinlinedopostchild}{%
8667   \gls@inlinepostchild
8668   \def\gls@inlinepostchild{}%
8669 }
```

inlineseparator Separator to use between entries.

```
8670 \newcommand*{\glsinlineseparator}{;\space}
```

inlinesubseparator Separator to use between sub-entries.

```
8671 \newcommand*{\glsinlinesubseparator}{,\space}
```

parentchildseparator Separator to use between parent and children.

```
8672 \newcommand*{\glsinlineparentchildseparator}{:\space}
```

inlinepostchild Hook to use between child and next entry

```
8673 \newcommand*{\glsinlinepostchild}{}
```

\glspostinline Terminator for inline glossary.

```
8674 \newcommand*{\glspostinline}{\glspostdescription\space}
```

inlinenameformat Formats the name of the entry (first argument label, second argument name):

```
8675 \newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
```

inlinedescformat Formats the entry's description, symbol and location list:

```
8676 \newcommand*{\glsinlinedescformat}[3]{\space#1}
```

emptydescformat Formats the entry's symbol and location list when the description is empty:

```
8677 \newcommand*{\glsinlineemptydescformat}[2]{}
```

inlinesubnameformat Formats the name of the subentry (first argument label, second argument name):

```
8678 \newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
```

inlinesubdescformat Formats the subentry's description, symbol and location list:

```
8679 \newcommand*{\glsinlinesubdescformat}[3]{#1}
```

3.3 List Style (glossary-list.sty)

The 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.

8680 \ProvidesPackage{glossary-list}[2021/09/20 v4.47 (NLCT)]

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
8681 \providecommand{\indexspace}{%
8682   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
8683 }
```

`tgrouphaderfmt` Provide a way of adjusting the format of the group headings.

```
8684 \newcommand*{\glslistgrouphaderfmt}[1]{#1}
```

`tnavigationitem` 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.

```
8685 \newcommand*{\glslistnavigationitem}[1]{\item[#1]}
```

`list` The list glossary style uses the description environment. The group separator `\glsgrpskip` 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.

```
8686 \newglossarystyle{list}{%
```

Use description environment:

```
8687   \renewenvironment{theglossary}%
8688     {\begin{description}}{\end{description}}%
```

No header at the start of the environment:

```
8689   \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
8690   \renewcommand*{\glsgrpskip}[1]{}%
```

Main (level 0) entries start a new item in the list:

```
8691   \renewcommand*{\glossentry}[2]{%
8692     \item[\glsentryitem{##1}%
8693       \glsentrydesc{##1}{\glsentryname{##1}}]
8694     \glsentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries continue on the same line:

```
8695   \renewcommand*{\subglossentry}[3]{%
8696     \glssubentryitem{##2}%
```

```

8697 \glstarget{##2}{\strut}\space
8698 \glossentrydesc{##2}\glspostdescription\space ##3.}%

Add vertical space between groups:

8699 \renewcommand*{\glsgroupskip}{\ifglsgroupskip\else\indexspace\fi}%
8700 }

```

listgroup The listgroup style is like the list style, but the glossary groups have headings.

```

8701 \newglossarystyle{listgroup}{%
    Base it on the list style:
8702 \setglossarystyle{list}%

    Each group has a heading:
8703 \renewcommand*{\glsgroupheading}[1]{%
8704 \item[\glslistgroupheaderfmt{\glsgrouptitle{##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.

```

8705 \newglossarystyle{listhypergroup}{%
    Base it on the list style:
8706 \setglossarystyle{list}%

    Add navigation links at the start of the environment.
8707 \renewcommand*{\glossaryheader}{%
8708 \glslistnavigationitem{\glsgroupnavigation}}%

    Each group has a heading with a hypertarget:
8709 \renewcommand*{\glsgroupheading}[1]{%
8710 \item[\glslistgroupheaderfmt
8711 {\glsgrouphypertarget{##1}{\glsgrouptitle{##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.

```

8712 \newglossarystyle{altlist}{%
    Base it on the list style:
8713 \setglossarystyle{list}%

    Main (level 0) entries start a new item in the list with a line break after the entry name:
8714 \renewcommand*{\glossentry}[2]{%
8715 \item[\glsgroupitem{##1}%
8716 \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.

```

8717 \mbox{}\par\nobreak\@afterheading
8718 \glossentrydesc{##1}\glspostdescription\space ##2}%

```

Sub-entries start a new paragraph:

```
8719 \renewcommand{\subglossentry}[3]{%
8720   \par
8721   \glssubentryitem{##2}%
8722   \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
8723 }
```

altlistgroup The altlistgroup glossary style is like the altlist style, but the glossary groups have headings.

```
8724 \newglossarystyle{altlistgroup}{%
  Base it on the altlist style:
8725   \setglossarystyle{altlist}%
  Each group has a heading:
8726   \renewcommand*{\glsgroupheading}[1]{%
8727     \item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
```

altlisthypergroup The altlisthypergroup glossary style is like the altlistgroup style, but has a set of links to the groups at the start of the glossary.

```
8728 \newglossarystyle{altlisthypergroup}{%
  Base it on the altlist style:
8729   \setglossarystyle{altlist}%
  Add navigation links at the start of the environment.
8730   \renewcommand*{\glossaryheader}{%
8731     \glslistnavigationitem{\glslnavigation}}%
  Each group has a heading with a hypertarget:
8732   \renewcommand*{\glsgroupheading}[1]{%
8733     \item[\glslistgroupheaderfmt
8734       {\glslnavhypertarget{##1}{\glsgrouptitle{##1}}}]}%
```

listdotted The listdotted glossary style was supplied by Axel Menzel. I've modified it slightly so that the distance from the start of the name to the end of the dotted line is specified by `\glslistdottedwidth`. Note that this style ignores the page numbers as well as the symbol. Sub-entries are displayed in the same way as top-level entries.

```
8735 \newglossarystyle{listdotted}{%
  Base it on the list style:
8736   \setglossarystyle{list}%
  Each main (level 0) entry starts a new item:
8737   \renewcommand*{\glossentry}[2]{%
8738     \item[]\makebox[\glslistdottedwidth][l]{%
8739       \glssentryitem{##1}%
8740       \glstarget{##1}{\glossentryname{##1}}%
8741       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
```

Sub entries have the same format as main entries:

```

8742 \renewcommand*{\subglossentry}[3]{%
8743   \item[\makebox[\glslstdottedwidth][l]{%
8744     \glssubentryitem{##2}%
8745     \glstarget{##2}{\glossentryname{##2}}}%
8746   \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
8747 }

```

`listdottedwidth`

```

8748 \newlength\glslstdottedwidth
8749 \setlength{\glslstdottedwidth}{.5\hsize}

```

`sublistdotted` This style is similar to the `glostylelistdotted` style, except that the main entries just have the name displayed.

```
8750 \newglossarystyle{sublistdotted}{%
```

Base it on the `listdotted` style:

```
8751 \setglossarystyle{listdotted}%
```

Main (level 0) entries just display the name:

```

8752 \renewcommand*{\glossentry}[2]{%
8753   \item[\glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}}}%
8754 }

```

3.4 Glossary Styles using `longtable` (the `glossary-long` package)

The glossary styles defined in the package used the `longtable` environment in the glossary.

```
8755 \ProvidesPackage{glossary-long}[2021/09/20 v4.47 (NLCT)]
```

Requires the package:

```
8756 \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 later, in which case `\glsdescwidth` may have already been defined by . The same goes for `\glspagelistwidth`.)

```

8757 \@ifundefined{glsdescwidth}{%
8758   \newlength\glsdescwidth
8759   \setlength{\glsdescwidth}{0.6\hsize}
8760 }{}

```

`lspagelistwidth` This is a length that governs the width of the page list column.

```

8761 \@ifundefined{glspagelistwidth}{%
8762   \newlength\glspagelistwidth
8763   \setlength{\glspagelistwidth}{0.1\hsize}
8764 }{}

```


long The long glossary style command which uses the longtable environment:

```
8765 \newglossarystyle{long}{%
```

Use longtable with two columns:

```
8766 \renewenvironment{theglossary}{%
8767     {\begin{longtable}{lp{\glsdescwidth}}}%
8768     {\end{longtable}}}%
```

Do nothing at the start of the environment:

```
8769 \renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
8770 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
8771 \renewcommand{\glossentry}[2]{%
8772     \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
8773     \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
8774 }%
```

Sub entries displayed on the following row without the name:

```
8775 \renewcommand{\subglossentry}[3]{%
8776     &
8777     \glssubentryitem{##2}%
8778     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
8779     ##3\tabularnewline
8780 }%
```

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>)

```
8781 \ifglsnogroupskip
8782 \renewcommand*{\glsgroupskip}{}%
8783 \else
8784 \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
8785 \fi
8786 }
```

longborder The longborder style is like the above, but with horizontal and vertical lines:

```
8787 \newglossarystyle{longborder}{%
```

Base it on the glostylelong style:

```
8788 \setglossarystyle{long}%
```

Use longtable with two columns with vertical lines between each column:

```
8789 \renewenvironment{theglossary}{%
8790     \begin{longtable}{|lp{\glsdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
8791 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8792 }
```

longheader The longheader style is like the long style but with a header:

```
8793 \newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
8794 \setglossarystyle{long}{%
```

Set the table's header:

```
8795 \renewcommand*{\glossaryheader}{%
8796 \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
8797 }
```

`longheaderborder` The `longheaderborder` style is like the `long` style but with a header and border:

```
8798 \newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
8799 \setglossarystyle{longborder}{%
```

Set the table's header and add horizontal line to table's foot:

```
8800 \renewcommand*{\glossaryheader}{%
8801 \hline\bfseries \entryname & \bfseries
8802 \descriptionname\tabularnewline\hline
8803 \endhead
8804 \hline\endfoot}%
8805 }
```

`long3col` The `long3col` style is like `long` but with 3 columns

```
8806 \newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
8807 \renewenvironment{theglossary}{%
8808 {\begin{longtable}{lp{\glsgdescwidth}p{\glspagelistwidth}}}%
8809 {\end{longtable}}}%
```

No table header:

```
8810 \renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
8811 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
8812 \renewcommand{\glossentry}[2]{%
8813 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
8814 \glossentrydesc{##1} & ##2\tabularnewline
8815 }%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
8816 \renewcommand{\subglossentry}[3]{%
8817 &
8818 \glssubentryitem{##2}%
8819 \glstarget{##2}{\strut}\glossentrydesc{##2} &
8820 ##3\tabularnewline
8821 }%
```

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>)

```
8822 \ifglsgroupskip
8823 \renewcommand*{\glsgroupskip}{}%
8824 \else
8825 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
8826 \fi
8827 }
```

long3colborder The long3colborder style is like the long3col style but with a border:

```
8828 \newglossarystyle{long3colborder}{%
    Base it on the glostylelong3col style:
8829 \setglossarystyle{long3col}%
    Use a longtable with 3 columns with vertical lines around them:
8830 \renewenvironment{theglossary}%
8831 {\begin{longtable}{|l|p{\glsgdescwidth}|p{\glspagelistwidth}|}%
8832 {\end{longtable}}%
    Place horizontal lines at the head and foot of the table:
8833 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8834 }
```

long3colheader The long3colheader style is like long3col but with a header row:

```
8835 \newglossarystyle{long3colheader}{%
    Base it on the glostylelong3col style:
8836 \setglossarystyle{long3col}%
    Set the table's header:
8837 \renewcommand*{\glossaryheader}{%
8838 \bfseries\entryname&\bfseries\descriptionname&
8839 \bfseries\pagelistname\tabularnewline\endhead}%
8840 }
```

colheaderborder The long3colheaderborder style is like the above but with a border

```
8841 \newglossarystyle{long3colheaderborder}{%
    Base it on the glostylelong3colborder style:
8842 \setglossarystyle{long3colborder}%
    Set the table's header and add horizontal line at table's foot:
8843 \renewcommand*{\glossaryheader}{%
8844 \hline
8845 \bfseries\entryname&\bfseries\descriptionname&
8846 \bfseries\pagelistname\tabularnewline\hline\endhead
8847 \hline\endfoot}%
8848 }
```

`long4col` The `long4col` style has four columns where the third column contains the value of the associated symbol key.

```
8849 \newglossarystyle{long4col}{%
```

Use a `longtable` with 4 columns:

```
8850 \renewenvironment{theglossary}{%
```

```
8851 {\begin{longtable}{l l l l}}%
```

```
8852 {\end{longtable}}}%
```

No table header:

```
8853 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
8854 \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):

```
8855 \renewcommand{\glossentry}[2]{%
```

```
8856 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
```

```
8857 \glossentrydesc{##1} &
```

```
8858 \glossentrysymbol{##1} &
```

```
8859 ##2\tabularnewline
```

```
8860 }%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
8861 \renewcommand{\subglossentry}[3]{%
```

```
8862 &
```

```
8863 \glssubentryitem{##2}%
```

```
8864 \glstarget{##2}{\strut}\glossentrydesc{##2} &
```

```
8865 \glossentrysymbol{##2} & ##3\tabularnewline
```

```
8866 }%
```

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>)

```
8867 \ifglsgnogroupskip
```

```
8868 \renewcommand*{\glsgroupskip}{}%
```

```
8869 \else
```

```
8870 \renewcommand*{\glsgroupskip}{ & & & \tabularnewline}%
```

```
8871 \fi
```

```
8872 }
```

`long4colheader` The `long4colheader` style is like `long4col` but with a header row.

```
8873 \newglossarystyle{long4colheader}{%
```

Base it on the `glostylelong4col` style:

```
8874 \setglossarystyle{long4col}%
```

Table has a header:

```
8875 \renewcommand*{\glossaryheader}{%
```

```
8876 \bfseries\entryname&\bfseries\descriptionname&
```

```
8877 \bfseries \symbolname&
```

```

8878 \bfseries\pagelistname\tabularnewline\endhead}%
8879 }

```

long4colborder The long4colborder style is like long4col but with a border.

```

8880 \newglossarystyle{long4colborder}{%
    Base it on the glostylelong4col style:
8881 \setglossarystyle{long4col}%
    Use a longtable with 4 columns surrounded by vertical lines:
8882 \renewenvironment{theglossary}%
8883 {\begin{longtable}{|l|l|l|l|}}%
8884 {\end{longtable}}%
    Add horizontal lines to the head and foot of the table:
8885 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8886 }

```

colheaderborder The long4colheaderborder style is like the above but with a border.

```

8887 \newglossarystyle{long4colheaderborder}{%
    Base it on the glostylelong4col style:
8888 \setglossarystyle{long4col}%
    Use a longtable with 4 columns surrounded by vertical lines:
8889 \renewenvironment{theglossary}%
8890 {\begin{longtable}{|l|l|l|l|}}%
8891 {\end{longtable}}%
    Add table header and horizontal line at the table's foot:
8892 \renewcommand*{\glossaryheader}{%
8893 \hline\bfseries\entryname&\bfseries\descriptionname&
8894 \bfseries \symbolname&
8895 \bfseries\pagelistname\tabularnewline\hline\endhead
8896 \hline\endfoot}%
8897 }

```

altlong4col The altlong4col style is like the long4col style but can have multiline descriptions and page lists.

```

8898 \newglossarystyle{altlong4col}{%
    Base it on the glostylelong4col style:
8899 \setglossarystyle{long4col}%
    Use a longtable with 4 columns where the second and last columns may have multiple lines
    in each row:
8900 \renewenvironment{theglossary}%
8901 {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
8902 {\end{longtable}}%
8903 }

```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```

8904 \newglossarystyle{altlong4colheader}{%
      Base it on the glostylelong4colheader style:
8905   \setglossarystyle{long4colheader}%
      Use a longtable with 4 columns where the second and last columns may have multiple lines
      in each row:
8906   \renewenvironment{theglossary}%
8907     {\begin{longtable}{\lp{\glsgdescwidth}\lp{\glspagelistwidth}}}%
8908     {\end{longtable}}}%
8909 }
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```

8910 \newglossarystyle{altlong4colborder}{%
      Base it on the glostylelong4colborder style:
8911   \setglossarystyle{long4colborder}%
      Use a longtable with 4 columns where the second and last columns may have multiple lines
      in each row:
8912   \renewenvironment{theglossary}%
8913     {\begin{longtable}{\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}%
8914     {\end{longtable}}}%
8915 }
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```

8916 \newglossarystyle{altlong4colheaderborder}{%
      Base it on the glostylelong4colheaderborder style:
8917   \setglossarystyle{long4colheaderborder}%
      Use a longtable with 4 columns where the second and last columns may have multiple lines
      in each row:
8918   \renewenvironment{theglossary}%
8919     {\begin{longtable}{\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}%
8920     {\end{longtable}}}%
8921 }
```

3.5 Glossary Styles using longtable and booktabs (the `glossary-longbooktabs`) package

The styles here are based on David Carlisle's patch at <http://tex.stackexchange.com/a/56890>

```

8922 \ProvidesPackage{glossary-longbooktabs}[2021/09/20 v4.47 (NLCT)]
```

Requires `booktabs` package:

```

8923 \RequirePackage{booktabs}
```

and the base packages for long styles:

```
8924 \RequirePackage{glossary-long}
8925 \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.

```
8926 \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.

```
8927 \glspatchLToutput
```

As with the longheader style, use the long style as a base.

```
8928 \setglossarystyle{long}%
```

Add a header with rules.

```
8929 \renewcommand*{\glossaryheader}{%
8930 \toprule \bfseries \entryname & \bfseries
8931 \descriptionname\tabularnewline\midrule\endhead
8932 \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.

```
8933 \ifglsgroupskip
8934 \renewcommand*{\glsgroupskip}{}%
8935 \else
8936 \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
8937 \fi
8938 }
```

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.

```
8939 \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.

```
8940 \glspatchLToutput
```

Use the long3col style as a base.

```
8941 \setglossarystyle{long3col}%
```

Add a header with rules.

```
8942 \renewcommand*{\glossaryheader}{%
8943 \toprule \bfseries \entryname &
8944 \bfseries \descriptionname &
8945 \bfseries \pagelistname
8946 \tabularnewline\midrule\endhead
8947 \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.

```

8948 \ifglsgroupskip
8949 \renewcommand*{\glsgroupskip}{}%
8950 \else
8951 \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
8952 \fi
8953 }

```

ng4col-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.

```

8954 \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.

```

8955 \glspatchLToutput

```

Use the long4col style as a base.

```

8956 \setglossarystyle{long4col}%

```

Add a header with rules.

```

8957 \renewcommand*{\glossaryheader}{%
8958 \toprule \bfseries \entryname &
8959 \bfseries \descriptionname &
8960 \bfseries \symbolname &
8961 \bfseries \pagelistname
8962 \tabularnewline\midrule\endhead
8963 \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.

```

8964 \ifglsgroupskip
8965 \renewcommand*{\glsgroupskip}{}%
8966 \else
8967 \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
8968 \fi
8969 }

```

ng4col-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.

```

8970 \newglossarystyle{altlong4col-booktabs}{%

```

The patch \glspatchLToutput is already applied in long4col-booktabs and so doesn't need to be here.

```

8971 \glspatchLToutput

```

Use the long4col-booktabs style as a base.

```

8972 \setglossarystyle{long4col-booktabs}%

```


Change the column specifications:

```
8973 \renewenvironment{theglossary}%  
8974   {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%  
8975   {\end{longtable}}%  
8976 }
```

Ragged styles.

ragged-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.

```
8977 \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.

```
8978 \glspatchLToutput
```

Use the long-booktabs style as a base.

```
8979 \setglossarystyle{long-booktabs}%
```

Adjust the column specification.

```
8980 \renewenvironment{theglossary}%  
8981   {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}}}%  
8982   {\end{longtable}}%  
8983 }
```

ed3col-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.

```
8984 \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.

```
8985 \glspatchLToutput
```

Use the long3col-booktabs style as a base.

```
8986 \setglossarystyle{long3col-booktabs}%
```

Adjust the column specification.

```
8987 \renewenvironment{theglossary}%  
8988   {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}%  
8989     >{\raggedright}p{\glspagelistwidth}}}%  
8990   {\end{longtable}}%  
8991 }
```

ed4col-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.

```
8992 \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.

```
8993 \glspatchLToutput
```

Use the altlong4col-booktabs style as a base.

```
8994 \setglossarystyle{altlong4col-booktabs}%
```

Adjust the column specification.

```
8995 \renewenvironment{theglossary}%  
8996 {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}l%  
8997 >{\raggedright}p{\glspagelistwidth}}}%  
8998 {\end{longtable}}%  
8999 }
```

sLTpenaltycheck

```
9000 \newcommand*{\glslTpenaltycheck}{%  
9001 \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi  
9002 }
```

enaltygroupskip

```
9003 \newcommand{\glspenaltygroupskip}{%  
9004 \noalign{\penalty-50\vskip\normalbaselineskip}}
```

restoreLToutput Provide a way of restoring \LT@output for the user.

```
9005 \let\@gls@org@LT@output\LT@output  
9006 \newcommand*{\glstoreLToutput}{\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.

lspatchLToutput

```
9007 \newcommand*{\glspatchLToutput}{%  
9008 \renewcommand*{\LT@output}{%  
9009 \ifnum\outputpenalty < -\@Mi  
9010 \ifnum\outputpenalty > -\LT@end@pen  
9011 \LT@err{floats and marginpars not allowed in a longtable}\@ehc  
9012 \else  
9013 \setbox\z@\vbox{\unvbox\@cclv}%  
9014 \ifdim \ht\LT@lastfoot>\ht\LT@foot  
9015 \dimen@ \pagegoal  
9016 \advance\dimen@-\ht\LT@lastfoot  
9017 \ifdim\dimen@<\ht\z@  
9018 \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%  
9019 \@makecol  
9020 \@outputpage  
9021 \setbox\z@\vbox{\box\LT@head\glslTpenaltycheck}%  
9022 \fi  
9023 \fi  
9024 \global\@colroom\@colht  
9025 \global\vsize\@colht  
9026 {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%  
9027 \fi  
9028 \else
```

```

9029 \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
9030 \@makecol
9031 \@outputpage
9032 \global\ysize\@colroom
9033 \copy\LT@head
9034 \glsLTpenaltycheck
9035 \nobreak
9036 \fi
9037 }%
9038 }

```

3.6 Glossary Styles using longtable (the glossary-longragged package)

The glossary styles defined in the package used the longtable environment in the glossary and use ragged right formatting for the multiline columns.

```
9039 \ProvidesPackage{glossary-longragged}[2021/09/20 v4.47 (NLCT)]
```

Requires the package:

```
9040 \RequirePackage{array}
```

Requires the package:

```
9041 \RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```

9042 \@ifundefined{glsdescwidth}{%
9043 \newlength\glsdescwidth
9044 \setlength{\glsdescwidth}{0.6\hsize}
9045 }{}

```

`glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```

9046 \@ifundefined{glspagelistwidth}{%
9047 \newlength\glspagelistwidth
9048 \setlength{\glspagelistwidth}{0.1\hsize}
9049 }{}

```

`longragged` The longragged glossary style is like the long but uses ragged right formatting for the description column.

```
9050 \newglossarystyle{longragged}{%
```

Use longtable with two columns:

```

9051 \renewenvironment{theglossary}%
9052 {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
9053 {\end{longtable}}%

```

Do nothing at the start of the environment:

```
9054 \renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
9055 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
9056 \renewcommand{\glossentry}[2]{%
9057   \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9058   \glossentrydesc{##1}\glspostdescription\space ##2%
9059   \tabularnewline
9060 }%
```

Sub entries displayed on the following row without the name:

```
9061 \renewcommand{\subglossentry}[3]{%
9062   &
9063   \glssubentryitem{##2}%
9064   \glstarget{##2}{\strut}\glossentrydesc{##2}%
9065   \glspostdescription\space ##3%
9066   \tabularnewline
9067 }%
```

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>)

```
9068 \ifglsgroupskip
9069 \renewcommand*{\glsgroupskip}{}%
9070 \else
9071 \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
9072 \fi
9073 }
```

`ongraggedborder` The `longraggedborder` style is like the above, but with horizontal and vertical lines:

```
9074 \newglossarystyle{longraggedborder}{%
```

Base it on the `glostylelongragged` style:

```
9075 \setglossarystyle{longragged}%
```

Use `longtable` with two columns with vertical lines between each column:

```
9076 \renewenvironment{theglossary}{%
9077   \begin{longtable}{|l|>{\raggedright}p{\glstdescwidth}|}%
9078   {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
9079 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9080 }
```

`ongraggedheader` The `longraggedheader` style is like the `longragged` style but with a header:

```
9081 \newglossarystyle{longraggedheader}{%
```

Base it on the `glostylelongragged` style:

```
9082 \setglossarystyle{longragged}%
```

Set the table's header:

```
9083 \renewcommand*{\glossaryheader}{%
9084   \bfseries \entryname & \bfseries \descriptionname
```

```

9085 \tabularnewline\endhead}%
9086 }

```

`longraggedheaderborder` The `longraggedheaderborder` style is like the `longragged` style but with a header and border:

```

9087 \newglossarystyle{longraggedheaderborder}{%
    Base it on the glostylelongraggedborder style:
9088 \setglossarystyle{longraggedborder}%
    Set the table's header and add horizontal line to table's foot:
9089 \renewcommand*{\glossaryheader}{%
9090 \hline\bfseries \entryname & \bfseries \descriptionname
9091 \tabularnewline\hline
9092 \endhead
9093 \hline\endfoot}%
9094 }

```

`longragged3col` The `longragged3col` style is like `longragged` but with 3 columns

```

9095 \newglossarystyle{longragged3col}{%
    Use a longtable with 3 columns:
9096 \renewenvironment{theglossary}%
9097 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}%
9098 >{\raggedright}p{\glspagelistwidth}}}%
9099 {\end{longtable}}%

```

No table header:

```

9100 \renewcommand*{\glossaryheader}{}%

```

No headings between groups:

```

9101 \renewcommand*{\glsgroupheading}[1]{}%

```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```

9102 \renewcommand{\glossentry}[2]{%
9103 \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9104 \glossentrydesc{##1} & ##2\tabularnewline
9105 }%

```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```

9106 \renewcommand{\subglossentry}[3]{%
9107 &
9108 \glssubentryitem{##2}%
9109 \glstarget{##2}{\strut}\glossentrydesc{##2} &
9110 ##3\tabularnewline
9111 }%

```

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>)

```

9112 \ifglsnogroupskip
9113 \renewcommand*{\glsgroupskip}{}%

```

```

9114 \else
9115 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
9116 \fi
9117 }

```

ragged3colborder The longragged3colborder style is like the longragged3col style but with a border:

```

9118 \newglossarystyle{longragged3colborder}{%
    Base it on the glostylelongragged3col style:
9119 \setglossarystyle{longragged3col}%
    Use a longtable with 3 columns with vertical lines around them:
9120 \renewenvironment{theglossary}%
9121 {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|%
9122 >{\raggedright}p{\glspagelistwidth}|}%
9123 {\end{longtable}}%
    Place horizontal lines at the head and foot of the table:
9124 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9125 }

```

ragged3colheader The longragged3colheader style is like longragged3col but with a header row:

```

9126 \newglossarystyle{longragged3colheader}{%
    Base it on the glostylelongragged3col style:
9127 \setglossarystyle{longragged3col}%
    Set the table's header:
9128 \renewcommand*{\glossaryheader}{%
9129 \bfseries\entryname&\bfseries\descriptionname&
9130 \bfseries\pagelistname\tabularnewline\endhead}%
9131 }

```

colheaderborder The longragged3colheaderborder style is like the above but with a border

```

9132 \newglossarystyle{longragged3colheaderborder}{%
    Base it on the glostylelongragged3colborder style:
9133 \setglossarystyle{longragged3colborder}%
    Set the table's header and add horizontal line at table's foot:
9134 \renewcommand*{\glossaryheader}{%
9135 \hline
9136 \bfseries\entryname&\bfseries\descriptionname&
9137 \bfseries\pagelistname\tabularnewline\hline\endhead
9138 \hline\endfoot}%
9139 }

```

altlongragged4col The altlongragged4col style is like the altlong4col style defined in the package, except that ragged right formatting is used for the description and page list columns.

```

9140 \newglossarystyle{altlongragged4col}{%

```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9141 \renewenvironment{theglossary}%
9142   {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}l%
9143     >{\raggedright}p{\glspagelistwidth}}}%
9144   {\end{longtable}}%
```

No table header:

```
9145 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9146 \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):

```
9147 \renewcommand{\glossentry}[2]{%
9148   \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9149   \glossentrydesc{##1} & \glossentrysymbol{##1} &
9150   ##2\tabularnewline
9151 }%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
9152 \renewcommand{\subglossentry}[3]{%
9153   &
9154   \glssubentryitem{##2}%
9155   \glstarget{##2}{\strut}\glossentrydesc{##2} &
9156   \glossentrysymbol{##2} & ##3\tabularnewline
9157 }%
```

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>)

```
9158 \ifglsgroupskip
9159   \renewcommand*{\glsgroupskip}{}%
9160 \else
9161   \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
9162 \fi
9163 }
```

agged4colheader The altlongragged4colheader style is like altlongragged4col but with a header row.

```
9164 \newglossarystyle{altlongragged4colheader}{%
```

Base it on the glostylealtlongragged4col style:

```
9165 \setglossarystyle{altlongragged4col}%
```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9166 \renewenvironment{theglossary}%
9167   {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}l%
9168     >{\raggedright}p{\glspagelistwidth}}}%
9169   {\end{longtable}}%
```

Table has a header:

```

9170 \renewcommand*{\glossaryheader}{%
9171   \bfseries\entryname&\bfseries\descriptionname&
9172   \bfseries \symbolname&
9173   \bfseries\pagelistname\tabularnewline\endhead}%
9174 }
```

`ragged4colborder` The `altlongragged4colborder` style is like `altlongragged4col` but with a border.

```

9175 \newglossarystyle{altlongragged4colborder}{%
```

Base it on the `glostylealtlongragged4col` style:

```

9176 \setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```

9177 \renewenvironment{theglossary}%
9178   {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|l|}%
9179    >{\raggedright}p{\glspagelistwidth}|}}%
9180   {\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```

9181 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9182 }
```

`colheaderborder` The `altlongragged4colheaderborder` style is like the above but with a header as well as a border.

```

9183 \newglossarystyle{altlongragged4colheaderborder}{%
```

Base it on the `glostylealtlongragged4col` style:

```

9184 \setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```

9185 \renewenvironment{theglossary}%
9186   {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|l|}%
9187    >{\raggedright}p{\glspagelistwidth}|}}%
9188   {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```

9189 \renewcommand*{\glossaryheader}{%
9190   \hline\bfseries\entryname&\bfseries\descriptionname&
9191   \bfseries \symbolname&
9192   \bfseries\pagelistname\tabularnewline\hline\endhead
9193   \hline\endfoot}%
9194 }
```

3.7 Glossary Styles using multicol (glossary-mcols.sty)

The style file defines glossary styles that use the `multicol` package. These use the tree-like glossary styles in a `multicol` environment.

```

9195 \ProvidesPackage{glossary-mcols}[2021/09/20 v4.47 (NLCT)]
```


Required packages:

```
9196 \RequirePackage{multicol}
9197 \RequirePackage{glossary-tree}
```

`\indexspace` The are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
9198 \providecommand{\indexspace}{%
9199   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
9200 }
```

`\glsmcols` Define macro in which to store the number of columns. (Defaults to 2.)

```
9201 \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.)

```
9202 \newglossarystyle{mcolindex}{%
9203   \setglossarystyle{index}%
9204   \renewenvironment{theglossary}%
9205     {%
9206       \begin{multicols}{\glsmcols}
9207       \setlength{\parindent}{0pt}%
9208       \setlength{\parskip}{0pt plus 0.3pt}%
9209       \let\item\glstreeitem
9210       \let\subitem\glstreesubitem
9211       \let\subsubitem\glstreesubsubitem
9212     }%
9213     {\end{multicols}}}%
9214 }
```

`mcolindexgroup` As `mcolindex` but has headings:

```
9215 \newglossarystyle{mcolindexgroup}{%
9216   \setglossarystyle{mcolindex}%
9217   \renewcommand*{\glsgroupheading}[1]{%
9218     \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
9219 }
```

`indexhypergroup` The `mcolindexhypergroup` style is like the `mcolindexgroup` style but has hyper navigation.

```
9220 \newglossarystyle{mcolindexhypergroup}{%
```

Base it on the `glostylemcolindex` style:

```
9221   \setglossarystyle{mcolindex}%
```

Put navigation links to the groups at the start of the glossary:

```
9222   \renewcommand*{\glossaryheader}{%
9223     \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.

```
9224 \renewcommand*{\glsgroupheading}[1]{%
9225   \item\glstreegroupheaderfmt
9226     {\glsnavigationhypertarget{##1}{\glsgrouptitle{##1}}}%
9227   \indexspace}%
9228 }
```

colindexspannav Similar to **colindexhypergroup**, but puts the navigation line in the optional argument of **multicols**.

```
9229 \newglossarystyle{colindexspannav}{%
9230   \setglossarystyle{index}%
9231   \renewenvironment{theglossary}%
9232     {%
9233       \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
9234       \setlength{\parindent}{0pt}%
9235       \setlength{\parskip}{0pt plus 0.3pt}%
9236       \let\item\glstreeitem}%
9237     {\end{multicols}}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
9238 \renewcommand*{\glsgroupheading}[1]{%
9239   \item\glstreegroupheaderfmt
9240     {\glsnavigationhypertarget{##1}{\glsgrouptitle{##1}}}%
9241   \indexspace}%
9242 }
```

mcoltree Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
9243 \newglossarystyle{mcoltree}{%
9244   \setglossarystyle{tree}%
9245   \renewenvironment{theglossary}%
9246     {%
9247       \begin{multicols}{\glsmcols}
9248       \setlength{\parindent}{0pt}%
9249       \setlength{\parskip}{0pt plus 0.3pt}%
9250     }%
9251     {\end{multicols}}%
9252 }
```

mcoltreegroup Like the **mcoltree** style but the glossary groups have headings.

```
9253 \newglossarystyle{mcoltreegroup}{%
  Base it on the glostylemcoltree style:
9254   \setglossarystyle{mcoltree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
9255 \renewcommand{\glsgroupheading}[1]{\par
9256 \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
9257 }
```

ltreehypergroup The `mcoltreehypergroup` style is like the `treegroup` style, but has a set of links to the groups at the start of the glossary.

```
9258 \newglossarystyle{mcoltreehypergroup}{%
```

Base it on the `glostylemcoltree` style:

```
9259 \setglossarystyle{mcoltree}{%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
9260 \renewcommand*{\glossaryheader}{%
9261 \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
9262 \renewcommand*{\glsgroupheading}[1]{%
9263 \par\noindent
9264 \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9265 \indexspace}%
9266 }
```

mcoltreespannav Similar to the `mcoltreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
9267 \newglossarystyle{mcoltreespannav}{%
9268 \setglossarystyle{tree}%
9269 \renewenvironment{theglossary}%
9270 {%
9271 \begin{multicols}{\glsmcols}\noindent\glstreenavigationfmt{\glsnavigation}]
9272 \setlength{\parindent}{0pt}%
9273 \setlength{\parskip}{0pt plus 0.3pt}%
9274 }%
9275 {\end{multicols}}}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
9276 \renewcommand*{\glsgroupheading}[1]{%
9277 \par\noindent
9278 \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9279 \indexspace}%
9280 }
```

mcoltreenoname Multi-column index style. Same as the `treenoname`, but puts the glossary in multiple columns.

```
9281 \newglossarystyle{mcoltreenoname}{%
9282 \setglossarystyle{treenoname}%
9283 \renewenvironment{theglossary}%
9284 {%

```

```

9285     \begin{multicols}{\glsmcols}
9286     \setlength{\parindent}{0pt}%
9287     \setlength{\parskip}{0pt plus 0.3pt}%
9288 }%
9289 {\end{multicols}}%
9290 }

```

treenonamegroup Like the `mcoltreenoname` style but the glossary groups have headings.

```

9291 \newglossarystyle{mcoltreenonamegroup}{%
    Base it on the glostylemcoltreenoname style:
9292 \setglossarystyle{mcoltreenoname}%
    Give each group a heading:
9293 \renewcommand{\glsgroupheading}[1]{\par
9294 \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
9295 }

```

onamehypergroup The `mcoltreenonamehypergroup` style is like the `mcoltreenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```

9296 \newglossarystyle{mcoltreenonamehypergroup}{%
    Base it on the glostylemcoltreenoname style:
9297 \setglossarystyle{mcoltreenoname}%
    Put navigation links to the groups at the start of the theglossary environment:
9298 \renewcommand*{\glossaryheader}{%
9299 \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    Each group has a heading (in bold with a target) followed by a vertical gap):
9300 \renewcommand*{\glsgroupheading}[1]{%
9301 \par\noindent
9302 \glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
9303 \indexspace}%
9304 }

```

eenonamespannav Similar to the `mcoltreenonamehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

9305 \newglossarystyle{mcoltreenonamespannav}{%
9306 \setglossarystyle{treenoname}%
9307 \renewenvironment{theglossary}%
9308 {%
9309 \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
9310 \setlength{\parindent}{0pt}%
9311 \setlength{\parskip}{0pt plus 0.3pt}%
9312 }%
9313 {\end{multicols}}%
    Each group has a heading (in bold with a target) followed by a vertical gap):
9314 \renewcommand*{\glsgroupheading}[1]{%
9315 \par\noindent

```

```

9316     \glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgrouptitle{##1}}}\par
9317     \indexspace}%
9318 }

```

mcolalttree Multi-column index style. Same as the alttree, but puts the glossary in multiple columns.

```

9319 \newglossarystyle{mcolalttree}{%
9320   \setglossarystyle{alttree}%
9321   \renewenvironment{theglossary}%
9322   {%
9323     \begin{multicols}{\glsmcols}
9324     \def\@gls@prevlevel{-1}%
9325     \mbox{}\par
9326   }%
9327   {\par\end{multicols}}}%
9328 }

```

colalttreegroup Like the mcolalttree style but the glossary groups have headings.

```

9329 \newglossarystyle{colalttreegroup}{%
    Base it on the glostylemcolalttree style:
9330   \setglossarystyle{mcolalttree}%
    Give each group a heading.
9331   \renewcommand{\glsgroupheading}[1]{\par
9332     \def\@gls@prevlevel{-1}%
9333     \hangindent0pt\relax
9334     \parindent0pt\relax
9335     \glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
9336 }

```

treehypergroup The mcolalttreehypergroup style is like the mcolalttreegroup style, but has a set of links to the groups at the start of the glossary.

```

9337 \newglossarystyle{mcolalttreehypergroup}{%
    Base it on the glostylemcolalttree style:
9338   \setglossarystyle{mcolalttree}%
    Put the navigation links in the header
9339   \renewcommand*{\glossaryheader}{%
9340     \par
9341     \def\@gls@prevlevel{-1}%
9342     \hangindent0pt\relax
9343     \parindent0pt\relax
9344     \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    Put a hypertarget at the start of each group
9345   \renewcommand*{\glsgroupheading}[1]{%
9346     \par
9347     \def\@gls@prevlevel{-1}%
9348     \hangindent0pt\relax

```

```

9349 \parindent0pt\relax
9350 \glstreegroupheaderfmt{\glshnavhypertarget{##1}{\glsggetgrouptitle{##1}}}\par
9351 \indexspace}%
9352 }

```

`lalttreespannav` Similar to the `mcolalttreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

9353 \newglossarystyle{mcolalttreespannav}{%
9354 \setglossarystyle{alttree}%
9355 \renewenvironment{theglossary}%
9356 {%
9357 \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
9358 \def\@gls@prevlevel{-1}%
9359 \mbox{}\par
9360 }%
9361 {\par\end{multicols}}}%

```

Put a hypertarget at the start of each group

```

9362 \renewcommand*{\glsgroupheading}[1]{%
9363 \par
9364 \def\@gls@prevlevel{-1}%
9365 \hangindent0pt\relax
9366 \parindent0pt\relax
9367 \glstreegroupheaderfmt{\glshnavhypertarget{##1}{\glsggetgrouptitle{##1}}}\par
9368 \indexspace}%
9369 }

```

3.8 Glossary Styles using supertabular environment (glossary-super package)

The glossary styles defined in the package use the `supertabular` environment.

```

9370 \ProvidesPackage{glossary-super}[2021/09/20 v4.47 (NLCT)]

```

Requires the package:

```

9371 \RequirePackage{supertabular}

```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined if `has` has been loaded.

```

9372 \@ifundefined{glsdescwidth}{%
9373 \newlength{glsdescwidth}
9374 \setlength{glsdescwidth}{0.6\hsize}
9375 }{}

```

`lspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if `has` has been loaded.

```

9376 \@ifundefined{glspagelistwidth}{%
9377 \newlength{glspagelistwidth}
9378 \setlength{glspagelistwidth}{0.1\hsize}

```

9379 }{}

super The super glossary style uses the supertabular environment (it uses lengths defined in the package.)

9380 \newglossarystyle{super}{%

Put the glossary in a supertabular environment with two columns and no head or tail:

```
9381 \renewenvironment{theglossary}%
9382 {\tablehead{}\tabletail{}}%
9383 \begin{supertabular}{lp{\glsdescwidth}}%
9384 {\end{supertabular}}%
```

Do nothing at the start of the table:

9385 \renewcommand*{\glossaryheader}{}%

No group headings:

9386 \renewcommand*{\glsgroupheading}[1]{}%

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
9387 \renewcommand{\glossentry}[2]{%
9388 \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9389 \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
9390 }%
```

Sub entries put in a row (no name, description and page list in second column):

```
9391 \renewcommand{\subglossentry}[3]{%
9392 &
9393 \glssubentryitem{##2}%
9394 \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
9395 ##3\tabularnewline
9396 }%
```

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>)

```
9397 \ifglsnogroupskip
9398 \renewcommand*{\glsgroupskip}{}%
9399 \else
9400 \renewcommand*{\glsgroupskip}{& \tabularnewline}%
9401 \fi
9402 }
```

superborder The superborder style is like the above, but with horizontal and vertical lines:

9403 \newglossarystyle{superborder}{%

Base it on the glostylesuper style:

9404 \setglossarystyle{super}%

Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:

```
9405 \renewenvironment{theglossary}%
9406 {\tablehead{\hline}\tabletail{\hline}%
```

```

9407     \begin{supertabular}{|l|p{\glsdescwidth}|}%
9408     {\end{supertabular}}%
9409 }

```

superheader The superheader style is like the super style, but with a header:

```

9410 \newglossarystyle{superheader}{%
    Base it on the glostylesuper style:
9411   \setglossarystyle{super}%
    Put the glossary in a supertabular environment with two columns, a header and no tail:
9412 \renewenvironment{theglossary}%
9413   {\tablehead{\bfseries \entryname &
9414     \bfseries\descriptionname\tabularnewline}%
9415   \tabletail{}}%
9416   \begin{supertabular}{lp{\glsdescwidth}}%
9417   {\end{supertabular}}%
9418 }

```

superheaderborder The superheaderborder style is like the super style but with a header and border:

```

9419 \newglossarystyle{superheaderborder}{%
    Base it on the glostylesuper style:
9420   \setglossarystyle{super}%
    Put the glossary in a supertabular environment with two columns, a header and horizontal
    lines above and below the table:
9421 \renewenvironment{theglossary}%
9422   {\tablehead{\hline\bfseries \entryname &
9423     \bfseries \descriptionname\tabularnewline\hline}%
9424   \tabletail{\hline}
9425   \begin{supertabular}{|l|p{\glsdescwidth}|}%
9426   {\end{supertabular}}%
9427 }

```

super3col The super3col style is like the super style, but with 3 columns:

```

9428 \newglossarystyle{super3col}{%
    Put the glossary in a supertabular environment with three columns and no head or tail:
9429 \renewenvironment{theglossary}%
9430   {\tablehead{}\tabletail{}}%
9431   \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
9432   {\end{supertabular}}%
    Do nothing at the start of the table:
9433 \renewcommand*{\glossaryheader}{}%
    No group headings:
9434 \renewcommand*{\glsgroupheading}[1]{}%

```


Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```

9435 \renewcommand{\glossentry}[2]{%
9436   \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9437   \glossentrydesc{##1} & ##2\tabularnewline
9438 }%
```

Sub entries on a row (no name, description in second column, page list in last column):

```

9439 \renewcommand{\subglossentry}[3]{%
9440   &
9441   \glssubentryitem{##2}%
9442   \glstarget{##2}{\strut}\glossentrydesc{##2} &
9443   ##3\tabularnewline
9444 }%
```

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>)

```

9445 \ifglsgroupskip
9446   \renewcommand*{\glsgroupskip}{}%
9447 \else
9448   \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9449 \fi
9450 }
```

super3colborder The super3colborder style is like the super3col style, but with a border:

```

9451 \newglossarystyle{super3colborder}{%
```

Base it on the glostylesuper3col style:

```

9452 \setglossarystyle{super3col}%
```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```

9453 \renewenvironment{theglossary}%
9454   {\tablehead{\hline}\tabletail{\hline}%
9455   \begin{supertabular}{|l|p{\glsgdescwidth}|p{\glspagelistwidth|}}%
9456   {\end{supertabular}}%
9457 }
```

super3colheader The super3colheader style is like the super3col style but with a header row:

```

9458 \newglossarystyle{super3colheader}{%
```

Base it on the glostylesuper3col style:

```

9459 \setglossarystyle{super3col}%
```

Put the glossary in a supertabular environment with three columns, a header and no tail:

```

9460 \renewenvironment{theglossary}%
9461   {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9462   \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9463   \begin{supertabular}{lp{\glsgdescwidth}p{\glspagelistwidth}}}%
9464   {\end{supertabular}}%
9465 }
```

colheaderborder The super3colheaderborder style is like the super3col style but with a header and border:

```
9466 \newglossarystyle{super3colheaderborder}{%
```

Base it on the glostylesuper3colborder style:

```
9467 \setglossarystyle{super3colborder}{%
```

Put the glossary in a supertabular environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
9468 \renewenvironment{theglossary}{%
9469   {\tablehead{\hline
9470     \bfseries\entryname&\bfseries\descriptionname&
9471     \bfseries\pagelistname\tabularnewline\hline}%
9472   \tabletail{\hline}%
9473   \begin{supertabular}{|l|p{\glsgdescwidth}|p{\glspagelistwidth}|}%
9474   {\end{supertabular}}}%
9475 }
```

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.

```
9476 \newglossarystyle{super4col}{%
```

Put the glossary in a supertabular environment with four columns and no head or tail:

```
9477 \renewenvironment{theglossary}{%
9478   {\tablehead{}\tabletail{}}%
9479   \begin{supertabular}{|l|l|l|l|}%
9480   \end{supertabular}}%
```

Do nothing at the start of the table:

```
9481 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9482 \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:

```
9483 \renewcommand{\glossentry}[2]{%
9484   \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9485   \glossentrydesc{##1} &
9486   \glossentrysymbol{##1} & ##2\tabularnewline
9487   }%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
9488 \renewcommand{\subglossentry}[3]{%
9489   &
9490   \glssubentryitem{##2}%
9491   \glstarget{##2}{\strut}\glossentrydesc{##2} &
9492   \glossentrysymbol{##2} & ##3\tabularnewline
9493   }%
```

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>)

```

9494 \ifglsgroupskip
9495 \renewcommand*{\glsgroupskip}{}%
9496 \else
9497 \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9498 \fi
9499 }

```

super4colheader The super4colheader style is like the super4col but with a header row.

```

9500 \newglossarystyle{super4colheader}{%
    Base it on the glostylesuper4col style:
9501 \setglossarystyle{super4col}%
    Put the glossary in a supertabular environment with four columns, a header and no tail:
9502 \renewenvironment{theglossary}%
9503 {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9504 \bfseries\symbolname &
9505 \bfseries\pagelistname\tabularnewline}%
9506 \tabletail{}}%
9507 \begin{supertabular}{l111}}%
9508 {\end{supertabular}}%
9509 }

```

super4colborder The super4colborder style is like the super4col but with a border.

```

9510 \newglossarystyle{super4colborder}{%
    Base it on the glostylesuper4col style:
9511 \setglossarystyle{super4col}%
    Put the glossary in a supertabular environment with four columns and a horizontal line in the
    head and tail:
9512 \renewenvironment{theglossary}%
9513 {\tablehead{\hline}\tabletail{\hline}%
9514 \begin{supertabular}{|l111|11|}}%
9515 {\end{supertabular}}%
9516 }

```

colheaderborder The super4colheaderborder style is like the super4col but with a header and border.

```

9517 \newglossarystyle{super4colheaderborder}{%
    Base it on the glostylesuper4col style:
9518 \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:
9519 \renewenvironment{theglossary}%
9520 {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
9521 \bfseries\symbolname &

```

```

9522      \bfseries\pagelistname\tabularnewline\hline}%
9523      \tabletail{\hline}%
9524      \begin{supertabular}{|l|l|l|l|}%
9525      {\end{supertabular}}}%
9526 }

```

altsuper4col The altsuper4col glossary style is like super4col but has provision for multiline descriptions.

```

9527 \newglossarystyle{altsuper4col}{%

```

Base it on the glostylessuper4col style:

```

9528 \setglossarystyle{super4col}%

```

Put the glossary in a supertabular environment with four columns and no head or tail:

```

9529 \renewenvironment{theglossary}%
9530 {\tablehead{}\tabletail{}}%
9531 \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}%
9532 {\end{supertabular}}%
9533 }

```

super4colheader The altsuper4colheader style is like the altsuper4col but with a header row.

```

9534 \newglossarystyle{altsuper4colheader}{%

```

Base it on the glostylessuper4colheader style:

```

9535 \setglossarystyle{super4colheader}%

```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```

9536 \renewenvironment{theglossary}%
9537 {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9538 \bfseries\symbolname &
9539 \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9540 \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}%
9541 {\end{supertabular}}%
9542 }

```

super4colborder The altsuper4colborder style is like the altsuper4col but with a border.

```

9543 \newglossarystyle{altsuper4colborder}{%

```

Base it on the glostylessuper4colborder style:

```

9544 \setglossarystyle{super4colborder}%

```

Put the glossary in a supertabular environment with four columns and a horizontal line in the head and tail:

```

9545 \renewenvironment{theglossary}%
9546 {\tablehead{\hline}\tabletail{\hline}%
9547 \begin{supertabular}%
9548 {lp{\glsgdescwidth}lp{\glspagelistwidth}}%
9549 {\end{supertabular}}%
9550 }

```

colheaderborder The altsuper4colheaderborder style is like the altsuper4col but with a header and border.

```

9551 \newglossarystyle{altsuper4colheaderborder}{%

```

Base it on the `glostylessuper4colheaderborder` style:

```
9552 \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:

```
9553 \renewenvironment{theglossary}%  
9554 {\tablehead{\hline  
9555   \bfseries\entryname &  
9556   \bfseries\descriptionname &  
9557   \bfseries\symbolname &  
9558   \bfseries\pagelistname\tabularnewline\hline}%  
9559 \tabletail{\hline}%  
9560 \begin{supertabular}%  
9561   {lllp{\glsdescwidth}llp{\glspagelistwidth}}}%  
9562 {\end{supertabular}}%  
9563 }
```

3.9 Glossary Styles using `supertabular` environment (`glossary-superragged` package)

The glossary styles defined in the package use the `supertabular` environment. These styles are like those provided by the package, except that the multiline columns have ragged right justification.

```
9564 \ProvidesPackage{glossary-superragged}[2021/09/20 v4.47 (NLCT)]
```

Requires the package:

```
9565 \RequirePackage{array}
```

Requires the package:

```
9566 \RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```
9567 \@ifundefined{glsdescwidth}{%  
9568   \newlength\glsdescwidth  
9569   \setlength{\glsdescwidth}{0.6\hsize}  
9570 }{}
```

`glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
9571 \@ifundefined{glspagelistwidth}{%  
9572   \newlength\glspagelistwidth  
9573   \setlength{\glspagelistwidth}{0.1\hsize}  
9574 }{}
```

`superragged` The `superragged` glossary style uses the `supertabular` environment.

```
9575 \newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
9576 \renewenvironment{theglossary}%
9577   {\tablehead{}\tabletail{}}%
9578   \begin{supertabular}{l>{\raggedright}p{\glsgdescwidth}}}%
9579   {\end{supertabular}}%
```

Do nothing at the start of the table:

```
9580 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9581 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
9582 \renewcommand{\glossentry}[2]{%
9583   \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9584   \glossentrydesc{##1}\glspostdescription\space ##2%
9585   \tabularnewline
9586 }%
```

Sub entries put in a row (no name, description and page list in second column):

```
9587 \renewcommand{\subglossentry}[3]{%
9588   &
9589   \glssubentryitem{##2}%
9590   \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
9591   ##3%
9592   \tabularnewline
9593 }%
```

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>)

```
9594 \ifglsgroupskip
9595   \renewcommand*{\glsgroupskip}{}%
9596 \else
9597   \renewcommand*{\glsgroupskip}{& \tabularnewline}%
9598 \fi
9599 }
```

`\perraggedborder` The `superraggedborder` style is like the above, but with horizontal and vertical lines:

```
9600 \newglossarystyle{superraggedborder}{%
```

Base it on the `glostylesuperragged` style:

```
9601 \setglossarystyle{superragged}%
```

Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:

```
9602 \renewenvironment{theglossary}%
9603   {\tablehead{\hline}\tabletail{\hline}%
9604   \begin{supertabular}{l|l>{\raggedright}p{\glsgdescwidth}|}%
9605   {\end{supertabular}}%
9606 }
```

`superraggedheader` The `superraggedheader` style is like the `super` style, but with a header:

```
9607 \newglossarystyle{superraggedheader}{%
```

Base it on the `glostylesuperragged` style:

```
9608 \setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and no tail:

```
9609 \renewenvironment{theglossary}%
```

```
9610 {\tablehead{\bfseries \entryname & \bfseries \descriptionname
```

```
9611 \tabularnewline}%
```

```
9612 \tabletail{}}%
```

```
9613 \begin{supertabular}{l>{\raggedright}p{\glstdescwidth}}%
```

```
9614 {\end{supertabular}}%
```

```
9615 }
```

`superraggedheaderborder` The `superraggedheaderborder` style is like the `superragged` style but with a header and border:

```
9616 \newglossarystyle{superraggedheaderborder}{%
```

Base it on the `glostylesuper` style:

```
9617 \setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and horizontal lines above and below the table:

```
9618 \renewenvironment{theglossary}%
```

```
9619 {\tablehead{\hline\bfseries \entryname &
```

```
9620 \bfseries \descriptionname\tablearnewline\hline}%
```

```
9621 \tabletail{\hline}
```

```
9622 \begin{supertabular}{ll|>{\raggedright}p{\glstdescwidth}}%
```

```
9623 {\end{supertabular}}%
```

```
9624 }
```

`superragged3col` The `superragged3col` style is like the `superragged` style, but with 3 columns:

```
9625 \newglossarystyle{superragged3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
9626 \renewenvironment{theglossary}%
```

```
9627 {\tablehead{}\tabletail{}}%
```

```
9628 \begin{supertabular}{l>{\raggedright}p{\glstdescwidth}%
```

```
9629 >{\raggedright}p{\glspagelistwidth}}%
```

```
9630 {\end{supertabular}}%
```

Do nothing at the start of the table:

```
9631 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9632 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
9633 \renewcommand{\glossentry}[2]{%
```

```
9634 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
```

```
9635 \glossentrydesc{##1} &
```

```

9636     ##2\tabularnewline
9637 }%

```

Sub entries on a row (no name, description in second column, page list in last column):

```

9638 \renewcommand{\subglossentry}[3]{%
9639     &
9640     \glssubentryitem{##2}%
9641     \glstarget{##2}{\strut}\glossentrydesc{##2} &
9642     ##3\tabularnewline
9643 }%

```

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>)

```

9644 \ifglsnogroupskip
9645 \renewcommand*{\glsgroupskip}{}%
9646 \else
9647 \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9648 \fi
9649 }

```

agged3colborder The superragged3colborder style is like the superragged3col style, but with a border:

```

9650 \newglossarystyle{superragged3colborder}{%

```

Base it on the glostylesuperragged3col style:

```

9651 \setglossarystyle{superragged3col}%

```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```

9652 \renewenvironment{theglossary}%
9653 {\tablehead{\hline}\tabletail{\hline}%
9654 \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||%
9655 >{\raggedright}p{\glspagelistwidth}||}%
9656 {\end{supertabular}}%
9657 }

```

agged3colheader The superragged3colheader style is like the superragged3col style but with a header row:

```

9658 \newglossarystyle{superragged3colheader}{%

```

Base it on the glostylesuperragged3col style:

```

9659 \setglossarystyle{superragged3col}%

```

Put the glossary in a supertabular environment with three columns, a header and no tail:

```

9660 \renewenvironment{theglossary}%
9661 {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9662 \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9663 \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
9664 >{\raggedright}p{\glspagelistwidth}}%
9665 {\end{supertabular}}%
9666 }

```


colheaderborder The superragged3colheaderborder style is like the superragged3col style but with a header and border:

```
9667 \newglossarystyle{superragged3colheaderborder}{%
```

Base it on the glostylesuperragged3colborder style:

```
9668 \setglossarystyle{superragged3colborder}{%
```

Put the glossary in a supertabular environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
9669 \renewenvironment{theglossary}{%
9670   {\tablehead{\hline
9671     \bfseries\entryname&\bfseries\descriptionname&
9672     \bfseries\pagelistname\tabularnewline\hline}%
9673   \tabletail{\hline}%
9674   \begin{supertabular}{|l|>{\raggedright}p{\glstdescwidth}|%
9675     >{\raggedright}p{\glspagelistwidth}|}%
9676   {\end{supertabular}}}%
9677 }
```

superragged4col The altsuperragged4col glossary style is like altsuper4col style in the package but uses ragged right formatting in the description and page list columns.

```
9678 \newglossarystyle{altsuperragged4col}{%
```

Put the glossary in a supertabular environment with four columns and no head or tail:

```
9679 \renewenvironment{theglossary}{%
9680   {\tablehead{}\tabletail{}%
9681   \begin{supertabular}{l>{\raggedright}p{\glstdescwidth}l%
9682     >{\raggedright}p{\glspagelistwidth}}}%
9683   {\end{supertabular}}}%

```

Do nothing at the start of the table:

```
9684 \renewcommand*{\glossaryheader}{}%

```

No group headings:

```
9685 \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:

```
9686 \renewcommand{\glossentry}[2]{%
9687   \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9688   \glossentrydesc{##1} &
9689   \glossentrysymbol{##1} & ##2\tabularnewline
9690   }%

```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
9691 \renewcommand{\subglossentry}[3]{%
9692   &
9693   \glssubentryitem{##2}%
9694   \glstarget{##2}{\strut}\glossentrydesc{##2} &
9695   \glossentrysymbol{##2} & ##3\tabularnewline
9696   }%

```

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>)

```
9697 \ifglsgroupskip
9698 \renewcommand*{\glsgroupskip}{}%
9699 \else
9700 \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9701 \fi
9702 }
```

agged4colheader The altsuperragged4colheader style is like the altsuperragged4col style but with a header row.

```
9703 \newglossarystyle{altsuperragged4colheader}{%
```

Base it on the glostylealtsuperragged4col style:

```
9704 \setglossarystyle{altsuperragged4col}{%
```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```
9705 \renewenvironment{theglossary}%
9706 {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9707 \bfseries\symbolname &
9708 \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9709 \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
9710 >{\raggedright}p{\glspagelistwidth}}}%
9711 {\end{supertabular}}}%
9712 }
```

agged4colborder The altsuperragged4colborder style is like the altsuperragged4col style but with a border.

```
9713 \newglossarystyle{altsuperragged4colborder}{%
```

Base it on the glostylealtsuperragged4col style:

```
9714 \setglossarystyle{altsuper4col}{%
```

Put the glossary in a supertabular environment with four columns and a horizontal line in the head and tail:

```
9715 \renewenvironment{theglossary}%
9716 {\tablehead{\hline}\tabletail{\hline}%
9717 \begin{supertabular}%
9718 {l|>{\raggedright}p{\glsdescwidth}l|}%
9719 >{\raggedright}p{\glspagelistwidth}l|}%
9720 {\end{supertabular}}}%
9721 }
```

colheaderborder The altsuperragged4colheaderborder style is like the altsuperragged4col style but with a header and border.

```
9722 \newglossarystyle{altsuperragged4colheaderborder}{%
```

Base it on the glostylealtsuperragged4col style:

```
9723 \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:

```

9724 \renewenvironment{theglossary}%
9725   {\tablehead{\hline
9726     \bfseries\entryname &
9727     \bfseries\descriptionname &
9728     \bfseries\symbolname &
9729     \bfseries\pagelistname\tabularnewline\hline}%
9730   \tabletail{\hline}%
9731   \begin{supertabular}%
9732     {|||>\raggedright}p{\glsdescwidth}|||%
9733     >\raggedright}p{\glspagelistwidth}{|}}%
9734   {\end{supertabular}}%
9735 }

```

3.10 Tree Styles (glossary-tree.sty)

The style file defines glossary styles that have a tree-like structure. These are designed for hierarchical glossaries.

```

9736 \ProvidesPackage{glossary-tree}[2021/09/20 v4.47 (NLCT)]

```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```

9737 \providecommand{\indexspace}{%
9738   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
9739 }

```

`\glstreenamefmt` Format used to display the name in the tree styles. (This may be counteracted by `\glstnamefont`.) This command was previously also used to format the group headings.

```

9740 \newcommand*{\glstreenamefmt}[1]{\textbf{#1}}

```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamefmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamefmt` would've also affected the group headings.

```

9741 \newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}

```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```

9742 \newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}

```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```

9743 \ifdef\@idxitem
9744 {\newcommand{\glstreeitem}{\@idxitem}}
9745 {\newcommand{\glstreeitem}{\par\hangindent40\p@}}

```

`\glstreesubitem` Level 1 item used in index style.

```
9746 \ifdef\subitem
9747 {\let\glstreesubitem\subitem}
9748 {\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
```

`\glstreesubsubitem` Level 1 item used in index style.

```
9749 \ifdef\subsubitem
9750 {\let\glstreesubsubitem\subsubitem}
9751 {\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `alttree` style).

```
9752 \newcommand{\glstreepredesc}{\space}
```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treenoname` and `alttree` style).

```
9753 \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.

```
9754 \newglossarystyle{index}{%
```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```
9755 \renewenvironment{theglossary}%
9756 {\setlength{\parindent}{0pt}}%
9757 {\setlength{\parskip}{0pt plus 0.3pt}}%
9758 \let\item\glstreeitem
9759 \let\subitem\glstreesubitem
9760 \let\subsubitem\glstreesubsubitem
9761 }%
```

```
9762 {\par}}%
```

Do nothing at the start of the environment:

```
9763 \renewcommand*{\glossaryheader}{}%
```

No group headers:

```
9764 \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.

```
9765 \renewcommand*{\glossentry}[2]{%
9766 \item\glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9767 \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9768 \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
9769 }%
```

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.

```

9770 \renewcommand{\subglossentry}[3]{%
9771   \ifcase##1\relax
9772     % level 0
9773     \item
9774   \or
9775     % level 1
9776     \subitem
9777     \glssubentryitem{##2}%
9778   \else
9779     % all other levels
9780     \subsubitem
9781   \fi
9782   \glstreenamfmt{\glstarget{##2}{\glossentryname{##2}}}%
9783   \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
9784   \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
9785 }%
```

Vertical gap between groups is the same as that used by indices:

```

9786 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
```

indexgroup The `indexgroup` style is like the `index` style but has headings.

```

9787 \newglossarystyle{indexgroup}{%
```

Base it on the `glostyleindex` style:

```

9788 \setglossarystyle{index}%
```

Add a heading for each group. This puts the group's title in bold followed by a vertical gap.

```

9789 \renewcommand*{\glsgroupheading}[1]{%
9790   \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
9791   \indexspace
9792 }%
9793 }
```

indexhypergroup The `indexhypergroup` style is like the `indexgroup` style but has hyper navigation.

```

9794 \newglossarystyle{indexhypergroup}{%
```

Base it on the `glostyleindex` style:

```

9795 \setglossarystyle{index}%
```

Put navigation links to the groups at the start of the glossary:

```

9796 \renewcommand*{\glossaryheader}{%
9797   \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.

```

9798 \renewcommand*{\glsgroupheading}[1]{%
9799   \item\glstreegroupheaderfmt

```

```

9800      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
9801      \indexspace}%
9802 }

```

tree The tree glossary style is similar in style to the index style, but can have arbitrary levels.

```

9803 \newglossarystyle{tree}{%

```

Set the paragraph indentation and skip:

```

9804   \renewenvironment{theglossary}%
9805     {\setlength{\parindent}{0pt}%
9806     \setlength{\parskip}{0pt plus 0.3pt}}%
9807     {}%

```

Do nothing at the start of the theglossary environment:

```

9808   \renewcommand*{\glossaryheader}{}%

```

No group headings:

```

9809   \renewcommand*{\glsgroupheading}[1]{}%

```

Main (level 0) entries: name in bold, followed by symbol in brackets (if it exists), the description and the page list:

```

9810   \renewcommand{\glossentry}[2]{%
9811     \hangindent0pt\relax
9812     \parindent0pt\relax
9813     \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9814     \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9815     \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
9816   }%

```

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.

```

9817   \renewcommand{\subglossentry}[3]{%
9818     \hangindent##1\glstreeindent\relax
9819     \parindent##1\glstreeindent\relax
9820     \ifnum##1=1\relax
9821       \glssubentryitem{##2}%
9822       \fi
9823       \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
9824       \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
9825       \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
9826   }%

```

Vertical gap between groups is the same as that used by indices:

```

9827   \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}

```

treegroup Like the tree style but the glossary groups have headings.

```

9828 \newglossarystyle{treegroup}{%

```

Base it on the glostyletree style:

```

9829   \setglossarystyle{tree}%

```

Each group has a heading (in bold) followed by a vertical gap):

```
9830 \renewcommand{\glsgroupheading}[1]{\par
9831 \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
9832 \indexspace}%
9833 }
```

treehypergroup The **treehypergroup** style is like the **treegroup** style, but has a set of links to the groups at the start of the glossary.

```
9834 \newglossarystyle{treehypergroup}{%
```

Base it on the **glostyletree** style:

```
9835 \setglossarystyle{tree}%
```

Put navigation links to the groups at the start of the **theglossary** environment:

```
9836 \renewcommand*{\glossaryheader}{%
9837 \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
9838 \renewcommand*{\glsgroupheading}[1]{%
9839 \par\noindent
9840 \glstreegroupheaderfmt
9841 {\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
9842 \indexspace}%
9843 }
```

\glstreeindent Length governing left indent for each level of the tree style.

```
9844 \newlength\glstreeindent
9845 \setlength{\glstreeindent}{10pt}
```

treenoname The **treenoname** glossary style is like the **tree** style, but doesn't print the name or symbol for sub-levels.

```
9846 \newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
9847 \renewenvironment{theglossary}%
9848 {\setlength{\parindent}{0pt}%
9849 \setlength{\parskip}{0pt plus 0.3pt}}%
9850 {}%
```

No header:

```
9851 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9852 \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.

```
9853 \renewcommand{\glossentry}[2]{%
9854 \hangindent0pt\relax
9855 \parindent0pt\relax
9856 \glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
```

```

9857 \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9858 \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
9859 }%

```

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.

```

9860 \renewcommand{\subglossentry}[3]{%
9861   \hangindent##1\glstreeindent\relax
9862   \parindent##1\glstreeindent\relax
9863   \ifnum##1=1\relax
9864     \glssubentryitem{##2}%
9865   \fi
9866   \glstarget{##2}{\strut}%
9867   \glossentrydesc{##2}\glspostdescription\space##3\par
9868 }%

```

Vertical gap between groups is the same as that used by indices:

```

9869 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
9870 }

```

`treenonamegroup` Like the `treenoname` style but the glossary groups have headings.

```

9871 \newglossarystyle{treenonamegroup}{%
  Base it on the glostyletreenoname style:
9872 \setglossarystyle{treenoname}%
  Give each group a heading:
9873 \renewcommand{\glsgroupheading}[1]{\par
9874   \noindent\glstreegroupheaderfmt
9875   {\glsgrouptitle{##1}}\par\indexspace}%
9876 }

```

`onamehypergroup` The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```

9877 \newglossarystyle{treenonamehypergroup}{%
  Base it on the glostyletreenoname style:
9878 \setglossarystyle{treenoname}%

```

Put navigation links to the groups at the start of the `theglossary` environment:

```

9879 \renewcommand*{\glossaryheader}{%
9880   \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```

9881 \renewcommand*{\glsgroupheading}[1]{%
9882   \par\noindent
9883   \glstreegroupheaderfmt
9884   {\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9885   \indexspace}%
9886 }

```


`esttoplevelname` Find the widest name over all parentless entries in the given glossary or glossaries.

```

9887 \newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
9888   \dimen@=0pt\relax
9889   \gls@tmplen=0pt\relax
9890   \forallglossaries[#1]{\@gls@type}%
9891   {%
9892     \forallglsentries[\@gls@type]{\@glo@label}%
9893     {%
9894       \ifglsahasparent{\@glo@label}%
9895       }%
9896       {%
9897         \settowidth{\dimen@}%
9898           {\glstreenamfmt{\glsentryname{\@glo@label}}}%
9899         \ifdim\dimen@>\gls@tmplen
9900           \gls@tmplen=\dimen@
9901           \letcs{\@glswidestname}{glo\glsdetoklabel{\@glo@label}@name}%
9902         \fi
9903       }%
9904     }%
9905   }%
9906 }
```

`\glssetwidest` `\glssetwidest[⟨level⟩]{⟨text⟩}` sets the widest text for the given level. It is used by the alt-tree glossary styles to determine the indentation of each level.

```

9907 \newcommand*{\glssetwidest}[2][0]{%
9908   \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
9909     #2}%
9910 }
```

`\@glswidestname` Initialise `\@glswidestname`.

```

9911 \newcommand*{\@glswidestname}{}
```

`\glstreenamebox` Used by the alttree style to create the box for the name and associated information.

```

9912 \newcommand*{\glstreenamebox}[2]{%
9913   \makebox[#1][l]{#2}%
9914 }
```

`alttree` The alttree glossary style is similar in style to the tree style, but the indentation is obtained from the width of `\@glswidestname` which is set using `\glssetwidest`.

```

9915 \newglossarystyle{alttree}{%
```

Redefine theglossary environment.

```

9916 \renewenvironment{theglossary}%
9917   {\def\@gls@prevlevel{-1}%
9918    \mbox{}\par}%
9919   {\par}%

```

Set the header and group headers to nothing.

```

9920 \renewcommand*{\glossaryheader}{}%
9921 \renewcommand*{\glsgroupheading}[1]{}%

```

Redefine the way that the level 0 entries are displayed.

```
9922 \renewcommand{\glossentry}[2]{%
9923   \ifnum\@gls@prevlevel=0\relax
9924   \else
```

Find out how big the indentation should be by measuring the widest entry.

```
9925     \settowidth{\glstreeindent}{\glstreenamfmt{\@glswidestname\space}}%
9926   \fi
```

Set the hangindent and paragraph indent.

```
9927   \hangindent\glstreeindent
9928   \parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
9929   \makebox[0pt][r]{\glstreenambox{\glstreeindent}{%
9930     \glssentryitem{##1}\glstreenamfmt{\glstarget{##1}{\glossentryname{##1}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
9931   \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
9932   \glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
9933   \def\@gls@prevlevel{0}%
9934 }%
```

Redefine the way sub-entries are displayed.

```
9935 \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.

```
9936   \ifnum##1=1\relax
9937     \glssubentryitem{##2}%
9938   \fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
9939   \ifnum\@gls@prevlevel=##1\relax
9940   \else
```

Compute the widest entry for this level, or for level 0 if not defined for this level. Store in `\gls@tmplen`

```
9941     \@ifundefined{@glswidestname\romannumeral##1}{%
9942       \settowidth{\gls@tmplen}{\glstreenamfmt{\@glswidestname\space}}{%
9943       \settowidth{\gls@tmplen}{\glstreenamfmt{%
9944         \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
9945   \ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```

9946      \setlength\glstreeindent\gls@tmplen
9947      \addtolength\glstreeindent\parindent
9948      \parindent\glstreeindent
9949      \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`.

```

9950      \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
9951      \settowidth{\glstreeindent}{\glstreenamfmt{%
9952      \@glswidestname\space}}}{%
9953      \settowidth{\glstreeindent}{\glstreenamfmt{%
9954      \csname @glswidestname\romannumeral\@gls@prevlevel
9955      \endcsname\space}}}{%

```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```

9956      \addtolength\parindent{-\glstreeindent}%
9957      \setlength\glstreeindent\parindent
9958      \fi
9959      \fi

```

Set the hanging indentation.

```

9960      \hangindent\glstreeindent

```

Put the name to the left of the paragraph block

```

9961      \makebox[0pt][r]{\glstreenambox{\gls@tmplen}{%
9962      \glstreenamfmt{\glstarget{##2}{\glossentryname{##2}}}}}%

```

If the symbol is missing, ignore it, otherwise put it in brackets.

```

9963      \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%

```

Do the description followed by the description terminator and location list.

```

9964      \glossentrydesc{##2}\glspostdescription\space ##3\par

```

Set the previous level macro to the current level.

```

9965      \def\@gls@prevlevel{##1}%
9966      }%

```

Vertical gap between groups is the same as that used by indices:

```

9967      \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
9968 }

```

`almtreegroup` Like the `almtree` style but the glossary groups have headings.

```

9969 \newglossarystyle{almtreegroup}{%

```

Base it on the `glostylealmtree` style:

```

9970      \setglossarystyle{almtree}%

```

Give each group a heading.

```

9971      \renewcommand{\glsgroupheading}[1]{\par
9972      \def\@gls@prevlevel{-1}%
9973      \hangindent0pt\relax

```

```

9974     \parindent0pt\relax
9975     \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}}%
9976     \par\indexspace}%
9977 }

```

alttreehypergroup The alttreehypergroup style is like the alttreegroup style, but has a set of links to the groups at the start of the glossary.

```

9978 \newglossarystyle{alttreehypergroup}{%
    Base it on the glostylealttree style:
9979     \setglossarystyle{alttree}%
    Put the navigation links in the header
9980     \renewcommand*{\glossaryheader}{%
9981         \par
9982         \def\@gls@prevlevel{-1}%
9983         \hangindent0pt\relax
9984         \parindent0pt\relax
9985         \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    Put a hypertarget at the start of each group
9986     \renewcommand*{\glsgroupheading}[1]{%
9987         \par
9988         \def\@gls@prevlevel{-1}%
9989         \hangindent0pt\relax
9990         \parindent0pt\relax
9991         \glstreegroupheaderfmt
9992         {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
9993         \indexspace}}

```

4 Backwards Compatibility

4.1 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.

```
9994 \NeedsTeXFormat{LaTeX2e}
9995 \ProvidesPackage{glossaries-compatible-207}[2021/09/20 v4.47 (NLCT)]
```

AddXdyAttribute Adds an attribute in old format.

```
9996 \ifglxsindy
9997   \renewcommand*\GlsAddXdyAttribute[1]{%
9998     \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
9999     \expandafter\toks@\expandafter{\@xdylocref}%
10000     \edef\@xdylocref{\the\toks@ ^^J}%
10001     (markup-locref
10002     :open \string"\string~n\string\setentrycounter
10003       {\noexpand\glscounter}%
10004       \expandafter\string\csname#1\endcsname
10005       \expandafter\@gobble\string\{\string" ^^J
10006       :close \string"\expandafter\@gobble\string\}\string" ^^J
10007       :attr \string"#1\string"))}}
```

Only has an effect before `\writeist`:

```
10008 \fi
```

sAddXdyCounters

```
10009 \renewcommand*\GlsAddXdyCounters[1]{%
10010   \GlossariesWarning{\string\GlsAddXdyCounters\space not available
10011     in compatibility mode.}%
10012 }
```

Add predefined attributes

```
10013 \GlsAddXdyAttribute{glsnumberformat}
10014 \GlsAddXdyAttribute{textrm}
10015 \GlsAddXdyAttribute{textsf}
10016 \GlsAddXdyAttribute{texttt}
10017 \GlsAddXdyAttribute{textbf}
10018 \GlsAddXdyAttribute{textmd}
10019 \GlsAddXdyAttribute{textit}
10020 \GlsAddXdyAttribute{textup}
10021 \GlsAddXdyAttribute{textsl}
```

```

10022 \GlsAddXdyAttribute{textsc}
10023 \GlsAddXdyAttribute{emph}
10024 \GlsAddXdyAttribute{glshypernumber}
10025 \GlsAddXdyAttribute{hyperrm}
10026 \GlsAddXdyAttribute{hypersf}
10027 \GlsAddXdyAttribute{hypertt}
10028 \GlsAddXdyAttribute{hyperbf}
10029 \GlsAddXdyAttribute{hypermd}
10030 \GlsAddXdyAttribute{hyperit}
10031 \GlsAddXdyAttribute{hyperup}
10032 \GlsAddXdyAttribute{hypersl}
10033 \GlsAddXdyAttribute{hypersc}
10034 \GlsAddXdyAttribute{hyperemph}

```

sAddXdyLocation Restore v2.07 definition:

```

10035 \ifglxindy
10036   \renewcommand*{\GlsAddXdyLocation}[2]{%
10037     \edef\@xdyuserlocationdefs{%
10038       \@xdyuserlocationdefs ^^J%
10039       (define-location-class \string"#1\string"^^J\space\space
10040       \space(#2))
10041     }%
10042     \edef\@xdyuserlocationnames{%
10043       \@xdyuserlocationnames^^J\space\space\space
10044       \string"#1\string"}%
10045   }
10046 \fi

```

\@do@wrglossary

```

10047 \renewcommand{\@do@wrglossary}[1]{%
  Determine whether to use xindy or makeindex syntax
10048 \ifglxindy
  Need to determine if the formatting information starts with a ( or ) indicating a range.
10049   \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
10050   \def\@glo@range{}%
10051   \expandafter\if\@glo@prefix\relax
10052   \def\@glo@range{:open-range}%
10053   \else
10054     \expandafter\if\@glo@prefix\relax
10055     \def\@glo@range{:close-range}%
10056   \fi
10057 \fi

  Get the location and escape any special characters
10058   \protected@edef\@glslocref{\theglentrycounter}%
10059   \@gls@checkmkidxchars\@glslocref

  Write to the glossary file using xindy syntax.
10060   \glossary[\csname glo@#1@type\endcsname]{%

```

```

10061 (indexentry :tkey (\csname glo@#1@index\endcsname)
10062   :locoref \string"\@glslocoref\string" %
10063   :attr \string"\@glo@suffix\string" \@glo@range
10064 )
10065 }%
10066 \else
    Convert the format information into the format required for makeindex
10067   \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
    Write to the glossary file using makeindex syntax.
10068   \glossary[\csname glo@#1@type\endcsname]{%
10069     \string\glossaryentry{\csname glo@#1@index\endcsname
10070       \@gls@encapchar\@glo@numfmt}{\theglsentrycounter}}%
10071 \fi
10072 }

```

t@glo@numformat Only had 3 arguments in v2.07

```

10073 \def\@set@glo@numformat#1#2#3{%
10074   \expandafter\@glo@check@mkidxrangechar#3\@nil
10075   \protected@edef#1{%
10076     \@glo@prefix setentrycounter[] {#2}%
10077     \expandafter\string\csname\@glo@suffix\endcsname
10078   }%
10079   \@gls@checkmkidxchars#1%
10080 }

```

\writeist Redefine \writeist back to the way it was in v2.07, but change \istfile to \glswrite.

```

10081 \ifglxindy
10082   \def\writeist{%
10083     \openout\glswrite=\istfilename
10084     \write\glswrite{;; xindy style file created by the glossaries
10085       package in compatible-2.07 mode}%
10086     \write\glswrite{;; for document '\jobname' on
10087       \the\year-\the\month-\the\day}%
10088     \write\glswrite{^^J; required styles^^J}
10089     \@for\@xdystyle:=\@xdyrequiredstyles\do{%
10090       \ifx\@xdystyle\@empty
10091       \else
10092         \protected@write\glswrite{{(require
10093           \string"\@xdystyle.xdy\string")}}%
10094       \fi
10095     }%
10096     \write\glswrite{^^J%
10097       ; list of allowed attributes (number formats)^^J}%
10098     \write\glswrite{(define-attributes ((\@xdyattributes)))}%
10099     \write\glswrite{^^J; user defined alphabets^^J}%
10100     \write\glswrite{\@xdyuseralphabets}%
10101     \write\glswrite{^^J; location class definitions^^J}%
10102     \protected@edef\@gls@roman{\@roman{0}\string"

```

```

10103     \string"roman-numbers-lowercase\string" :sep \string"}}%
10104 \@onelevel@sanitize\@gls@roman
10105 \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
10106     :sep \string"}}%
10107 \@onelevel@sanitize\@tmp
10108 \ifx\@tmp\@gls@roman
10109     \write\glswrite{(define-location-class
10110         \string"roman-page-numbers\string"^^J\space\space\space
10111         (\string"roman-numbers-lowercase\string")
10112         :min-range-length \@glsminrange)}}%
10113 \else
10114     \write\glswrite{(define-location-class
10115         \string"roman-page-numbers\string"^^J\space\space\space
10116         (:sep "\@gls@roman")
10117         :min-range-length \@glsminrange)}}%
10118 \fi
10119 \write\glswrite{(define-location-class
10120     \string"Roman-page-numbers\string"^^J\space\space\space
10121     (\string"roman-numbers-uppercase\string")
10122     :min-range-length \@glsminrange)}}%
10123 \write\glswrite{(define-location-class
10124     \string"arabic-page-numbers\string"^^J\space\space\space
10125     (\string"arabic-numbers\string")
10126     :min-range-length \@glsminrange)}}%
10127 \write\glswrite{(define-location-class
10128     \string"alpha-page-numbers\string"^^J\space\space\space
10129     (\string"alpha\string")
10130     :min-range-length \@glsminrange)}}%
10131 \write\glswrite{(define-location-class
10132     \string"Alpha-page-numbers\string"^^J\space\space\space
10133     (\string"ALPHA\string")
10134     :min-range-length \@glsminrange)}}%
10135 \write\glswrite{(define-location-class
10136     \string"Appendix-page-numbers\string"^^J\space\space\space
10137     (\string"ALPHA\string"
10138     :sep \string"\@glsAlphacompositor\string"
10139     \string"arabic-numbers\string")
10140     :min-range-length \@glsminrange)}}%
10141 \write\glswrite{(define-location-class
10142     \string"arabic-section-numbers\string"^^J\space\space\space
10143     (\string"arabic-numbers\string"
10144     :sep \string"\glscompositor\string"
10145     \string"arabic-numbers\string")
10146     :min-range-length \@glsminrange)}}%
10147 \write\glswrite{^^J; user defined location classes}%
10148 \write\glswrite{\@xdyuserlocationdefs}%
10149 \write\glswrite{^^J; define cross-reference class^^J}%
10150 \write\glswrite{(define-crossref-class \string"see\string"
10151     :unverified )}%

```



```

10152 \write\glswrite{(markup-crossref-list
10153 :class \string"see\string"^^J\space\space\space
10154 :open \string"\string\glseeformat\string"
10155 :close \string"{}\string")}%
10156 \write\glswrite{^^J; define the order of the location classes}%
10157 \write\glswrite{(define-location-class-order
10158 (\@xdylocationclassorder))}%
10159 \write\glswrite{^^J; define the glossary markup^^J}%
10160 \write\glswrite{(markup-index^^J\space\space\space
10161 :open \string"\string
10162 \glossarysection[\string\glossarytoctitle]{\string
10163 \glossarytitle}\string\glossarypreamble\string~n\string\begin
10164 {theglossary}\string\glossaryheader\string~n\string" ^^J\space
10165 \space\space:close \string"\expandafter\@gobble
10166 \string%\string~n\string
10167 \end{theglossary}\string\glossarypostamble
10168 \string~n\string" ^^J\space\space\space
10169 :tree)}}%
10170 \write\glswrite{(markup-letter-group-list
10171 :sep \string"\string\glsgroupskip\string~n\string")}%
10172 \write\glswrite{(markup-indexentry
10173 :open \string"\string\relax \string\glresetentrylist
10174 \string~n\string")}%
10175 \write\glswrite{(markup-locclass-list :open
10176 \string"\glsoopenbrace\string\glossaryentrynumbers
10177 \glsoopenbrace\string\relax\space \string"^^J\space\space\space
10178 :sep \string", \string"
10179 :close \string"\glsclosebrace\glsclosebrace\string")}%
10180 \write\glswrite{(markup-locref-list
10181 :sep \string"\string\delimN\space\string")}%
10182 \write\glswrite{(markup-range
10183 :sep \string"\string\delimR\space\string")}%
10184 \@onelevel@sanitize\gls@suffixF
10185 \@onelevel@sanitize\gls@suffixFF
10186 \ifx\gls@suffixF\@empty
10187 \else
10188 \write\glswrite{(markup-range
10189 :close "\gls@suffixF" :length 1 :ignore-end)}%
10190 \fi
10191 \ifx\gls@suffixFF\@empty
10192 \else
10193 \write\glswrite{(markup-range
10194 :close "\gls@suffixFF" :length 2 :ignore-end)}%
10195 \fi
10196 \write\glswrite{^^J; define format to use for locations^^J}%
10197 \write\glswrite{\@xdylocref}%
10198 \write\glswrite{^^J; define letter group list format^^J}%
10199 \write\glswrite{(markup-letter-group-list
10200 :sep \string"\string\glsgroupskip\string~n\string")}%

```

```

10201 \write\glswrite{^^J; letter group headings^^J}%
10202 \write\glswrite{(markup-letter-group
10203   :open-head \string"\string\glsgroupheading
10204   \glsoopenbrace\string"^^J\space\space\space
10205   :close-head \string"\glsclosebrace\string")}%
10206 \write\glswrite{^^J; additional letter groups^^J}%
10207 \write\glswrite{\@xdylettergroups}%
10208 \write\glswrite{^^J; additional sort rules^^J}
10209 \write\glswrite{\@xdysortrules}%
10210 \noist}
10211 \else
10212 \edef\@gls@actualchar{\string?}
10213 \edef\@gls@encapchar{\string|}
10214 \edef\@gls@levelchar{\string!}
10215 \edef\@gls@quotechar{\string"}
10216 \def\writeist{\relax
10217   \openout\glswrite=\istfilename
10218   \write\glswrite{\expandafter\@gobble\string\% makeindex style file
10219     created by the glossaries package}
10220   \write\glswrite{\expandafter\@gobble\string\% for document
10221     '\jobname' on \the\year-\the\month-\the\day}
10222   \write\glswrite{actual '\@gls@actualchar'}
10223   \write\glswrite{encap '\@gls@encapchar'}
10224   \write\glswrite{level '\@gls@levelchar'}
10225   \write\glswrite{quote '\@gls@quotechar'}
10226   \write\glswrite{keyword \string"\string\glossaryentry\string"}
10227   \write\glswrite{preamble \string"\string\glossarysection[\string
10228     \glossarytoctitle]{\string\glossarytitle}\string
10229     \glossarypreamble\string\n\string\begin{theglossary}\string
10230     \glossaryheader\string\n\string"}
10231   \write\glswrite{postamble \string"\string%\string\n\string
10232     \end{theglossary}\string\glossarypostamble\string\n
10233     \string"}
10234   \write\glswrite{group_skip \string"\string\glsgroupskip\string\n
10235     \string"}
10236   \write\glswrite{item_0 \string"\string%\string\n\string"}
10237   \write\glswrite{item_1 \string"\string%\string\n\string"}
10238   \write\glswrite{item_2 \string"\string%\string\n\string"}
10239   \write\glswrite{item_01 \string"\string%\string\n\string"}
10240   \write\glswrite{item_x1
10241     \string"\string\relax \string\glsresetentrylist\string\n
10242     \string"}
10243   \write\glswrite{item_12 \string"\string%\string\n\string"}
10244   \write\glswrite{item_x2
10245     \string"\string\relax \string\glsresetentrylist\string\n
10246     \string"}
10247   \write\glswrite{delim_0 \string"\string\{\string
10248     \glossaryentrynumbers\string\{\string\relax \string"}
10249   \write\glswrite{delim_1 \string"\string\{\string

```

```

10250      \glossaryentrynumbers\string\{\string\relax \string}
10251      \write\glswrite{delim_2 \string"\string\{\string
10252      \glossaryentrynumbers\string\{\string\relax \string}
10253      \write\glswrite{delim_t \string"\string\}\string\}\string}
10254      \write\glswrite{delim_n \string"\string\delimN \string}
10255      \write\glswrite{delim_r \string"\string\delimR \string}
10256      \write\glswrite{headings_flag 1}
10257      \write\glswrite{heading_prefix
10258        \string"\string\glsgroupheading\string\{\string}
10259      \write\glswrite{heading_suffix
10260        \string"\string\}\string\relax
10261        \string\glsgroupresetentrylist \string}
10262      \write\glswrite{symhead_positive \string"glssymbols\string}
10263      \write\glswrite{numhead_positive \string"glslnumbers\string}
10264      \write\glswrite{page_compositor \string"glscpositor\string}
10265      \@gls@escbsdq\gls@suffixF
10266      \@gls@escbsdq\gls@suffixFF
10267      \ifx\gls@suffixF\@empty
10268      \else
10269        \write\glswrite{suffix_2p \string"\gls@suffixF\string}
10270      \fi
10271      \ifx\gls@suffixFF\@empty
10272      \else
10273        \write\glswrite{suffix_3p \string"\gls@suffixFF\string}
10274      \fi
10275      \noist
10276    }
10277  \fi

```

\noist

```
10278 \renewcommand*{\noist}{\let\writeist\relax}
```

4.2 glossaries-compatible-307

```

10279 \NeedsTeXFormat{LaTeX2e}
10280 \ProvidesPackage{glossaries-compatible-307}[2021/09/20 v4.47 (NLCT)]

```

Compatibility macros for predefined glossary styles:

`\atglossarystyle` Defines a compatibility glossary style.

```

10281 \newcommand{\compatglossarystyle}[2]{%
10282   \ifcsundef{@glscompstyle@#1}%
10283   {%
10284     \csdef{@glscompstyle@#1}{#2}%
10285   }%
10286   {%
10287     \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
10288   }%
10289 }

```

Backward compatible inline style.

```

10290 \compatglossarystyle{inline}{%
10291   \renewcommand{\glossaryentryfield}[5]{%
10292     \glsinlinedopostchild
10293     \gls@inlinesep
10294     \def\glo@desc{##3}%
10295     \def\@no@post@desc{\nopostdesc}%
10296     \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
10297     \ifx\glo@desc\@no@post@desc
10298       \glsinlineemptydescformat{##4}{##5}%
10299     \else
10300       \ifstrepty{##3}%
10301         {\glsinlineemptydescformat{##4}{##5}}%
10302         {\glsinlinedescformat{##3}{##4}{##5}}%
10303     \fi
10304     \ifglshaschildren{##1}%
10305     {%
10306       \glsresetsubentrycounter
10307       \glsinlineparentchildseparator
10308       \def\gls@inlinesubsep{}%
10309       \def\gls@inlinepostchild{\glsinlinepostchild}%
10310     }%
10311   }%
10312   \def\gls@inlinesep{\glsinlineseparator}%
10313 }%

```

Sub-entries display description:

```

10314 \renewcommand{\glossarysubentryfield}[6]{%
10315   \gls@inlinesubsep%
10316   \glsinlinesubnameformat{##2}{##3}%
10317   \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
10318   \def\gls@inlinesubsep{\glsinlinesubseparator}%
10319 }%
10320 }

```

Backward compatible list style.

```

10321 \compatglossarystyle{list}{%
10322   \renewcommand*{\glossaryentryfield}[5]{%
10323     \item[\glsentryitem{##1}\glstarget{##1}{##2}]
10324     ##3\glspostdescription\space ##5}%

```

Sub-entries continue on the same line:

```

10325   \renewcommand*{\glossarysubentryfield}[6]{%
10326     \glssubentryitem{##2}%
10327     \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
10328 }

```

Backward compatible listgroup style.

```

10329 \compatglossarystyle{listgroup}{%
10330   \csuse{@glscompstyle@list}%
10331 }%

```

Backward compatible listhypergroup style.

```
10332 \compatglossarystyle{listhypergroup}{%
10333   \csuse{@glscompstyle@list}%
10334 }%
```

Backward compatible altlist style.

```
10335 \compatglossarystyle{altlist}{%
10336   \renewcommand*{\glossaryentryfield}[5]{%
10337     \item[\glstentryitem{##1}\glstarget{##1}{##2}]%
10338     \mbox{}\par\nobreak\@afterheading
10339     ##3\glspostdescription\space ##5}%
10340   \renewcommand{\glossarysubentryfield}[6]{%
10341     \par
10342     \glssubentryitem{##2}%
10343     \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
10344 }%
```

Backward compatible altlistgroup style.

```
10345 \compatglossarystyle{altlistgroup}{%
10346   \csuse{@glscompstyle@altlist}%
10347 }%
```

Backward compatible altlisthypergroup style.

```
10348 \compatglossarystyle{altlisthypergroup}{%
10349   \csuse{@glscompstyle@altlist}%
10350 }%
```

Backward compatible listdotted style.

```
10351 \compatglossarystyle{listdotted}{%
10352   \renewcommand*{\glossaryentryfield}[5]{%
10353     \item[\makebox[\glslistdottedwidth][l]{%
10354       \glstentryitem{##1}\glstarget{##1}{##2}%
10355       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
10356   \renewcommand*{\glossarysubentryfield}[6]{%
10357     \item[\makebox[\glslistdottedwidth][l]{%
10358       \glssubentryitem{##2}%
10359       \glstarget{##2}{##3}%
10360       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
10361 }%
```

Backward compatible sublistdotted style.

```
10362 \compatglossarystyle{sublistdotted}{%
10363   \csuse{@glscompstyle@listdotted}%
10364   \renewcommand*{\glossaryentryfield}[5]{%
10365     \item[\glstentryitem{##1}\glstarget{##1}{##2}]}%
10366 }%
```

Backward compatible long style.

```
10367 \compatglossarystyle{long}{%
10368   \renewcommand*{\glossaryentryfield}[5]{%
10369     \glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
10370   \renewcommand*{\glossarysubentryfield}[6]{%
10371     \glssubentryitem{##2}\glstarget{##2}{##3} & ##4\glspostdescription\space ##5\\}%
10372 }%
```

```

10371      &
10372      \glssubentryitem{##2}%
10373      \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
10374 }%

```

Backward compatible longborder style.

```

10375 \compatglossarystyle{longborder}{%
10376   \csuse{@glscmpstyle@long}%
10377 }%

```

Backward compatible longheader style.

```

10378 \compatglossarystyle{longheader}{%
10379   \csuse{@glscmpstyle@long}%
10380 }%

```

Backward compatible longheaderborder style.

```

10381 \compatglossarystyle{longheaderborder}{%
10382   \csuse{@glscmpstyle@long}%
10383 }%

```

Backward compatible long3col style.

```

10384 \compatglossarystyle{long3col}{%
10385   \renewcommand*{\glossaryentryfield}[5]{%
10386     \glstentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
10387   \renewcommand*{\glossarysubentryfield}[6]{%
10388     &
10389     \glssubentryitem{##2}%
10390     \glstarget{##2}{\strut}##4 & ##6\\}%
10391 }%

```

Backward compatible long3colborder style.

```

10392 \compatglossarystyle{long3colborder}{%
10393   \csuse{@glscmpstyle@long3col}%
10394 }%

```

Backward compatible long3colheader style.

```

10395 \compatglossarystyle{long3colheader}{%
10396   \csuse{@glscmpstyle@long3col}%
10397 }%

```

Backward compatible long3colheaderborder style.

```

10398 \compatglossarystyle{long3colheaderborder}{%
10399   \csuse{@glscmpstyle@long3col}%
10400 }%

```

Backward compatible long4col style.

```

10401 \compatglossarystyle{long4col}{%
10402   \renewcommand*{\glossaryentryfield}[5]{%
10403     \glstentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
10404   \renewcommand*{\glossarysubentryfield}[6]{%
10405     &
10406     \glssubentryitem{##2}%

```

10407 \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
10408 }%

Backward compatible long4colheader style.

10409 \compatglossarystyle{long4colheader}{%
10410 \csuse{@glscompstyle@long4col}%
10411 }%

Backward compatible long4colborder style.

10412 \compatglossarystyle{long4colborder}{%
10413 \csuse{@glscompstyle@long4col}%
10414 }%

Backward compatible long4colheaderborder style.

10415 \compatglossarystyle{long4colheaderborder}{%
10416 \csuse{@glscompstyle@long4col}%
10417 }%

Backward compatible altlong4col style.

10418 \compatglossarystyle{altlong4col}{%
10419 \csuse{@glscompstyle@long4col}%
10420 }%

Backward compatible altlong4colheader style.

10421 \compatglossarystyle{altlong4colheader}{%
10422 \csuse{@glscompstyle@long4col}%
10423 }%

Backward compatible altlong4colborder style.

10424 \compatglossarystyle{altlong4colborder}{%
10425 \csuse{@glscompstyle@long4col}%
10426 }%

Backward compatible altlong4colheaderborder style.

10427 \compatglossarystyle{altlong4colheaderborder}{%
10428 \csuse{@glscompstyle@long4col}%
10429 }%

Backward compatible long style.

10430 \compatglossarystyle{longragged}{%
10431 \renewcommand*{\glossaryentryfield}[5]{%
10432 \glssentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
10433 \tabularnewline}%
10434 \renewcommand*{\glossarysubentryfield}[6]{%
10435 &
10436 \glssubentryitem{##2}%
10437 \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
10438 \tabularnewline}%
10439 }%

Backward compatible longraggedborder style.

10440 \compatglossarystyle{longraggedborder}{%
10441 \csuse{@glscompstyle@longragged}%
10442 }%

Backward compatible longraggedheader style.

```
10443 \compatglossarystyle{longraggedheader}{%
10444   \csuse{@glscompstyle@longragged}%
10445 }%
```

Backward compatible longraggedheaderborder style.

```
10446 \compatglossarystyle{longraggedheaderborder}{%
10447   \csuse{@glscompstyle@longragged}%
10448 }%
```

Backward compatible longragged3col style.

```
10449 \compatglossarystyle{longragged3col}{%
10450   \renewcommand*{\glossaryentryfield}[5]{%
10451     \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
10452   \renewcommand*{\glossarysubentryfield}[6]{%
10453     &
10454     \glssubentryitem{##2}%
10455     \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
10456 }%
```

Backward compatible longragged3colborder style.

```
10457 \compatglossarystyle{longragged3colborder}{%
10458   \csuse{@glscompstyle@longragged3col}%
10459 }%
```

Backward compatible longragged3colheader style.

```
10460 \compatglossarystyle{longragged3colheader}{%
10461   \csuse{@glscompstyle@longragged3col}%
10462 }%
```

Backward compatible longragged3colheaderborder style.

```
10463 \compatglossarystyle{longragged3colheaderborder}{%
10464   \csuse{@glscompstyle@longragged3col}%
10465 }%
```

Backward compatible altlongragged4col style.

```
10466 \compatglossarystyle{altlongragged4col}{%
10467   \renewcommand*{\glossaryentryfield}[5]{%
10468     \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
10469   \renewcommand*{\glossarysubentryfield}[6]{%
10470     &
10471     \glssubentryitem{##2}%
10472     \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
10473 }%
```

Backward compatible altlongragged4colheader style.

```
10474 \compatglossarystyle{altlongragged4colheader}{%
10475   \csuse{@glscompstyle@altlong4col}%
10476 }%
```

Backward compatible altlongragged4colborder style.

```
10477 \compatglossarystyle{altlongragged4colborder}{%
```



```

10478 \csuse{@glscompstyle@altlong4col}%
10479 }%

```

Backward compatible altlongragged4colheaderborder style.

```

10480 \compatglossarystyle{altlongragged4colheaderborder}{%
10481 \csuse{@glscompstyle@altlong4col}%
10482 }%

```

Backward compatible index style.

```

10483 \compatglossarystyle{index}{%
10484 \renewcommand*{\glossaryentryfield}[5]{%
10485 \item\glsentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10486 \ifx\relax##4\relax
10487 \else
10488 \space{##4}%
10489 \fi
10490 \space ##3\glspostdescription \space ##5}%
10491 \renewcommand*{\glossarysubentryfield}[6]{%
10492 \ifcase##1\relax
10493 % level 0
10494 \item
10495 \or
10496 % level 1
10497 \subitem
10498 \glssubentryitem{##2}%
10499 \else
10500 % all other levels
10501 \subsubitem
10502 \fi
10503 \textbf{\glstarget{##2}{##3}}%
10504 \ifx\relax##5\relax
10505 \else
10506 \space{##5}%
10507 \fi
10508 \space##4\glspostdescription\space ##6}%
10509 }%

```

Backward compatible indexgroup style.

```

10510 \compatglossarystyle{indexgroup}{%
10511 \csuse{@glscompstyle@index}%
10512 }%

```

Backward compatible indexhypergroup style.

```

10513 \compatglossarystyle{indexhypergroup}{%
10514 \csuse{@glscompstyle@index}%
10515 }%

```

Backward compatible tree style.

```

10516 \compatglossarystyle{tree}{%
10517 \renewcommand{\glossaryentryfield}[5]{%
10518 \hangindent0pt\relax

```

```

10519 \parindent0pt\relax
10520 \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10521 \ifx\relax##4\relax
10522 \else
10523 \space(##4)%
10524 \fi
10525 \space ##3\glspostdescription \space ##5\par}%
10526 \renewcommand{\glossarysubentryfield}[6]{%
10527 \hangindent##1\glstreeindent\relax
10528 \parindent##1\glstreeindent\relax
10529 \ifnum##1=1\relax
10530 \glssubentryitem{##2}%
10531 \fi
10532 \textbf{\glstarget{##2}{##3}}%
10533 \ifx\relax##5\relax
10534 \else
10535 \space(##5)%
10536 \fi
10537 \space##4\glspostdescription\space ##6\par}%
10538 }%

```

Backward compatible treegroup style.

```

10539 \compatglossarystyle{treegroup}{%
10540 \csuse{@glscmpstyle@tree}%
10541 }%

```

Backward compatible treehypergroup style.

```

10542 \compatglossarystyle{treehypergroup}{%
10543 \csuse{@glscmpstyle@tree}%
10544 }%

```

Backward compatible treenoname style.

```

10545 \compatglossarystyle{treenoname}{%
10546 \renewcommand{\glossaryentryfield}[5]{%
10547 \hangindent0pt\relax
10548 \parindent0pt\relax
10549 \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10550 \ifx\relax##4\relax
10551 \else
10552 \space(##4)%
10553 \fi
10554 \space ##3\glspostdescription \space ##5\par}%
10555 \renewcommand{\glossarysubentryfield}[6]{%
10556 \hangindent##1\glstreeindent\relax
10557 \parindent##1\glstreeindent\relax
10558 \ifnum##1=1\relax
10559 \glssubentryitem{##2}%
10560 \fi
10561 \glstarget{##2}{\strut}%
10562 ##4\glspostdescription\space ##6\par}%
10563 }%

```

Backward compatible treenonamegroup style.

```
10564 \compatglossarystyle{treenonamegroup}{%
10565   \csuse{@glscompstyle@treenoname}%
10566 }%
```

Backward compatible treenonamehypergroup style.

```
10567 \compatglossarystyle{treenonamehypergroup}{%
10568   \csuse{@glscompstyle@treenoname}%
10569 }%
```

Backward compatible alttree style.

```
10570 \compatglossarystyle{alttree}{%
10571   \renewcommand{\glossaryentryfield}[5]{%
10572     \ifnum\@gls@prevlevel=0\relax
10573     \else
10574       \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
10575       \hangindent\glstreeindent
10576       \parindent\glstreeindent
10577     \fi
10578     \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
10579       \glssentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
10580     \ifx\relax##4\relax
10581     \else
10582       (##4)\space
10583     \fi
10584     ##3\glspostdescription \space ##5\par
10585     \def\@gls@prevlevel{0}%
10586   }%
10587   \renewcommand{\glossarysubentryfield}[6]{%
10588     \ifnum##1=1\relax
10589       \glssubentryitem{##2}%
10590     \fi
10591     \ifnum\@gls@prevlevel=##1\relax
10592     \else
10593       \@ifundefined{@glswidestname\romannumeral##1}{%
10594         \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
10595         \settowidth{\gls@tmplen}{\textbf{%
10596           \csname @glswidestname\romannumeral##1\endcsname\space}}}%
10597       \ifnum\@gls@prevlevel<##1\relax
10598         \setlength\glstreeindent\gls@tmplen
10599         \addtolength\glstreeindent\parindent
10600       \parindent\glstreeindent
10601     \else
10602       \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
10603         \settowidth{\glstreeindent}{\textbf{%
10604           \@glswidestname\space}}{%
10605         \settowidth{\glstreeindent}{\textbf{%
10606           \csname @glswidestname\romannumeral\@gls@prevlevel
10607             \endcsname\space}}}%
10608         \addtolength\parindent{-\glstreeindent}%

```

```

10609         \setlength\glstreeindent\parindent
10610     \fi
10611 \fi
10612 \hangindent\glstreeindent
10613 \makebox[0pt][r]{\makebox[\glstemplen][l]{%
10614     \textbf{\glstarget{##2}{##3}}}%
10615 \ifx##5\relax\relax
10616 \else
10617     (##5)\space
10618 \fi
10619 ##4\glspostdescription\space ##6\par
10620 \def\@gls@prevlevel{##1}%
10621 }%
10622 }%

```

Backward compatible alttreegroup style.

```

10623 \compatglossarystyle{alttreegroup}{%
10624 \csuse{@glscompstyle@alttree}%
10625 }%

```

Backward compatible alttreehypergroup style.

```

10626 \compatglossarystyle{alttreehypergroup}{%
10627 \csuse{@glscompstyle@alttree}%
10628 }%

```

Backward compatible mcolindex style.

```

10629 \compatglossarystyle{mcolindex}{%
10630 \csuse{@glscompstyle@index}%
10631 }%

```

Backward compatible mcolindexgroup style.

```

10632 \compatglossarystyle{mcolindexgroup}{%
10633 \csuse{@glscompstyle@index}%
10634 }%

```

Backward compatible mcolindexhypergroup style.

```

10635 \compatglossarystyle{mcolindexhypergroup}{%
10636 \csuse{@glscompstyle@index}%
10637 }%

```

Backward compatible mcoltree style.

```

10638 \compatglossarystyle{mcoltree}{%
10639 \csuse{@glscompstyle@tree}%
10640 }%

```

Backward compatible mcoltreegroup style.

```

10641 \compatglossarystyle{mcolindextreegroup}{%
10642 \csuse{@glscompstyle@tree}%
10643 }%

```

Backward compatible mcoltreehypergroup style.

```

10644 \compatglossarystyle{mcolindextreehypergroup}{%

```

```

10645 \csuse{@glscompstyle@tree}%
10646 }%

    Backward compatible mcoltreenoname style.
10647 \compatglossarystyle{mcoltreenoname}{%
10648 \csuse{@glscompstyle@tree}%
10649 }%

    Backward compatible mcoltreenonamegroup style.
10650 \compatglossarystyle{mcoltreenonamegroup}{%
10651 \csuse{@glscompstyle@tree}%
10652 }%

    Backward compatible mcoltreenonamehypergroup style.
10653 \compatglossarystyle{mcoltreenonamehypergroup}{%
10654 \csuse{@glscompstyle@tree}%
10655 }%

    Backward compatible mcolalmtree style.
10656 \compatglossarystyle{mcolalmtree}{%
10657 \csuse{@glscompstyle@almtree}%
10658 }%

    Backward compatible mcolalmtreegroup style.
10659 \compatglossarystyle{mcolalmtreegroup}{%
10660 \csuse{@glscompstyle@almtree}%
10661 }%

    Backward compatible mcolalmtreehypergroup style.
10662 \compatglossarystyle{mcolalmtreehypergroup}{%
10663 \csuse{@glscompstyle@almtree}%
10664 }%

    Backward compatible superragged style.
10665 \compatglossarystyle{superragged}{%
10666 \renewcommand*{\glossaryentryfield}[5]{%
10667 \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
10668 \tabularnewline}%
10669 \renewcommand*{\glossarysubentryfield}[6]{%
10670 &
10671 \glssubentryitem{##2}%
10672 \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
10673 \tabularnewline}%
10674 }%

    Backward compatible superraggedborder style.
10675 \compatglossarystyle{superraggedborder}{%
10676 \csuse{@glscompstyle@superragged}%
10677 }%

    Backward compatible superraggedheader style.
10678 \compatglossarystyle{superraggedheader}{%
10679 \csuse{@glscompstyle@superragged}%
10680 }%

```

Backward compatible superraggedheaderborder style.

```
10681 \compatglossarystyle{superraggedheaderborder}{%
10682   \csuse{@glscompstyle@superragged}%
10683 }%
```

Backward compatible superragged3col style.

```
10684 \compatglossarystyle{superragged3col}{%
10685   \renewcommand*{\glossaryentryfield}[5]{%
10686     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
10687   \renewcommand*{\glossarysubentryfield}[6]{%
10688     &
10689     \glssubentryitem{##2}%
10690     \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
10691 }%
```

Backward compatible superragged3colborder style.

```
10692 \compatglossarystyle{superragged3colborder}{%
10693   \csuse{@glscompstyle@superragged3col}%
10694 }%
```

Backward compatible superragged3colheader style.

```
10695 \compatglossarystyle{superragged3colheader}{%
10696   \csuse{@glscompstyle@superragged3col}%
10697 }%
```

Backward compatible superragged3colheaderborder style.

```
10698 \compatglossarystyle{superragged3colheaderborder}{%
10699   \csuse{@glscompstyle@superragged3col}%
10700 }%
```

Backward compatible altsuperragged4col style.

```
10701 \compatglossarystyle{altsuperragged4col}{%
10702   \renewcommand*{\glossaryentryfield}[5]{%
10703     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
10704   \renewcommand*{\glossarysubentryfield}[6]{%
10705     &
10706     \glssubentryitem{##2}%
10707     \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
10708 }%
```

Backward compatible altsuperragged4colheader style.

```
10709 \compatglossarystyle{altsuperragged4colheader}{%
10710   \csuse{@glscompstyle@altsuperragged4col}%
10711 }%
```

Backward compatible altsuperragged4colborder style.

```
10712 \compatglossarystyle{altsuperragged4colborder}{%
10713   \csuse{@glscompstyle@altsuperragged4col}%
10714 }%
```

Backward compatible altsuperragged4colheaderborder style.

```
10715 \compatglossarystyle{altsuperragged4colheaderborder}{%
```

```
10716 \csuse{@glscompstyle@altsuperragged4col}%
10717 }%
```

Backward compatible super style.

```
10718 \compatglossarystyle{super}{%
10719   \renewcommand*{\glossaryentryfield}[5]{%
10720     \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
10721   \renewcommand*{\glossarysubentryfield}[6]{%
10722     &
10723     \glssubentryitem{##2}%
10724     \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
10725 }%
```

Backward compatible superborder style.

```
10726 \compatglossarystyle{superborder}{%
10727   \csuse{@glscompstyle@super}%
10728 }%
```

Backward compatible superheader style.

```
10729 \compatglossarystyle{superheader}{%
10730   \csuse{@glscompstyle@super}%
10731 }%
```

Backward compatible superheaderborder style.

```
10732 \compatglossarystyle{superheaderborder}{%
10733   \csuse{@glscompstyle@super}%
10734 }%
```

Backward compatible super3col style.

```
10735 \compatglossarystyle{super3col}{%
10736   \renewcommand*{\glossaryentryfield}[5]{%
10737     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
10738   \renewcommand*{\glossarysubentryfield}[6]{%
10739     &
10740     \glssubentryitem{##2}%
10741     \glstarget{##2}{\strut}##4 & ##6\\}%
10742 }%
```

Backward compatible super3colborder style.

```
10743 \compatglossarystyle{super3colborder}{%
10744   \csuse{@glscompstyle@super3col}%
10745 }%
```

Backward compatible super3colheader style.

```
10746 \compatglossarystyle{super3colheader}{%
10747   \csuse{@glscompstyle@super3col}%
10748 }%
```

Backward compatible super3colheaderborder style.

```
10749 \compatglossarystyle{super3colheaderborder}{%
10750   \csuse{@glscompstyle@super3col}%
10751 }%
```

Backward compatible super4col style.

```
10752 \compatglossarystyle{super4col}{%
10753   \renewcommand*{\glossaryentryfield}[5]{%
10754     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
10755   \renewcommand*{\glossarysubentryfield}[6]{%
10756     &
10757     \glssubentryitem{##2}%
10758     \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
10759 }%
```

Backward compatible super4colheader style.

```
10760 \compatglossarystyle{super4colheader}{%
10761   \csuse{@glscompstyle@super4col}%
10762 }%
```

Backward compatible super4colborder style.

```
10763 \compatglossarystyle{super4colborder}{%
10764   \csuse{@glscompstyle@super4col}%
10765 }%
```

Backward compatible super4colheaderborder style.

```
10766 \compatglossarystyle{super4colheaderborder}{%
10767   \csuse{@glscompstyle@super4col}%
10768 }%
```

Backward compatible altsuper4col style.

```
10769 \compatglossarystyle{altsuper4col}{%
10770   \csuse{@glscompstyle@super4col}%
10771 }%
```

Backward compatible altsuper4colheader style.

```
10772 \compatglossarystyle{altsuper4colheader}{%
10773   \csuse{@glscompstyle@super4col}%
10774 }%
```

Backward compatible altsuper4colborder style.

```
10775 \compatglossarystyle{altsuper4colborder}{%
10776   \csuse{@glscompstyle@super4col}%
10777 }%
```

Backward compatible altsuper4colheaderborder style.

```
10778 \compatglossarystyle{altsuper4colheaderborder}{%
10779   \csuse{@glscompstyle@super4col}%
10780 }%
```


5 Accessibility Support (glossaries-accsupp Code)

The package is experimental. It is intended to provide a means of using the PDF accessibility support in glossary entries. See the documentation for further details about accessibility support.

```
10781 \NeedsTeXFormat{LaTeX2e}
```

Package version number now in line with main glossaries package number.

```
10782 \ProvidesPackage{glossaries-accsupp}[2021/09/20 v4.47 (NLCT)]
```

```
10783 Experimental glossaries accessibility]
```

Pass all options to glossaries:

```
10784 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
10785 \ProcessOptions
```

This package should be loaded before glossaries-extra, so complain if that has already been loaded.

```
10786 \ifpackageloaded{glossaries-extra}
```

```
10787 {%
```

If the accsupp option was used, \@glsxtr@doaccsupp will have been set, otherwise it will be empty.

```
10788 \ifx\@glsxtr@doaccsupp\empty
```

```
10789 \GlossariesWarning{The 'glossaries-accsupp'
```

```
10790 package has been loaded\MessageBreak
```

```
10791 after the 'glossaries-extra' package. This\MessageBreak
```

```
10792 can cause a failure to integrate both packages. \MessageBreak
```

```
10793 Either use the 'accsupp' option when you load\MessageBreak
```

```
10794 'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
```

```
10795 before loading 'glossaries-extra'}%
```

```
10796 \fi
```

```
10797 }
```

```
10798 {}
```

tibleglossentry Override style compatibility macros:

```
10799 \def\compatibleglossentry#1#2{%
```

```
10800 \toks@{#2}%
```

```
10801 \protected@edef\do@glossentry{%
```

```
10802 \noexpand\accsuppglossaryentryfield{#1}%
```

```
10803 {\noexpand\glsnamefont
```

```
10804 {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
```

```

10805     {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
10806     {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
10807     {\the\toks@}%
10808   }%
10809   \@do@glossentry
10810 }

```

lesubglossentry

```

10811 \def\compatiblesubglossentry#1#2#3{%
10812   \toks@{#3}%
10813   \protected@edef\@do@subglossentry{%
10814     \noexpand\accsuppglossarysubentryfield{\number#1}%
10815     {#2}%
10816     {\noexpand\glsnamefont
10817       {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
10818     {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
10819     {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
10820     {\the\toks@}%
10821   }%
10822   \@do@subglossentry
10823 }

```

Required packages:

```

10824 \RequirePackage{glossaries}

```

@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.

```

10825 \providecommand{\gls@accsupp@engine}{accsupp}

```

s@accessibility `\gls@accessibility[\<options>]{\<PDF element>}{\<value>}{\<content>}`

```

10826 \providecommand{\gls@accessibility}[4]{#4}
10827 \ifdefstring\gls@accsupp@engine{accsupp}
10828 {
10829   \RequirePackage{accsupp}
10830   \renewcommand{\gls@accessibility}[4]{%
10831     \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp}%
10832   }
10833 }
10834 {}

```

lsaccessibility `\glsaccessibility[\<options>]{\<PDF element>}{\<value>}{\<content>}`

User-level command that includes debug info if required.

```

10835 \newcommand{\glsaccessibility}[4] [] {%
10836   \@glsshowaccsupp{#1}{#2}{#3}%
10837   \gls@accessibility{#1}{#2}{#3}{#4}%
10838 }

```

5.1 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:

```

10839 \define@key{glossentry}{access}{%
10840   \def\@glo@access{#1}%
10841 }

```

textaccess The replacement text corresponding to the text key:

```

10842 \define@key{glossentry}{textaccess}{%
10843   \def\@glo@textaccess{#1}%
10844 }

```

firstaccess The replacement text corresponding to the first key:

```

10845 \define@key{glossentry}{firstaccess}{%
10846   \def\@glo@firstaccess{#1}%
10847 }

```

pluralaccess The replacement text corresponding to the plural key:

```

10848 \define@key{glossentry}{pluralaccess}{%
10849   \def\@glo@pluralaccess{#1}%
10850 }

```

firstpluralaccess The replacement text corresponding to the firstplural key:

```

10851 \define@key{glossentry}{firstpluralaccess}{%
10852   \def\@glo@firstpluralaccess{#1}%
10853 }

```

symbolaccess The replacement text corresponding to the symbol key:

```

10854 \define@key{glossentry}{symbolaccess}{%
10855   \def\@glo@symbolaccess{#1}%
10856 }

```

symbolpluralaccess The replacement text corresponding to the symbolplural key:

```

10857 \define@key{glossentry}{symbolpluralaccess}{%
10858   \def\@glo@symbolpluralaccess{#1}%
10859 }

```

descriptionaccess The replacement text corresponding to the description key:

```
10860 \define@key{glossentry}{descriptionaccess}{%
10861   \def\@glo@descaccess{#1}%
10862 }
```

ionpluralaccess The replacement text corresponding to the descriptionplural key:

```
10863 \define@key{glossentry}{descriptionpluralaccess}{%
10864   \def\@glo@descpluralaccess{#1}%
10865 }
```

shortaccess The replacement text corresponding to the short key:

```
10866 \define@key{glossentry}{shortaccess}{%
10867   \def\@glo@shortaccess{#1}%
10868 }
```

ortpluralaccess The replacement text corresponding to the shortplural key:

```
10869 \define@key{glossentry}{shortpluralaccess}{%
10870   \def\@glo@shortpluralaccess{#1}%
10871 }
```

longaccess The replacement text corresponding to the long key:

```
10872 \define@key{glossentry}{longaccess}{%
10873   \def\@glo@longaccess{#1}%
10874 }
```

ongpluralaccess The replacement text corresponding to the longplural key:

```
10875 \define@key{glossentry}{longpluralaccess}{%
10876   \def\@glo@longpluralaccess{#1}%
10877 }
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
10878 \define@key{glossentry}{user1access}{%
10879   \def\@glo@useriaccess{#1}%
10880 }
```

user2access The replacement text corresponding to the user2 key:

```
10881 \define@key{glossentry}{user2access}{%
10882   \def\@glo@useriiaccess{#1}%
10883 }
```

user3access The replacement text corresponding to the user3 key:

```
10884 \define@key{glossentry}{user3access}{%
10885   \def\@glo@useriiiaccess{#1}%
10886 }
```

user4access The replacement text corresponding to the user4 key:

```
10887 \define@key{glossentry}{user4access}{%
10888   \def\@glo@userivaccess{#1}%
10889 }
```

user5access The replacement text corresponding to the user5 key:

```
10890 \define@key{glossentry}{user5access}{%
10891   \def\@glo@uservaccess{#1}%
10892 }
```

user6access The replacement text corresponding to the user6 key:

```
10893 \define@key{glossentry}{user6access}{%
10894   \def\@glo@userviaccess{#1}%
10895 }
```

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:

```
10896 \appto\@gls@keymap{,%
10897   {access}{access},%
10898   {textaccess}{textaccess},%
10899   {firstaccess}{firstaccess},%
10900   {pluralaccess}{pluralaccess},%
10901   {firstpluralaccess}{firstpluralaccess},%
10902   {symbolaccess}{symbolaccess},%
10903   {symbolpluralaccess}{symbolpluralaccess},%
10904   {descaccess}{descaccess},%
10905   {descpluralaccess}{descpluralaccess},%
10906   {shortaccess}{shortaccess},%
10907   {shortpluralaccess}{shortpluralaccess},%
10908   {longaccess}{longaccess},%
10909   {longpluralaccess}{longpluralaccess},%
10910   {user1access}{useriaccess},%
10911   {user2access}{useriiaccess},%
10912   {user3access}{useriiiaccess},%
10913   {user4access}{userivaccess},%
10914   {user5access}{uservaccess},%
10915   {user6access}{userviaccess}%
10916 }
```

\@gls@noaccess Indicates that no replacement text has been provided.

```
10917 \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.

```
10918 \let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
10919 \renewcommand*{\@newglossaryentryprehook}{%
```

```

10920 \@gls@oldnewglossaryentryprehook
10921 \def\@glo@access{\relax}%

```

Initialise the other keys:

```

10922 \def\@glo@textaccess{\@glo@access}%
10923 \def\@glo@firstaccess{\@glo@access}%
10924 \def\@glo@pluralaccess{\@glo@textaccess}%
10925 \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
10926 \def\@glo@symbolaccess{\relax}%
10927 \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
10928 \def\@glo@descaccess{\relax}%
10929 \def\@glo@descpluralaccess{\@glo@descaccess}%
10930 \def\@glo@shortaccess{\relax}%
10931 \def\@glo@shortpluralaccess{\@glo@shortaccess}%
10932 \def\@glo@longaccess{\relax}%
10933 \def\@glo@longpluralaccess{\@glo@longaccess}%
10934 \def\@glo@useriaccess{\relax}%
10935 \def\@glo@useriiaaccess{\relax}%
10936 \def\@glo@useriiiaaccess{\relax}%
10937 \def\@glo@userivaccess{\relax}%
10938 \def\@glo@uservaccess{\relax}%
10939 \def\@glo@userviiaaccess{\relax}%
10940 }

```

Add to the end hook:

```

10941 \let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
10942 \renewcommand*{\@newglossaryentryposthook}{%
10943 \@gls@oldnewglossaryentryposthook

```

Store the access information:

```

10944 \expandafter
10945 \protected@xdef\csname glo@\@glo@label @access\endcsname{%
10946 \@glo@access}%
10947 \expandafter
10948 \protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
10949 \@glo@textaccess}%
10950 \expandafter
10951 \protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
10952 \@glo@firstaccess}%
10953 \expandafter
10954 \protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
10955 \@glo@pluralaccess}%
10956 \expandafter
10957 \protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
10958 \@glo@firstpluralaccess}%
10959 \expandafter
10960 \protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
10961 \@glo@symbolaccess}%
10962 \expandafter
10963 \protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
10964 \@glo@symbolpluralaccess}%

```

```

10965 \expandafter
10966   \protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
10967     \@glo@descaccess}%
10968 \expandafter
10969   \protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
10970     \@glo@descpluralaccess}%
10971 \expandafter
10972   \protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
10973     \@glo@shortaccess}%
10974 \expandafter
10975   \protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
10976     \@glo@shortpluralaccess}%
10977 \expandafter
10978   \protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
10979     \@glo@longaccess}%
10980 \expandafter
10981   \protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
10982     \@glo@longpluralaccess}%
10983 \expandafter
10984   \protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
10985     \@glo@useriaccess}%
10986 \expandafter
10987   \protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
10988     \@glo@useriiaccess}%
10989 \expandafter
10990   \protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
10991     \@glo@useriiiaccess}%
10992 \expandafter
10993   \protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
10994     \@glo@userivaccess}%
10995 \expandafter
10996   \protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
10997     \@glo@uservaccess}%
10998 \expandafter
10999   \protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
11000     \@glo@userviaccess}%
11001 }

```

5.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

11002 \newcommand*{\glsentryaccess}[1]{%
11003   \@gls@entry@field{#1}{access}%
11004 }

```

`entrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

11005 \newcommand*{\glsentrytextaccess}[1]{%
11006   \@gls@entry@field{#1}{textaccess}%

```

```

11007 }

entryfirstaccess  Get the value of the firstaccess key for the entry with the given label:
11008 \newcommand*{\glentryfirstaccess}[1]{%
11009   \@gls@entry@field{#1}{firstaccess}%
11010 }

entrypluralaccess  Get the value of the pluralaccess key for the entry with the given label:
11011 \newcommand*{\glentrypluralaccess}[1]{%
11012   \@gls@entry@field{#1}{pluralaccess}%
11013 }

firstpluralaccess  Get the value of the firstpluralaccess key for the entry with the given label:
11014 \newcommand*{\glentryfirstpluralaccess}[1]{%
11015   \@gls@entry@field{#1}{firstpluralaccess}%
11016 }

entrysymbolaccess  Get the value of the symbolaccess key for the entry with the given label:
11017 \newcommand*{\glentrysymbolaccess}[1]{%
11018   \@gls@entry@field{#1}{symbolaccess}%
11019 }

symbolpluralaccess  Get the value of the symbolpluralaccess key for the entry with the given label:
11020 \newcommand*{\glentrysymbolpluralaccess}[1]{%
11021   \@gls@entry@field{#1}{symbolpluralaccess}%
11022 }

entrydescaccess  Get the value of the descriptionaccess key for the entry with the given label:
11023 \newcommand*{\glentrydescaccess}[1]{%
11024   \@gls@entry@field{#1}{descaccess}%
11025 }

descpluralaccess  Get the value of the descriptionpluralaccess key for the entry with the given label:
11026 \newcommand*{\glentrydescpluralaccess}[1]{%
11027   \@gls@entry@field{#1}{descpluralaccess}%
11028 }

entryshortaccess  Get the value of the shortaccess key for the entry with the given label:
11029 \newcommand*{\glentryshortaccess}[1]{%
11030   \@gls@entry@field{#1}{shortaccess}%
11031 }

shortpluralaccess  Get the value of the shortpluralaccess key for the entry with the given label:
11032 \newcommand*{\glentryshortpluralaccess}[1]{%
11033   \@gls@entry@field{#1}{shortpluralaccess}%
11034 }

```


entrylongaccess Get the value of the longaccess key for the entry with the given label:

```
11035 \newcommand*{\glentrylongaccess}[1]{%
11036   \@gls@entry@field{#1}{longaccess}%
11037 }
```

ongpluralaccess Get the value of the longpluralaccess key for the entry with the given label:

```
11038 \newcommand*{\glentrylongpluralaccess}[1]{%
11039   \@gls@entry@field{#1}{longpluralaccess}%
11040 }
```

entryuseriaccess Get the value of the user1access key for the entry with the given label:

```
11041 \newcommand*{\glentryuseriaccess}[1]{%
11042   \@gls@entry@field{#1}{useriaccess}%
11043 }
```

tryuseriiaccess Get the value of the user2access key for the entry with the given label:

```
11044 \newcommand*{\glentryuseriiaccess}[1]{%
11045   \@gls@entry@field{#1}{useriiaccess}%
11046 }
```

ryuseriiiaccess Get the value of the user3access key for the entry with the given label:

```
11047 \newcommand*{\glentryuseriiiaccess}[1]{%
11048   \@gls@entry@field{#1}{useriiiaccess}%
11049 }
```

tryuserivaccess Get the value of the user4access key for the entry with the given label:

```
11050 \newcommand*{\glentryuserivaccess}[1]{%
11051   \@gls@entry@field{#1}{userivaccess}%
11052 }
```

entryuservaccess Get the value of the user5access key for the entry with the given label:

```
11053 \newcommand*{\glentryuservaccess}[1]{%
11054   \@gls@entry@field{#1}{uservaccess}%
11055 }
```

tryuserviaccess Get the value of the user6access key for the entry with the given label:

```
11056 \newcommand*{\glentryuserviaccess}[1]{%
11057   \@gls@entry@field{#1}{userviaccess}%
11058 }
```

There are three types of replacement text:

Alt Description of some content that's non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, “St” could be short for “saint” or “street”).

Therefore, rather than having one command for all fields, it’s better to have a command dependent on the field type. For example, the short and shortpl keys would require E, the symbol key would require ActualText, and a field that contains an image would require Alt.

glsfieldaccsupp

`\glsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}`

Test if there’s a command called `\gls<field>accsupp`. If there is then use that otherwise use `\glsaccsupp`. The first argument should be the internal field label (not the key). The final argument is the entry label. If glossaries-extra has been loaded, this first checks for `\glsxtr<category><field>accsupp` and `\glsxtr<category>accsupp`.

```
11059 \newcommand{\glsfieldaccsupp}[4]{%
11060   \ifdef\glscategory
11061   {%
11062     \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
11063     {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
11064     {%
11065       \ifcsdef{glsxtr\glscategory{#4}accsupp}%
11066       {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
11067       {%
11068         \ifcsdef{gls#3accsupp}%
11069         {\csname gls#3accsupp\endcsname{#1}{#2}}%
11070         {\glsaccsupp{#1}{#2}}%
11071       }%
11072     }%
11073   }%
11074   {%
11075     \ifcsdef{gls#3accsupp}%
11076     {\csname gls#3accsupp\endcsname{#1}{#2}}%
11077     {\glsaccsupp{#1}{#2}}%
11078   }%
11079 }
```

glsfieldaccsupp

`\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}`

As `\glsfieldaccsupp` but fully expand replacement text.

```
11080 \newcommand{\xglsfieldaccsupp}[1]{%
11081   \protected@edef\@gls@replacementtext{#1}%
11082   \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
11083 }
```

glsshortaccsupp

`\glsshortaccsupp{<replacement text>}{<text>}`

```
11084 \newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

`\glsshortplaccsupp` `\glsshortplaccsupp{<replacement text>}{<text>}`

```
11085 \newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

`\glsaccsupp` `\glsaccsupp{<replacement text>}{<text>}`

```
11086 \newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
```

`\xglsaccsupp` Fully expands replacement text before calling `\glsaccsupp`

```
11087 \newcommand*{\xglsaccsupp}[2]{%
11088   \protected@edef\@gls@replacementtext{#1}%
11089   \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
11090 }
```

`@access@display` Deprecated. Use `\@gls@fieldaccess@display` instead.

```
11091 \newcommand*{\@gls@access@display}[2]{%
11092   \protected@edef\@glo@access{#2}%
11093   \ifx\@glo@access\@gls@noaccess
11094     #1%
11095   \else
11096     \xglsaccsupp{\@glo@access}{#1}%
11097   \fi
11098 }
```

`daccess@display` `\@gls@fieldaccess@display{<label>}{<field>}{<content>}{<replacement>}`

```
11099 \newcommand*{\@gls@fieldaccess@display}[4]{%
11100   \protected@edef\@glo@access{#4}%
11101   \ifdefequal\@glo@access\@gls@noaccess
11102     {#3}%
11103   {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
11104 }
```

`meaccessdisplay` Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```
11105 \newrobustcmd*{\glsnameaccessdisplay}[2]{%
11106   \ifcsundef{glo@glsdetoklabel{#2}@access}%
11107     {#1}%
11108   {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
11109 }
```

`xtaccessdisplay` As above but for the textaccess replacement text.

```
11110 \newrobustcmd*{\glsxtaccessdisplay}[2]{%
11111   \ifcsundef{glo@glsdetoklabel{#2}@textaccess}%
11112     {#1}%
11113 }
```

```

11113 {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
11114 }

alaccessdisplay As above but for the pluralaccess replacement text.
11115 \newrobustcmd*{\glspluralaccessdisplay}[2]{%
11116 \ifcsundef{glo@glstetoklabel{#2}@pluralaccess}%
11117 {#1}%
11118 {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
11119 }

staccessdisplay As above but for the firstaccess replacement text.
11120 \newrobustcmd*{\glsfirstaccessdisplay}[2]{%
11121 \ifcsundef{glo@glstetoklabel{#2}@firstaccess}%
11122 {#1}%
11123 {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
11124 }

alaccessdisplay As above but for the firstpluralaccess replacement text.
11125 \newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
11126 \ifcsundef{glo@glstetoklabel{#2}@firstpluralaccess}%
11127 {#1}%
11128 {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
11129 }

olaccessdisplay As above but for the symbolaccess replacement text.
11130 \newrobustcmd*{\glsymbolaccessdisplay}[2]{%
11131 \ifcsundef{glo@glstetoklabel{#2}@symbolaccess}%
11132 {#1}%
11133 {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
11134 }

alaccessdisplay As above but for the symbolpluralaccess replacement text.
11135 \newrobustcmd*{\glsymbolpluralaccessdisplay}[2]{%
11136 \ifcsundef{glo@glstetoklabel{#2}@symbolpluralaccess}%
11137 {#1}%
11138 {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
11139 }

onaccessdisplay As above but for the descriptionaccess replacement text.
11140 \newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
11141 \ifcsundef{glo@glstetoklabel{#2}@descaccess}%
11142 {#1}%
11143 {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
11144 }

alaccessdisplay As above but for the descriptionpluralaccess replacement text.
11145 \newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
11146 \ifcsundef{glo@glstetoklabel{#2}@descpluralaccess}%

```

```

11147 {#1}%
11148 {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glentrydescpluralaccess{#2}}}%
11149 }

```

rtaccessdisplay As above but for the shortaccess replacement text.

```

11150 \newrobustcmd*{\glsshortaccessdisplay}[2]{%
11151 \ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
11152 {#1}%
11153 {\@gls@fieldaccess@display{#2}{short}{#1}{\glentryshortaccess{#2}}}%
11154 }

```

alaccessdisplay As above but for the shortpluralaccess replacement text.

```

11155 \newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
11156 \ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
11157 {#1}%
11158 {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glentryshortpluralaccess{#2}}}%
11159 }

```

ngaccessdisplay As above but for the longaccess replacement text.

```

11160 \newrobustcmd*{\glslongaccessdisplay}[2]{%
11161 \ifcsundef{glo@glsdetoklabel{#2}@longaccess}%
11162 {#1}%
11163 {\@gls@fieldaccess@display{#2}{long}{#1}{\glentrylongaccess{#2}}}%
11164 }

```

alaccessdisplay As above but for the longpluralaccess replacement text.

```

11165 \newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
11166 \ifcsundef{glo@glsdetoklabel{#2}@longpluralaccess}%
11167 {#1}%
11168 {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glentrylongpluralaccess{#2}}}%
11169 }

```

riaccessdisplay As above but for the user1access replacement text.

```

11170 \newrobustcmd*{\glssuseriaccessdisplay}[2]{%
11171 \ifcsundef{glo@glsdetoklabel{#2}@useriaccess}%
11172 {#1}%
11173 {\@gls@fieldaccess@display{#2}{useri}{#1}{\glentryuseriaccess{#2}}}%
11174 }

```

iiaccessdisplay As above but for the user2access replacement text.

```

11175 \newrobustcmd*{\glssuseriiaccessdisplay}[2]{%
11176 \ifcsundef{glo@glsdetoklabel{#2}@useriiaccess}%
11177 {#1}%
11178 {\@gls@fieldaccess@display{#2}{userii}{#1}{\glentryuseriiaccess{#2}}}%
11179 }

```

iiiaccessdisplay As above but for the user3access replacement text.

```

11180 \newrobustcmd*{\glssuseriiiaccessdisplay}[2]{%

```

```

11181 \ifcsundef{glo@glsdetoklabel{#2}@useriiiaccess}%
11182 {#1}%
11183 {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glentryuseriiiaccess{#2}}}%
11184 }

```

ivaccessdisplay As above but for the user4access replacement text.

```

11185 \newrobustcmd*{\gluserivaccessdisplay}[2]{%
11186 \ifcsundef{glo@glsdetoklabel{#2}@userivaccess}%
11187 {#1}%
11188 {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glentryuserivaccess{#2}}}%
11189 }

```

rvaccessdisplay As above but for the user5access replacement text.

```

11190 \newrobustcmd*{\glservaccessdisplay}[2]{%
11191 \ifcsundef{glo@glsdetoklabel{#2}@uservaccess}%
11192 {#1}%
11193 {\@gls@fieldaccess@display{#2}{userv}{#1}{\glentryuservaccess{#2}}}%
11194 }

```

viaccessdisplay As above but for the user6access replacement text.

```

11195 \newrobustcmd*{\glserviaccessdisplay}[2]{%
11196 \ifcsundef{glo@glsdetoklabel{#2}@userviaccess}%
11197 {#1}%
11198 {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glentryuserviaccess{#2}}}%
11199 }

```

lsaccessdisplay Gets the replacement text corresponding to the named key given by the first argument and calls the appropriate command defined above.

```

11200 \newrobustcmd*{\glaccessdisplay}[3]{%
11201 \ifcsundef{gls#1accessdisplay}%
11202 {%
11203 \PackageError{glossaries-accsupp}{No accessibility support
11204 for key ‘#1’}{}%
11205 }%
11206 {%
11207 \csname gls#1accessdisplay\endcsname{#2}{#3}%
11208 }%
11209 }

```

default@entryfmt Redefine the default entry format to use accessibility information

```

11210 \renewcommand*{\@gls@default@entryfmt}[2]{%
11211 \ifdefempty{glscustomtext}
11212 {%
11213 \glsifplural
11214 {%
11215 \glscapscase
11216 {%

```

Don't adjust case

```
11217      \ifglsused\glslabel
11218      {%
```

Subsequent use

```
11219      #2{\glspluralaccessdisplay
11220          {\glsentryplural{\glslabel}}{\glslabel}}%
11221          {\glsdescriptionpluralaccessdisplay
11222              {\glsentrydescplural{\glslabel}}{\glslabel}}%
11223              {\glsymbolpluralaccessdisplay
11224                  {\glsentrysymbolplural{\glslabel}}{\glslabel}}
11225              {\glsinsert}}%
11226      }%
11227      {%
```

First use

```
11228      #1{\glsfirstpluralaccessdisplay
11229          {\glsentryfirstplural{\glslabel}}{\glslabel}}%
11230          {\glsdescriptionpluralaccessdisplay
11231              {\glsentrydescplural{\glslabel}}{\glslabel}}%
11232              {\glsymbolpluralaccessdisplay
11233                  {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
11234              {\glsinsert}}%
11235      }%
11236      }%
11237      {%
```

Make first letter upper case

```
11238      \ifglsused\glslabel
11239      {%
```

Subsequent use.

```
11240      #2{\glspluralaccessdisplay
11241          {\Glsentryplural{\glslabel}}{\glslabel}}%
11242          {\glsdescriptionpluralaccessdisplay
11243              {\glsentrydescplural{\glslabel}}{\glslabel}}%
11244              {\glsymbolpluralaccessdisplay
11245                  {\Glsentrysymbolplural{\glslabel}}{\glslabel}}%
11246              {\glsinsert}}%
11247      }%
11248      {%
```

First use

```
11249      #1{\glsfirstpluralaccessdisplay
11250          {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
11251          {\glsdescriptionpluralaccessdisplay
11252              {\glsentrydescplural{\glslabel}}{\glslabel}}%
11253              {\glsymbolpluralaccessdisplay
11254                  {\Glsentrysymbolplural{\glslabel}}{\glslabel}}%
11255              {\glsinsert}}%
11256      }%
```

11257 }%
 11258 {%

Make all upper case

11259 \ifglsused\glslabel
 11260 {%

Subsequent use

11261 \MakeUppercase{%
 11262 #2{\glspluralaccessdisplay
 11263 {\glsentryplural{\glslabel}}{\glslabel}}%
 11264 {\glsdescriptionpluralaccessdisplay
 11265 {\glsentrydescplural{\glslabel}}{\glslabel}}%
 11266 {\glsymbolpluralaccessdisplay
 11267 {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
 11268 {\glsinsert}}%
 11269 }%
 11270 {%

First use

11271 \MakeUppercase{%
 11272 #1{\glsfirstpluralaccessdisplay
 11273 {\glsentryfirstplural{\glslabel}}{\glslabel}}%
 11274 {\glsdescriptionpluralaccessdisplay
 11275 {\glsentrydescplural{\glslabel}}{\glslabel}}%
 11276 {\glsymbolpluralaccessdisplay
 11277 {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
 11278 {\glsinsert}}%
 11279 }%
 11280 }%
 11281 }%
 11282 {%

Singular form

11283 \glscapscase
 11284 {%

Don't adjust case

11285 \ifglsused\glslabel
 11286 {%

Subsequent use

11287 #2{\glstextaccessdisplay
 11288 {\glsentrytext{\glslabel}}{\glslabel}}%
 11289 {\glsdescriptionaccessdisplay
 11290 {\glsentrydesc{\glslabel}}{\glslabel}}%
 11291 {\glsymbolaccessdisplay
 11292 {\glsentrysymbol{\glslabel}}{\glslabel}}%
 11293 {\glsinsert}}%
 11294 }%
 11295 {%

First use

```

11296      #1{\glsfirstaccessdisplay
11297          {\glsentryfirst{\glslabel}}{\glslabel}}%
11298          {\glsdescriptionaccessdisplay
11299          {\glsentrydesc{\glslabel}}{\glslabel}}%
11300          {\glsymbolaccessdisplay
11301          {\glsentrysymbol{\glslabel}}{\glslabel}}%
11302          {\glsinsert}}%
11303      }%
11304  }%
11305  {%

```

Make first letter upper case

```

11306      \ifglsused\glslabel
11307      {%

```

Subsequent use

```

11308      #2{\glstextaccessdisplay
11309          {\Glsentrytext{\glslabel}}{\glslabel}}%
11310          {\glsdescriptionaccessdisplay
11311          {\glsentrydesc{\glslabel}}{\glslabel}}%
11312          {\glsymbolaccessdisplay
11313          {\glsentrysymbol{\glslabel}}{\glslabel}}%
11314          {\glsinsert}}%
11315      }%
11316  {%

```

First use

```

11317      #1{\glsfirstaccessdisplay
11318          {\Glsentryfirst{\glslabel}}{\glslabel}}%
11319          {\glsdescriptionaccessdisplay
11320          {\glsentrydesc{\glslabel}}{\glslabel}}%
11321          {\glsymbolaccessdisplay
11322          {\glsentrysymbol{\glslabel}}{\glslabel}}%
11323          {\glsinsert}}%
11324      }%
11325  }%
11326  {%

```

Make all upper case

```

11327      \ifglsused\glslabel
11328      {%

```

Subsequent use

```

11329      \MakeUppercase{%
11330      #2{\glstextaccessdisplay
11331          {\glsentrytext{\glslabel}}{\glslabel}}%
11332          {\glsdescriptionaccessdisplay
11333          {\glsentrydesc{\glslabel}}{\glslabel}}%
11334          {\glsymbolaccessdisplay
11335          {\glsentrysymbol{\glslabel}}{\glslabel}}%

```

```

11336         {\glsinsert}}}%
11337     }%
11338     {%

```

First use

```

11339     \MakeUppercase{%
11340         #1{\glsfirstaccessdisplay
11341             {\glsentryfirst{\glslabel}}{\glslabel}}}%
11342             {\glsdescriptionaccessdisplay
11343                 {\glsentrydesc{\glslabel}}{\glslabel}}}%
11344                 {\glsymbolaccessdisplay
11345                     {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11346                     {\glsinsert}}}%
11347     }%
11348 }%
11349 }%
11350 }%
11351 {%

```

Custom text provided in \glsdisp

```

11352     \ifglsused{\glslabel}%
11353     {%

```

Subsequent use

```

11354         #2{\glscustomtext}%
11355         {\glsdescriptionaccessdisplay
11356             {\glsentrydesc{\glslabel}}{\glslabel}}}%
11357             {\glsymbolaccessdisplay
11358                 {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11359                 {\glsinsert}}}%
11360     }%
11361     {%

```

First use

```

11362         #1{\glscustomtext}%
11363         {\glsdescriptionaccessdisplay
11364             {\glsentrydesc{\glslabel}}{\glslabel}}}%
11365             {\glsymbolaccessdisplay
11366                 {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11367                 {\glsinsert}}}%
11368     }%
11369 }%
11370 }

```

\glsгенentryfmt Redefine to use accessibility information.

```

11371 \renewcommand*{\glsгенentryfmt}{%
11372     \ifdefempty\glscustomtext
11373     {%
11374         \glsifplural
11375         {%

```

Plural form

```
11376      \glscapscase
11377      {%
```

Don't adjust case

```
11378      \ifglused\glslabel
11379      {%
```

Subsequent use

```
11380      \glspluralaccessdisplay
11381      {\glentryplural{\glslabel}}{\glslabel}%
11382      \glinsert
11383      }%
11384      {%
```

First use

```
11385      \glfirstpluralaccessdisplay
11386      {\glentryfirstplural{\glslabel}}{\glslabel}%
11387      \glinsert
11388      }%
11389      }%
11390      {%
```

Make first letter upper case

```
11391      \ifglused\glslabel
11392      {%
```

Subsequent use.

```
11393      \glspluralaccessdisplay
11394      {\Glsentryplural{\glslabel}}{\glslabel}%
11395      \glinsert
11396      }%
11397      {%
```

First use

```
11398      \glfirstpluralaccessdisplay
11399      {\Glsentryfirstplural{\glslabel}}{\glslabel}%
11400      \glinsert
11401      }%
11402      }%
11403      {%
```

Make all upper case

```
11404      \ifglused\glslabel
11405      {%
```

Subsequent use

```
11406      \glspluralaccessdisplay
11407      {\mfirstucMakeUppercase{\glentryplural{\glslabel}}}%
11408      {\glslabel}%
11409      \mfirstucMakeUppercase{\glinsert}%
11410      }%
11411      {%
```

First use

```

11412      \glsfirstpluralacessdisplay
11413      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
11414      {\glslabel}%
11415      \mfirstucMakeUppercase{\glsinsert}%
11416      }%
11417      }%
11418      }%
11419      {%

```

Singular form

```

11420      \glscapscase
11421      {%

```

Don't adjust case

```

11422      \ifglsused\glslabel
11423      {%

```

Subsequent use

```

11424      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
11425      \glsinsert
11426      }%
11427      {%

```

First use

```

11428      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
11429      \glsinsert
11430      }%
11431      }%
11432      {%

```

Make first letter upper case

```

11433      \ifglsused\glslabel
11434      {%

```

Subsequent use

```

11435      \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
11436      \glsinsert
11437      }%
11438      {%

```

First use

```

11439      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
11440      \glsinsert
11441      }%
11442      }%
11443      {%

```

Make all upper case

```

11444      \ifglsused\glslabel
11445      {%

```

Subsequent use

```
11446      \glstextaccessdisplay
11447      {\mfirstucMakeUppercase{\glentrytext{\glslabel}}}{\glslabel}%
11448      \mfirstucMakeUppercase{\glsinsert}%
11449      }%
11450      {%
```

First use

```
11451      \glsfirstaccessdisplay
11452      {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
11453      \mfirstucMakeUppercase{\glsinsert}%
11454      }%
11455      }%
11456      }%
11457      }%
11458      {%
```

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.

```
11459      \glscustomtext\glsinsert
11460      }%
11461      }
```

`\glsgenacfmt` Redefine to include accessibility information.

```
11462 \renewcommand*{\glsgenacfmt}{%
11463   \ifdefempty\glscustomtext
11464   {%
11465     \ifglsused\glslabel
11466     {%
```

Subsequent use:

```
11467      \glsifplural
11468      {%
```

Subsequent plural form:

```
11469      \glscapscase
11470      {%
```

Subsequent plural form, don't adjust case:

```
11471      \acronymfont
11472      {\glsshortpluralaccessdisplay
11473       {\glentryshortpl{\glslabel}}{\glslabel}}%
11474      \glsinsert
11475      }%
11476      {%
```

Subsequent plural form, make first letter upper case:

```
11477      \acronymfont
11478      {\glsshortpluralaccessdisplay
11479       {\Glsentryshortpl{\glslabel}}{\glslabel}}%
11480      \glsinsert
```

11481 }%
 11482 {%

Subsequent plural form, all caps:

11483 \mfirstucMakeUppercase
 11484 {\acronymfont
 11485 {\glsshortpluralaccessdisplay
 11486 {\glentryshortpl{\glslabel}}{\glslabel}}}%
 11487 \glsinsert}%
 11488 }%
 11489 }%
 11490 {%

Subsequent singular form

11491 \glscapscase
 11492 {%

Subsequent singular form, don't adjust case:

11493 \acronymfont
 11494 {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}}%
 11495 \glsinsert
 11496 }%
 11497 {%

Subsequent singular form, make first letter upper case:

11498 \acronymfont
 11499 {\glsshortaccessdisplay{\Glentryshort{\glslabel}}{\glslabel}}}%
 11500 \glsinsert
 11501 }%
 11502 {%

Subsequent singular form, all caps:

11503 \mfirstucMakeUppercase
 11504 {\acronymfont{%
 11505 \glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}}%
 11506 \glsinsert}%
 11507 }%
 11508 }%
 11509 }%
 11510 {%

First use:

11511 \glsifplural
 11512 {%

First use plural form:

11513 \glscapscase
 11514 {%

First use plural form, don't adjust case:

11515 \genplacrformat{\glslabel}{\glsinsert}%
 11516 }%
 11517 {%

First use plural form, make first letter upper case:

```
11518      \Genplacrfullformat{\glslabel}{\glsinsert}%
11519      }%
11520      {%
```

First use plural form, all caps:

```
11521      \mfirstucMakeUppercase
11522      {\genplacrfullformat{\glslabel}{\glsinsert}}%
11523      }%
11524      }%
11525      {%
```

First use singular form

```
11526      \glscapscase
11527      {%
```

First use singular form, don't adjust case:

```
11528      \genacrfullformat{\glslabel}{\glsinsert}%
11529      }%
11530      {%
```

First use singular form, make first letter upper case:

```
11531      \Genacrfullformat{\glslabel}{\glsinsert}%
11532      }%
11533      {%
```

First use singular form, all caps:

```
11534      \mfirstucMakeUppercase
11535      {\genacrfullformat{\glslabel}{\glsinsert}}%
11536      }%
11537      }%
11538      }%
11539      }%
11540      {%
```

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.

```
11541      \glscustomtext
11542      }%
11543 }
```

`\enacrfullformat` Redefine to include accessibility information.

```
11544 \renewcommand*{\genacrfullformat}[2]{%
11545   \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
11546   (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}{#1}})%
11547 }
```

`\enacrfullformat` Redefine to include accessibility information.

```
11548 \renewcommand*{\Genacrfullformat}[2]{%
11549   \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
11550   (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}{#1}})%
11551 }
```

placrfullformat Redefine to include accessibility information.

```
11552 \renewcommand*{\genplacrfullformat}[2]{%
11553   \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
11554   (\glsshortpluralaccessdisplay
11555     {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
11556 }
```

placrfullformat Redefine to include accessibility information.

```
11557 \renewcommand*{\Genplacrfullformat}[2]{%
11558   \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
11559   (\glsshortpluralaccessdisplay
11560     {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
11561 }
```

\@acrshort

```
11562 \def\@acrshort#1#2[#3]{%
11563   \glsdoifexists{#2}%
11564   {%
11565     \let\do@gls@link@checkfirsthyper\relax

11566     \let\glsifplural\@secondoftwo
11567     \let\glscapscase\@firstofthree
11568     \let\glsinsert\@empty
11569     \def\glscustomtext{%
11570       \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
11571     }%

    Call \@gls@link

11572     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11573   }%

11574   \glspostlinkhook
11575 }
```

\@Acrshort

```
11576 \def\@Acrshort#1#2[#3]{%
11577   \glsdoifexists{#2}%
11578   {%
11579     \let\do@gls@link@checkfirsthyper\relax

11580     \let\glsifplural\@secondoftwo
11581     \let\glscapscase\@secondofthree
11582     \let\glsinsert\@empty
11583     \def\glscustomtext{%
11584       \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
11585     }%

    Call \@gls@link

11586     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11587   }%
```



```

11588 \glspostlinkhook
11589 }

```

\@ACRshort

```

11590 \def\@ACRshort#1#2[#3]{%
11591   \glstoifexists{#2}%
11592   {%
11593     \let\do@gl@link@checkfirsthyper\relax

11594     \let\gl@sifplural\@secondoftwo
11595     \let\gl@scapscase\@thirdofthree
11596     \let\gl@insert\@empty
11597     \def\glscustomtext{%
11598       \acronymfont{\glsshortaccessdisplay
11599         {\MakeUppercase{\glentryshort{#2}}}{#2}}#3%
11600     }%

    Call\@gl@link

11601     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11602     }%

11603   \glspostlinkhook
11604 }

```

\@acrlong

```

11605 \def\@acrlong#1#2[#3]{%
11606   \glstoifexists{#2}%
11607   {%
11608     \let\do@gl@link@checkfirsthyper\relax

11609     \let\gl@sifplural\@secondoftwo
11610     \let\gl@scapscase\@firstofthree
11611     \let\gl@insert\@empty
11612     \def\glscustomtext{%
11613       \acronymfont{\glslongaccessdisplay{\glentrylong{#2}}{#2}}#3%
11614     }%

    Call\@gl@link

11615     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11616     }%

11617   \glspostlinkhook
11618 }

```

\@Acrlong

```

11619 \def\@Acrlong#1#2[#3]{%
11620   \glstoifexists{#2}%
11621   {%
11622     \let\do@gl@link@checkfirsthyper\relax

```

```

11623 \let\glsifplural\@secondoftwo
11624 \let\glsifcaps\@firstofthree
11625 \let\glsinsert\@empty
11626 \def\glsfont{\fontfamily\glsfontfamily\fontseries\glsfontseries\fontshape\glsfontshape\fontsize\glsfontsize\fontstyle\glsfontstyle}
11627 \def\glsfontfamily{\fontfamily\glsfontfamily}
11628 }%

```

Call \@gls@link

```

11629 \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
11630 }%

11631 \glspostlinkhook
11632 }

```

\@ACRlong

```

11633 \def\@ACRlong#1#2[#3]{%
11634 \glsdoifexists{#2}%
11635 {%
11636 \let\do@gls@link@checkfirsthyper\relax

11637 \let\glsifplural\@secondoftwo
11638 \let\glsifcaps\@firstofthree
11639 \let\glsinsert\@empty
11640 \def\glsfont{\fontfamily\glsfontfamily\fontseries\glsfontseries\fontshape\glsfontshape\fontsize\glsfontsize\fontstyle\glsfontstyle}
11641 \def\glsfontfamily{\fontfamily\glsfontfamily}
11642 \MakeUppercase{\glsentrylong{#2}}{#2}#3}%
11643 }%

```

Call \@gls@link

```

11644 \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
11645 }%

11646 \glspostlinkhook
11647 }

```

\@gls@text@

```

11648 \def\@gls@text@#1#2[#3]{%
11649 \gls@field@link{#1}{#2}{\gls@text@accessdisplay{\glsentrytext{#2}}{#2}#3}%
11650 }

```

\@Gls@text@

```

11651 \def\@Gls@text@#1#2[#3]{%
11652 \gls@field@link{#1}{#2}{\Gls@text@accessdisplay{\Glsentrytext{#2}}{#2}#3}%
11653 }

```

\@GLStext@

```

11654 \def\@GLStext@#1#2[#3]{%
11655 \gls@field@link{#1}{#2}%
11656 {\gls@text@accessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}{#2}%
11657 \mfirstucMakeUppercase{#3}}}%
11658 }

```

```

\@glsfirst@
11659 \def\@glsfirst@#1#2[#3]{%
11660   \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glentryfirst{#2}}{#2}#3}%
11661 }

\@Glsfirst@
11662 \def\@Glsfirst@#1#2[#3]{%
11663   \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
11664 }

\@GLSfirst@
11665 \def\@GLSfirst@#1#2[#3]{%
11666   \@gls@field@link{#1}{#2}%
11667   {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glentryfirst{#2}}}{#2}%
11668     \mfirstucMakeUppercase{#3}}%
11669 }

\@glsplural@
11670 \def\@glsplural@#1#2[#3]{%
11671   \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glentryplural{#2}}{#2}#3}%
11672 }

\@Glsplural@
11673 \def\@Glsplural@#1#2[#3]{%
11674   \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
11675 }

\@GLSplural@
11676 \def\@GLSplural@#1#2[#3]{%
11677   \@gls@field@link{#1}{#2}%
11678   {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glentryplural{#2}}}{#2}%
11679     \mfirstucMakeUppercase{#3}}%
11680 }

glsfirstplural@
11681 \def\@glsfirstplural@#1#2[#3]{%
11682   \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glentryfirstplural{#2}}{#2}#3}%
11683 }

Glsfirstplural@
11684 \def\@glsfirstplural@#1#2[#3]{%
11685   \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
11686 }

GLSfirstplural@
11687 \def\@GLSfirstplural@#1#2[#3]{%
11688   \@gls@field@link{#1}{#2}%
11689   {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glentryfirstplural{#2}}}{#2}%

```

```

11690     \mfirstucMakeUppercase{#3}}}%
11691 }

\@glsname@
11692 \def\@glsname@#1#2[#3]{%
11693   \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
11694 }

\@Glsname@
11695 \def\@Glsname@#1#2[#3]{%
11696   \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
11697 }

\@GLSname@
11698 \def\@GLSname@#1#2[#3]{%
11699   \@gls@field@link{#1}{#2}%
11700   {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
11701     \mfirstucMakeUppercase{#3}}}%
11702 }

\@glsdesc@
11703 \def\@glsdesc@#1#2[#3]{%
11704   \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
11705 }

\@Glsdesc@
11706 \def\@Glsdesc@#1#2[#3]{%
11707   \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
11708 }

\@GLSdesc@
11709 \def\@GLSdesc@#1#2[#3]{%
11710   \@gls@field@link{#1}{#2}%
11711   {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
11712     \mfirstucMakeUppercase{#3}}}%
11713 }

@glsdescplural@
11714 \def\@glsdescplural@#1#2[#3]{%
11715   \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
11716 }

@Glsdescplural@
11717 \def\@Glsdescplural@#1#2[#3]{%
11718   \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
11719 }

```

@GLSdescplural@

```
11720 \def\@GLSdescplural@#1#2[#3]{%
11721   \@gls@field@link{#1}{#2}%
11722   {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
11723   \mfirstucMakeUppercase{#3}}%
11724 }
```

\@glssymbol@

```
11725 \def\@glssymbol@#1#2[#3]{%
11726   \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
11727 }
```

\@GLssymbol@

```
11728 \def\@GLssymbol@#1#2[#3]{%
11729   \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
11730 }
```

\@GLSsymbol@

```
11731 \def\@GLSsymbol@#1#2[#3]{%
11732   \@gls@field@link{#1}{#2}%
11733   {\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
11734   \mfirstucMakeUppercase{#3}}%
11735 }
```

lssymbolplural@

```
11736 \def\@glssymbolplural@#1#2[#3]{%
11737   \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
11738 }
```

lssymbolplural@

```
11739 \def\@Glssymbolplural@#1#2[#3]{%
11740   \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
11741 }
```

LSymbolplural@

```
11742 \def\@GLSymbolplural@#1#2[#3]{%
11743   \@gls@field@link{#1}{#2}%
11744   {\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
11745   \mfirstucMakeUppercase{#3}}%
11746 }
```

\@glsuseri@

```
11747 \def\@glsuseri@#1#2[#3]{%
11748   \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
11749 }
```

\@Glsuseri@

```
11750 \def \@Glsuser@i#1#2[#3]{%
11751   \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
11752 }
```

\@GLSuseri@

```
11753 \def \@GLSuseri@#1#2[#3]{%
11754   \@gls@field@link{#1}{#2}%
11755   {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
11756     \mfirstucMakeUppercase{#3}}%
11757 }
```

\@glsuserii@

```
11758 \def \@glsuserii@#1#2[#3]{%
11759   \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
11760 }
```

\@Glsuserii@

```
11761 \def \@Glsuser@i#1#2[#3]{%
11762   \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
11763 }
```

\@GLSuserii@

```
11764 \def \@GLSuserii@#1#2[#3]{%
11765   \@gls@field@link{#1}{#2}%
11766   {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
11767     \mfirstucMakeUppercase{#3}}%
11768 }
```

\@glsuseriii@

```
11769 \def \@glsuseriii@#1#2[#3]{%
11770   \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
11771 }
```

\@Glsuseriii@

```
11772 \def \@Glsuser@i#1#2[#3]{%
11773   \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
11774 }
```

\@GLSuseriii@

```
11775 \def \@GLSuseriii@#1#2[#3]{%
11776   \@gls@field@link{#1}{#2}%
11777   {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
11778     \mfirstucMakeUppercase{#3}}%
11779 }
```

```

\@glsuseriv@
11780 \def \@glsuseriv@#1#2[#3]{%
11781   \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
11782 }

\@Glsuseriv@
11783 \def \@Glsuser@i#1#2[#3]{%
11784   \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
11785 }

\@GLSuseriv@
11786 \def \@GLSuseriv@#1#2[#3]{%
11787   \@gls@field@link{#1}{#2}%
11788   {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}{#2}%
11789     \mfirstucMakeUppercase{#3}}}%
11790 }

\@glsuserv@
11791 \def \@glsuserv@#1#2[#3]{%
11792   \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
11793 }

\@Glsuserv@
11794 \def \@Glsuser@i#1#2[#3]{%
11795   \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
11796 }

\@GLSuserv@
11797 \def \@GLSuserv@#1#2[#3]{%
11798   \@gls@field@link{#1}{#2}%
11799   {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}{#2}%
11800     \mfirstucMakeUppercase{#3}}}%
11801 }

\@glsuservi@
11802 \def \@glsuservi@#1#2[#3]{%
11803   \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
11804 }

\@Glsuservi@
11805 \def \@Glsuser@i#1#2[#3]{%
11806   \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
11807 }

\@GLSuservi@
11808 \def \@GLSuservi@#1#2[#3]{%
11809   \@gls@field@link{#1}{#2}%
11810   {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#2}}{#2}%
11811     \mfirstucMakeUppercase{#3}}}%
11812 }

```

5.3 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.

```
11813 \renewcommand*{\glossentryname}[1]{%
11814   \glsdoifexists{#1}%
11815   {%
11816     \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
11817   }%
11818 }

11819 \renewcommand*{\glossentryname}[1]{%
11820   \glsdoifexists{#1}%
11821   {%
11822     \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
11823   }%
11824 }

11825 \renewcommand*{\glossentrydesc}[1]{%
11826   \glsdoifexists{#1}%
11827   {%
11828     \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
11829   }%
11830 }

11831 \renewcommand*{\Glossentrydesc}[1]{%
11832   \glsdoifexists{#1}%
11833   {%
11834     \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
11835   }%
11836 }

11837 \renewcommand*{\glossentrysymbol}[1]{%
11838   \glsdoifexists{#1}%
11839   {%
11840     \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
11841   }%
11842 }

11843 \renewcommand*{\Glossentrysymbol}[1]{%
11844   \glsdoifexists{#1}%
11845   {%
11846     \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
11847   }%
11848 }
```

ssaryentryfield


```

11849 \newcommand*{\accsuppglossaryentryfield}[5]{%
11850   \glossaryentryfield{#1}%
11851   {\glslnameaccessdisplay{#2}{#1}}%
11852   {\glslsdescriptionaccessdisplay{#3}{#1}}%
11853   {\glslsymbolaccessdisplay{#4}{#1}}{#5}%
11854 }

```

rysubentryfield

```

11855 \newcommand*{\accsuppglossarysubentryfield}[6]{%
11856   \glossarysubentryfield{#1}{#2}%
11857   {\glslnameaccessdisplay{#3}{#2}}%
11858   {\glslsdescriptionaccessdisplay{#4}{#2}}%
11859   {\glslsymbolaccessdisplay{#5}{#2}}{#6}%
11860 }

```

5.4 Acronyms

Redefine acronym styles provided by glossaries:

long-short *<long>* (*<short>*) acronym style.

```

11861 \renewacronymstyle{long-short}%
11862 {%

```

Check for long form in case this is a mixed glossary.

```

11863   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
11864 }%
11865 {%
11866   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
11867   \renewcommand*{\genacrfullformat}[2]{%
11868     \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
11869     (\glsshortaccessdisplay
11870       {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
11871   }%
11872   \renewcommand*{\Genacrfullformat}[2]{%
11873     \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
11874     (\glsshortaccessdisplay
11875       {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
11876   }%
11877   \renewcommand*{\genplacrfullformat}[2]{%
11878     \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
11879     (\glsshortpluralaccessdisplay
11880       {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
11881   }%
11882   \renewcommand*{\Genplacrfullformat}[2]{%
11883     \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
11884     (\glsshortpluralaccessdisplay
11885       {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
11886   }%
11887   \renewcommand*{\acronymentry}[1]{%

```

```

11888 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
11889 \renewcommand*{\acronymsort}[2]{##1}%
11890 \renewcommand*{\acronymfont}[1]{##1}%
11891 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
11892 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
11893 }

```

short-long *<short>* (*<long>*) acronym style.

```

11894 \renewacronymstyle{short-long}%
11895 {%

```

Check for long form in case this is a mixed glossary.

```

11896 \ifglshaslong{\glslabel}{\glsacacfmt}{\glsacacentryfmt}%
11897 }%
11898 {%
11899 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
11900 \renewcommand*{\genacrfullformat}[2]{%
11901 \glsshortaccessdisplay
11902 {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
11903 (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
11904 }%
11905 \renewcommand*{\Genacrfullformat}[2]{%
11906 \glsshortaccessdisplay
11907 {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
11908 (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
11909 }%
11910 \renewcommand*{\genplacrfullformat}[2]{%
11911 \glsshortpluralaccessdisplay
11912 {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
11913 (\glslongpluralaccessdisplay
11914 {\glentrylongpl{##1}}{##1})%
11915 }%
11916 \renewcommand*{\Genplacrfullformat}[2]{%
11917 \glsshortpluralaccessdisplay
11918 {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
11919 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
11920 }%
11921 \renewcommand*{\acronymentry}[1]{%
11922 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
11923 \renewcommand*{\acronymsort}[2]{##1}%
11924 \renewcommand*{\acronymfont}[1]{##1}%
11925 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
11926 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
11927 }

```

long-short-desc *<long>* (*<short>*) acronym style that has an accompanying description (which the user needs to supply).

```

11928 \renewacronymstyle{long-short-desc}%
11929 {%

```

```

11930 \GlsUseAcrEntryDisplayStyle{long-short}%
11931 }%
11932 {%
11933 \GlsUseAcrStyleDefs{long-short}%
11934 \renewcommand*{\GenericAcronymFields}{}%
11935 \renewcommand*{\acronymsort}[2]{##2}%
11936 \renewcommand*{\acronymentry}[1]{%
11937 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
11938 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1)}}%
11939 }

```

g-sc-short-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

11940 \renewacronymstyle{long-sc-short-desc}%
11941 {%
11942 \GlsUseAcrEntryDisplayStyle{long-sc-short}%
11943 }%
11944 {%
11945 \GlsUseAcrStyleDefs{long-sc-short}%
11946 \renewcommand*{\GenericAcronymFields}{}%
11947 \renewcommand*{\acronymsort}[2]{##2}%
11948 \renewcommand*{\acronymentry}[1]{%
11949 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
11950 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1)}}%
11951 }

```

g-sm-short-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

11952 \renewacronymstyle{long-sm-short-desc}%
11953 {%
11954 \GlsUseAcrEntryDisplayStyle{long-sm-short}%
11955 }%
11956 {%
11957 \GlsUseAcrStyleDefs{long-sm-short}%
11958 \renewcommand*{\GenericAcronymFields}{}%
11959 \renewcommand*{\acronymsort}[2]{##2}%
11960 \renewcommand*{\acronymentry}[1]{%
11961 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
11962 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1)}}%
11963 }

```

short-long-desc *<short>* ({<long>}) acronym style that has an accompanying description (which the user needs to supply).

```

11964 \renewacronymstyle{short-long-desc}%
11965 {%
11966 \GlsUseAcrEntryDisplayStyle{short-long}%
11967 }%
11968 {%
11969 \GlsUseAcrStyleDefs{short-long}%

```

```

11970 \renewcommand*{\GenericAcronymFields}{}%
11971 \renewcommand*{\acronymsort}[2]{##2}%
11972 \renewcommand*{\acronymentry}[1]{%
11973     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
11974     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
11975 }

```

short-long-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

11976 \renewacronymstyle{sc-short-long-desc}%
11977 {%
11978     \GlsUseAcrEntryDispStyle{sc-short-long}%
11979 }%
11980 {%
11981     \GlsUseAcrStyleDefs{sc-short-long}%
11982     \renewcommand*{\GenericAcronymFields}{}%
11983     \renewcommand*{\acronymsort}[2]{##2}%
11984     \renewcommand*{\acronymentry}[1]{%
11985         \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
11986         (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
11987     }

```

short-long-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

11988 \renewacronymstyle{sm-short-long-desc}%
11989 {%
11990     \GlsUseAcrEntryDispStyle{sm-short-long}%
11991 }%
11992 {%
11993     \GlsUseAcrStyleDefs{sm-short-long}%
11994     \renewcommand*{\GenericAcronymFields}{}%
11995     \renewcommand*{\acronymsort}[2]{##2}%
11996     \renewcommand*{\acronymentry}[1]{%
11997         \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
11998         (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
11999     }

```

dua *<long>* only acronym style.

```

12000 \renewacronymstyle{dua}%
12001 {%

```

Check for long form in case this is a mixed glossary.

```

12002 \ifdefempty\glscustomtext
12003 {%
12004     \ifglshaslong{\glslabel}%
12005     {%
12006         \glsifplural
12007         {%

```

Plural form:

12008 `\glscapscase`
12009 `{%`

Plural form, don't adjust case:

12010 `\glslongpluralaccessdisplay{\glentrylongpl{\glslabel}}{\glslabel}%`
12011 `\glsinsert`
12012 `}%`
12013 `{%`

Plural form, make first letter upper case:

12014 `\glslongpluralaccessdisplay{\Glentrylongpl{\glslabel}}{\glslabel}%`
12015 `\glsinsert`
12016 `}%`
12017 `{%`

Plural form, all caps:

12018 `\glslongpluralaccessdisplay`
12019 `{\mfirstucMakeUppercase{\glentrylongpl{\glslabel}}}{\glslabel}%`
12020 `\mfirstucMakeUppercase{\glsinsert}%`
12021 `}%`
12022 `}%`
12023 `{%`

Singular form

12024 `\glscapscase`
12025 `{%`

Singular form, don't adjust case:

12026 `\glslongaccessdisplay{\glentrylong{\glslabel}}{\glslabel}\glsinsert`
12027 `}%`
12028 `{%`

Subsequent singular form, make first letter upper case:

12029 `\glslongaccessdisplay{\Glentrylong{\glslabel}}{\glslabel}\glsinsert`
12030 `}%`
12031 `{%`

Subsequent singular form, all caps:

12032 `\glslongaccessdisplay`
12033 `{\mfirstucMakeUppercase`
12034 `{\glentrylong{\glslabel}\glsinsert}}{\glslabel}%`
12035 `\mfirstucMakeUppercase{\glsinsert}%`
12036 `}%`
12037 `}%`
12038 `}%`
12039 `{%`

Not an acronym:

12040 `\glsgenentryfmt`
12041 `}%`
12042 `}%`

```

12043 {\glscustomtext\glsinsert}%
12044 }%
12045 {%
12046 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
12047 \renewcommand*{\acrfullfmt}[3]{%
12048   \glslink[##1]{##2}{%
12049     \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
12050     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
12051 \renewcommand*{\Acrfullfmt}[3]{%
12052   \glslink[##1]{##2}{%
12053     \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
12054     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
12055 \renewcommand*{\ACRfullfmt}[3]{%
12056   \glslink[##1]{##2}{%
12057     \glslongaccessdisplay
12058     {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
12059     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}}%
12060 \renewcommand*{\acrfullplfmt}[3]{%
12061   \glslink[##1]{##2}{%
12062     \glslongpluralaccessdisplay
12063     {\glsentrylongpl{##2}}{##2}##3\space
12064     (\glsshortpluralaccessdisplay
12065     {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
12066 \renewcommand*{\Acrfullplfmt}[3]{%
12067   \glslink[##1]{##2}{%
12068     \glslongpluralaccessdisplay
12069     {\Glsentrylongpl{##2}}{##2}##3\space
12070     (\glsshortpluralaccessdisplay
12071     {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
12072 \renewcommand*{\ACRfullplfmt}[3]{%
12073   \glslink[##1]{##2}{%
12074     \glslongpluralaccessdisplay
12075     {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
12076     (\glsshortpluralaccessdisplay
12077     {\acronymfont{\glsentryshortpl{##2}}}{##2})}}}%
12078 \renewcommand*{\glsentryfull}[1]{%
12079   \glslongaccessdisplay{\glsentrylong{##1}}\space
12080   (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
12081 }%
12082 \renewcommand*{\Glsentryfull}[1]{%
12083   \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
12084   (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
12085 }%
12086 \renewcommand*{\glsentryfullpl}[1]{%
12087   \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
12088   (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
12089 }%
12090 \renewcommand*{\Glsentryfullpl}[1]{%
12091   \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space

```

```

12092 (\glsshortpluralaccessdisplay{\acronymfont{\glentryshortpl{##1}}}{##1})%
12093 }%
12094 \renewcommand*{\acronymentry}[1]{%
12095     \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
12096 \renewcommand*{\acronymsort}[2]{##1}%
12097 \renewcommand*{\acronymfont}[1]{##1}%
12098 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
12099 }

```

dua-desc *<long>* only acronym style with user-supplied description.

```

12100 \renewacronymstyle{dua-desc}%
12101 {%
12102     \GlsUseAcrEntryDispStyle{dua}%
12103 }%
12104 {%
12105     \GlsUseAcrStyleDefs{dua}%
12106 \renewcommand*{\GenericAcronymFields}{}%
12107 \renewcommand*{\acronymentry}[1]{%
12108     \glslongaccessdisplay{\acronymfont{\glentrylong{##1}}}{##1}}%
12109 \renewcommand*{\acronymsort}[2]{##2}%
12110 }%

```

footnote *<short>*\footnote{*<long>*} acronym style.

```

12111 \renewacronymstyle{footnote}%
12112 {%
12113     Check for long form in case this is a mixed glossary.
12114     \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
12115 }%
12116 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

```

Need to ensure hyperlinks are switched off on first use:

```

12117 \glshyperfirstfalse
12118 \renewcommand*{\genacrfullformat}[2]{%
12119     \glsshortaccessdisplay
12120     {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2%
12121     \protect\footnote{\glslongaccessdisplay{\glentrylong{##1}}}{##1}}%
12122 }%
12123 \renewcommand*{\Genacrfullformat}[2]{%
12124     \glsshortaccessdisplay
12125     {\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
12126     \protect\footnote{\glslongaccessdisplay{\glentrylong{##1}}}{##1}}%
12127 }%
12128 \renewcommand*{\genplacrfullformat}[2]{%
12129     \glsshortpluralaccessdisplay
12130     {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2%
12131     \protect\footnote{\glslongpluralaccessdisplay{\glentrylongpl{##1}}}{##1}}%
12132 }%
12133 \renewcommand*{\Genplacrfullformat}[2]{%

```

```

12134 \glsshortpluralaccessdisplay
12135 {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
12136 \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
12137 }%
12138 \renewcommand*{\acronymentry}[1]{%
12139 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
12140 \renewcommand*{\acronymsort}[2]{##1}%
12141 \renewcommand*{\acronymfont}[1]{##1}%
12142 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%

```

Don't use footnotes for \acrfull:

```

12143 \renewcommand*{\acrfullfmt}[3]{%
12144 \glslink[##1]{##2}{%
12145 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
12146 (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
12147 \renewcommand*{\Acrfullfmt}[3]{%
12148 \glslink[##1]{##2}{%
12149 \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
12150 (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
12151 \renewcommand*{\ACRfullfmt}[3]{%
12152 \glslink[##1]{##2}{%
12153 \glsshortaccessdisplay
12154 {\mfirstucMakeUppercase
12155 {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
12156 (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}}%
12157 \renewcommand*{\acrfullplfmt}[3]{%
12158 \glslink[##1]{##2}{%
12159 \glsshortpluralaccessdisplay
12160 {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
12161 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
12162 \renewcommand*{\ACRfullplfmt}[3]{%
12163 \glslink[##1]{##2}{%
12164 \glsshortpluralaccessdisplay
12165 {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
12166 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
12167 \renewcommand*{\ACRfullplplfmt}[3]{%
12168 \glslink[##1]{##2}{%
12169 \glsshortpluralaccessdisplay
12170 {\mfirstucMakeUppercase
12171 {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
12172 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%

```

Similarly for \glsentryfull etc:

```

12173 \renewcommand*{\glsentryfull}[1]{%
12174 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
12175 (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
12176 \renewcommand*{\Glsentryfull}[1]{%
12177 \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
12178 (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
12179 \renewcommand*{\glsentryfullpl}[1]{%

```



```

12180 \glsshortpluralaccessdisplay
12181 {\acronymfont{\glentryshortpl{##1}}}{##1}\space
12182 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}%
12183 \renewcommand*{\Glsentryfullpl}[1]{%
12184 \glsshortpluralaccessdisplay
12185 {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
12186 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}%
12187 }

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

12188 \renewacronymstyle{footnote-sc}%
12189 {%
12190 \GlsUseAcrEntryDisplayStyle{footnote}%
12191 }%
12192 {%
12193 \GlsUseAcrStyleDefs{footnote}%
12194 \renewcommand{\acronymentry}[1]{%
12195 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
12196 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
12197 \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
12198 }%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```

12199 \renewacronymstyle{footnote-sm}%
12200 {%
12201 \GlsUseAcrEntryDisplayStyle{footnote}%
12202 }%
12203 {%
12204 \GlsUseAcrStyleDefs{footnote}%
12205 \renewcommand{\acronymentry}[1]{%
12206 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
12207 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
12208 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
12209 }%

```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

12210 \renewacronymstyle{footnote-desc}%
12211 {%
12212 \GlsUseAcrEntryDisplayStyle{footnote}%
12213 }%
12214 {%
12215 \GlsUseAcrStyleDefs{footnote}%
12216 \renewcommand*{\GenericAcronymFields}{}%
12217 \renewcommand*{\acronymsort}[2]{##2}%
12218 \renewcommand*{\acronymentry}[1]{%
12219 \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
12220 (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
12221 }

```

ootnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
12222 \renewacronymstyle{footnote-sc-desc}%
12223 {%
12224   \GlsUseAcrEntryDispStyle{footnote-sc}%
12225 }%
12226 {%
12227   \GlsUseAcrStyleDefs{footnote-sc}%
12228   \renewcommand*{\GenericAcronymFields}{}%
12229   \renewcommand*{\acronymsort}[2]{##2}%
12230   \renewcommand*{\acronymentry}[1]{%
12231     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12232     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
12233 }
```

ootnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
12234 \renewacronymstyle{footnote-sm-desc}%
12235 {%
12236   \GlsUseAcrEntryDispStyle{footnote-sm}%
12237 }%
12238 {%
12239   \GlsUseAcrStyleDefs{footnote-sm}%
12240   \renewcommand*{\GenericAcronymFields}{}%
12241   \renewcommand*{\acronymsort}[2]{##2}%
12242   \renewcommand*{\acronymentry}[1]{%
12243     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12244     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
12245 }
```

aultshortaccess \glsdefaultshortaccess{<long>}{<short>}

Default shortaccess value.

```
12246 \newcommand*{\glsdefaultshortaccess}[2]{#1}
```

Use \newacronymhook to modify the key list to set the access text to the long version by default.

```
12247 \renewcommand*{\newacronymhook}{%
12248   \protected@edef\@gls@keylist{%
12249     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12250     shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12251     \the\glskeylisttok}%
12252   \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
12253 }
```

ltNewAcronymDef Modify default style to use access text:

```
12254 \renewcommand*{\DefaultNewAcronymDef}{%
```

```

12255 \edef\@do@newglossaryentry{%
12256   \noexpand\newglossaryentry{\the\glslabeltok}%
12257   {%
12258     type=\acronymtype,%
12259     name={\the\glsshorttok},%
12260     description={\the\glslongtok},%
12261     descriptionaccess=\relax,
12262     text={\the\glsshorttok},%
12263     access={\noexpand\@glo@textaccess},%
12264     sort={\the\glsshorttok},%
12265     short={\the\glsshorttok},%
12266     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12267     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12268     long={\the\glslongtok},%
12269     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12270     descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12271     first={\noexpand\glslongaccessdisplay
12272       {\the\glslongtok}{\the\glslabeltok}\space
12273       (\noexpand\glsshortaccessdisplay
12274         {\the\glsshorttok}{\the\glslabeltok})},%
12275     plural={\the\glsshorttok\acrpluralsuffix},%
12276     firstplural={\noexpand\glslongpluralaccessdisplay
12277       {\noexpand\@glo@longpl}{\the\glslabeltok}\space
12278       (\noexpand\glsshortpluralaccessdisplay
12279         {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
12280     firstaccess=\relax,
12281     firstpluralaccess=\relax,
12282     textaccess={\noexpand\@glo@shortaccess},%
12283     \the\glskeylisttok
12284   }%
12285 }%
12286 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12287 \let\@org@gls@assign@plural\gls@assign@plural
12288 \let\@org@gls@assign@descplural\gls@assign@descplural
12289 \def\gls@assign@firstpl##1##2{%
12290   \@gls@expand@field{##1}{firstpl}{##2}%
12291 }%
12292 \def\gls@assign@plural##1##2{%
12293   \@gls@expand@field{##1}{plural}{##2}%
12294 }%
12295 \def\gls@assign@descplural##1##2{%
12296   \@gls@expand@field{##1}{descplural}{##2}%
12297 }%
12298 \@do@newglossaryentry
12299 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12300 \let\gls@assign@plural\@org@gls@assign@plural
12301 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12302 \@gls@noexpand@assign@default@plurals
12303 }

```

teNewAcronymDef

```

12304 \renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
12305   \edef\@do@newglossaryentry{%
12306     \noexpand\newglossaryentry{\the\glslabeltok}%
12307     {%
12308       type=\acronymtype,%
12309       name={\noexpand\acronymfont{\the\glsshorttok}},%
12310       sort={\the\glsshorttok},%
12311       text={\the\glsshorttok},%
12312       short={\the\glsshorttok},%
12313       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12314       shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12315       long={\the\glslongtok},%
12316       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12317       access={\noexpand\@glo@textaccess},%
12318       plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12319       symbol={\the\glslongtok},%
12320       symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12321       firstpluralaccess=\relax,
12322       textaccess={\noexpand\@glo@shortaccess},%
12323       \the\glskeylisttok
12324     }%
12325   }%
12326   \let\@org@gls@assign@firstpl\gls@assign@firstpl
12327   \let\@org@gls@assign@plural\gls@assign@plural
12328   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12329   \def\gls@assign@firstpl##1##2{%
12330     \@@gls@expand@field{##1}{firstpl}{##2}%
12331   }%
12332   \def\gls@assign@plural##1##2{%
12333     \@@gls@expand@field{##1}{plural}{##2}%
12334   }%
12335   \def\gls@assign@symbolplural##1##2{%
12336     \@@gls@expand@field{##1}{symbolplural}{##2}%
12337   }%
12338   \@do@newglossaryentry
12339   \let\gls@assign@plural\@org@gls@assign@plural
12340   \let\gls@assign@firstpl\@org@gls@assign@firstpl
12341   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12342 }

```

onNewAcronymDef

```

12343 \renewcommand*{\DescriptionNewAcronymDef}{%
12344   \edef\@do@newglossaryentry{%
12345     \noexpand\newglossaryentry{\the\glslabeltok}%
12346     {%
12347       type=\acronymtype,%
12348       name={\noexpand
12349         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%

```

```

12350     access={\noexpand\@glo@textaccess},%
12351     sort={\the\glsshorttok},%
12352     short={\the\glsshorttok},%
12353     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12354     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12355     long={\the\glslongtok},%
12356     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12357     first={\the\glslongtok},%
12358     firstaccess=\relax,
12359     firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12360     text={\the\glsshorttok},%
12361     textaccess={\the\glslongtok},%
12362     plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12363     symbol={\noexpand\@glo@text},%
12364     symbolaccess={\noexpand\@glo@textaccess},%
12365     symbolplural={\noexpand\@glo@plural},%
12366     firstpluralaccess=\relax,
12367     textaccess={\noexpand\@glo@shortaccess},%
12368     \the\glskeylisttok}%
12369 }%
12370 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12371 \let\@org@gls@assign@plural\gls@assign@plural
12372 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12373 \def\gls@assign@firstpl##1##2{%
12374   \@@gls@expand@field{##1}{firstpl}{##2}%
12375 }%
12376 \def\gls@assign@plural##1##2{%
12377   \@@gls@expand@field{##1}{plural}{##2}%
12378 }%
12379 \def\gls@assign@symbolplural##1##2{%
12380   \@@gls@expand@field{##1}{symbolplural}{##2}%
12381 }%
12382 \do@newglossaryentry
12383 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12384 \let\gls@assign@plural\@org@gls@assign@plural
12385 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12386 }

```

teNewAcronymDef

```

12387 \renewcommand*{\FootnoteNewAcronymDef}{%
12388   \edef\@do@newglossaryentry{%
12389     \noexpand\newglossaryentry{\the\glslabeltok}%
12390     {%
12391       type=\acronymtype,%
12392       name={\noexpand\acronymfont{\the\glsshorttok}},%
12393       sort={\the\glsshorttok},%
12394       text={\the\glsshorttok},%
12395       textaccess={\the\glslongtok},%
12396       access={\noexpand\@glo@textaccess},%

```

```

12397 plural={\the\glssshorttok\noexpand\acrpluralsuffix},%
12398 short={\the\glssshorttok},%
12399 shortplural={\the\glssshorttok\noexpand\acrpluralsuffix},%
12400 long={\the\glslongtok},%
12401 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12402 description={\the\glslongtok},%
12403 descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12404 \the\glskeylisttok
12405 }%
12406 }%
12407 \let\@org@gls@assign@plural\gls@assign@plural
12408 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12409 \let\@org@gls@assign@descplural\gls@assign@descplural
12410 \def\gls@assign@firstpl##1##2{%
12411   \@@gls@expand@field{##1}{firstpl}{##2}%
12412 }%
12413 \def\gls@assign@plural##1##2{%
12414   \@@gls@expand@field{##1}{plural}{##2}%
12415 }%
12416 \def\gls@assign@descplural##1##2{%
12417   \@@gls@expand@field{##1}{descplural}{##2}%
12418 }%
12419 \do@newglossaryentry
12420 \let\gls@assign@plural\@org@gls@assign@plural
12421 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12422 \let\gls@assign@descplural\@org@gls@assign@descplural
12423 }

```

11NewAcronymDef

```

12424 \renewcommand*{\SmallNewAcronymDef}{%
12425   \edef\@do@newglossaryentry{%
12426     \noexpand\newglossaryentry{\the\glslabeltok}%
12427     {%
12428       type=\acronymtype,%
12429       name={\noexpand\acronymfont{\the\glssshorttok}},%
12430       access={\noexpand\@glo@symbolaccess},%
12431       sort={\the\glssshorttok},%
12432       short={\the\glssshorttok},%
12433       shortplural={\the\glssshorttok\noexpand\acrpluralsuffix},%
12434       shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glssshorttok}},%
12435       long={\the\glslongtok},%
12436       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12437       text={\noexpand\@glo@short},%
12438       textaccess={\noexpand\@glo@shortaccess},%
12439       plural={\noexpand\@glo@shortpl},%
12440       first={\the\glslongtok},%
12441       firstaccess=\relax,%
12442       firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12443       description={\noexpand\@glo@first},%

```

```

12444     descriptionplural={\noexpand\@glo@firstplural},%
12445     symbol={\the\glsshorttok},%
12446     symbolaccess={\the\glslongtok},%
12447     symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12448     \the\glskeylisttok
12449   }%
12450 }%
12451 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12452 \let\@org@gls@assign@plural\gls@assign@plural
12453 \let\@org@gls@assign@descplural\gls@assign@descplural
12454 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12455 \def\gls@assign@firstpl##1##2{%
12456   \@@gls@expand@field{##1}{firstpl}{##2}%
12457 }%
12458 \def\gls@assign@plural##1##2{%
12459   \@@gls@expand@field{##1}{plural}{##2}%
12460 }%
12461 \def\gls@assign@descplural##1##2{%
12462   \@@gls@expand@field{##1}{descplural}{##2}%
12463 }%
12464 \def\gls@assign@symbolplural##1##2{%
12465   \@@gls@expand@field{##1}{symbolplural}{##2}%
12466 }%
12467 \do@newglossaryentry
12468 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12469 \let\gls@assign@plural\@org@gls@assign@plural
12470 \let\gls@assign@descplural\@org@gls@assign@descplural
12471 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12472 }

```

The following are kept for compatibility with versions before 3.0:

sshortaccesskey

```
12473 \newcommand*{\glsshortaccesskey}{\glsshortkey access}%
```

pluralaccesskey

```
12474 \newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
```

lslongaccesskey

```
12475 \newcommand*{\glslongaccesskey}{\glslongkey access}%
```

pluralaccesskey

```
12476 \newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

5.5 Debugging Commands

owglongnameaccess

```
12477 \newcommand*{\showglongnameaccess}[1]{%
```

```

12478 \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
12479 }

owglotextaccess

12480 \newcommand*{\showglotextaccess}[1]{%
12481 \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
12482 }

glopluralaccess

12483 \newcommand*{\showglopluralaccess}[1]{%
12484 \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
12485 }

wglofirstaccess

12486 \newcommand*{\showglofirstaccess}[1]{%
12487 \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
12488 }

rstpluralaccess

12489 \newcommand*{\showglofirstpluralaccess}[1]{%
12490 \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
12491 }

glosymbolaccess

12492 \newcommand*{\showglosymbolaccess}[1]{%
12493 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
12494 }

bolpluralaccess

12495 \newcommand*{\showglosymbolpluralaccess}[1]{%
12496 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
12497 }

owglodescaccess

12498 \newcommand*{\showglodescaccess}[1]{%
12499 \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
12500 }

escpluralaccess

12501 \newcommand*{\showglodescpluralaccess}[1]{%
12502 \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
12503 }

wgloshortaccess

12504 \newcommand*{\showgloshortaccess}[1]{%
12505 \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
12506 }

```


ortpluralaccess

```
12507 \newcommand*{\showgloshortpluralaccess}[1]{%  
12508   \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname  
12509 }
```

owglolongaccess

```
12510 \newcommand*{\showglolongaccess}[1]{%  
12511   \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname  
12512 }
```

ongpluralaccess

```
12513 \newcommand*{\showglolongpluralaccess}[1]{%  
12514   \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname  
12515 }
```

6 Multi-Lingual Support

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.

```
12516 \NeedsTeXFormat{LaTeX2e}
12517 \ProvidesPackage{glossaries-babel}[2021/09/20 v4.47 (NLCT)]
```

Load tracklang to obtain language settings.

```
12518 \RequirePackage{tracklang}
12519 \let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
12520 \AnyTrackedLanguages
12521 {%
12522   \ForEachTrackedDialect{\this@dialect}{%
12523     \IfTrackedLanguageFileExists{\this@dialect}%
12524       {glossaries-}% prefix
12525       {.ldf}%
12526       {%
12527         \RequireGlossariesLang{\CurrentTrackedTag}%
12528       }%
12529       {%
12530         \PackageWarningNoLine{glossaries}%
12531           {No language module detected for ‘\this@dialect’.\MessageBreak
12532             Language modules need to be installed separately.\MessageBreak
12533             Please check on CTAN for a bundle called\MessageBreak
12534             ‘glossaries-\CurrentTrackedLanguage’ or similar}%
12535       }%
12536     }%
12537   }%
12538 }
```

6.1 Polyglossia Captions

Language support has now been split off into independent language modules.

```
12539 \NeedsTeXFormat{LaTeX2e}
12540 \ProvidesPackage{glossaries-polyglossia}[2021/09/20 v4.47 (NLCT)]
```

Load tracklang to obtain language settings.

```
12541 \RequirePackage{tracklang}
12542 \let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
12543 \AnyTrackedLanguages
```

```

12544 {%
12545   \ForEachTrackedDialect{\this@dialect}{%
12546     \IfTrackedLanguageFileExists{\this@dialect}%
12547     {glossaries-}% prefix
12548     {.ldf}%
12549     {%
12550       \RequireGlossariesLang{\CurrentTrackedTag}%
12551     }%
12552     {%
12553       \PackageWarningNoLine{glossaries}%
12554       {No language module detected for ‘\this@dialect’.\MessageBreak
12555       Language modules need to be installed separately.\MessageBreak
12556       Please check on CTAN for a bundle called\MessageBreak
12557       ‘glossaries-\CurrentTrackedLanguage’ or similar}%
12558     }%
12559   }%
12560 }%
12561 {}%

```

6.2 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

12562 \NeedsTeXFormat{LaTeX2e}
12563 \ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]
12564 \RequirePackage{ifthen}
12565 \RequirePackage{xkeyval}[2006/11/18]
12566 \RequirePackage{mfirstuc}
12567 \RequirePackage{textcase}
12568 \renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
12569 \RequirePackage{xfor}
12570 \RequirePackage{datatool-base}
12571 \RequirePackage{amsgen}
12572 \RequirePackage{etoolbox}
12573 \newif\if@gls@docloaded
12574 \@ifpackageloaded{doc}%
12575 {%
12576   \@gls@docloadedtrue
12577 }%
12578 {%
12579   \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
12580 }
12581 \if@gls@docloaded
12582   \let\glsorg@theglossary\theglossary
12583   \let\glsorg@endtheglossary\endtheglossary
12584   \let\glsorg@PrintChanges\PrintChanges
12585   \renewcommand{\PrintChanges}{%
12586     \begingroup
12587     \let\theglossary\glsorg@theglossary

```

```

12588     \let\endtheglossary\glsorg@endtheglossary
12589     \glsorg@PrintChanges
12590 \endgroup
12591 }
12592 \fi
12593 \newif\if@gls@debug
12594 \define@choicekey{glossaries.sty}{debug}[\@gls@debug@val\@gls@debug@nr]{%
12595 {true,false,showtargets,showaccsupp}[true]{%
12596 \ifcase\@gls@debug@nr\relax
12597   % debug=true
12598   \@gls@debugtrue
12599   \renewcommand*\GlossariesWarning}[1]{%
12600     \PackageWarning{glossaries}{##1}%
12601   }%
12602   \renewcommand*\GlossariesWarningNoLine}[1]{%
12603     \PackageWarningNoLine{glossaries}{##1}%
12604   }%
12605   \let\@glsshowtarget\@gobble
12606   \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12607 \or
12608   % debug=false
12609   \@gls@debugfalse
12610   \let\@glsshowtarget\@gobble
12611   \let\@glsshowaccsupp\@gobblethree
12612   \PackageInfo{glossaries}{debug mode OFF}%
12613 \or
12614   % debug=showtargets
12615   \@gls@debugtrue
12616   \renewcommand*\GlossariesWarning}[1]{%
12617     \PackageWarning{glossaries}{##1}%
12618   }%
12619   \renewcommand*\GlossariesWarningNoLine}[1]{%
12620     \PackageWarningNoLine{glossaries}{##1}%
12621   }%
12622   \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12623   \renewcommand{\@glsshowtarget}{\@glsshowtarget}%
12624 \or
12625   % debug=showaccsupp
12626   \@gls@debugtrue
12627   \renewcommand*\GlossariesWarning}[1]{%
12628     \PackageWarning{glossaries}{##1}%
12629   }%
12630   \renewcommand*\GlossariesWarningNoLine}[1]{%
12631     \PackageWarningNoLine{glossaries}{##1}%
12632   }%
12633   \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12634   \renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
12635 \fi
12636 }

```

```

12637 \newcommand*{\glsshowtarget}[1]{%
12638   \ifmode
12639     \nfss@text{\glsshowtargetfont [1]}%
12640   \else
12641     \ifinner
12642       {\glsshowtargetfont [1]}%
12643     \else
12644       \glsshowtargetouter{1}%
12645     \fi
12646   \fi
12647 }
12648 \newcommand*{\glsshowtargetouter}[1]{%
12649   \glsshowtargetsymbols\marginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
12650 \newcommand*{\glsshowtargetsymbols}{\tiny$\triangleright$}
12651 \newcommand*{\glsshowtargetfont}{\ttfamily\footnotesize}
12652 \newcommand*{\@glsshowtarget}[1]{%
12653   \newrobustcmd*{\@@glsshowtarget}[1]{%
12654     \begingroup
12655     \protected@edef\@gls@tmp{#1}%
12656     \@onelevel@sanitize\@gls@tmp
12657     \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
12658     \endgroup
12659   }
12660   \newcommand*{\@glsshowaccsupp}[3]{%
12661     \newrobustcmd*{\glsshowaccsupp}[3]{%
12662       \ifstrepty{#1}%
12663         {\@@glsshowtarget{/#2 (#3)}}%
12664         {\@@glsshowtarget{/#2 (#3) [1]}}%
12665       }
12666     \newcommand*{\@gls@see@noindex}{%
12667       \PackageError{glossaries}%
12668       {'\gls@xr@key' key may only be used after \string\makeglossaries\space
12669       or \string\makenoidxglossaries\space (or move
12670       \string\newglossaryentry\space
12671       definitions into the preamble)}}%
12672     {You must use \string\makeglossaries\space
12673     or \string\makenoidxglossaries\space before defining
12674     any entries that have a '\gls@xr@key' key. It may
12675     be that the 'see' key has been written to the .glsdefs
12676     file from the previous run, in which case you need to
12677     move your definitions
12678     to the preamble if you don't want to use
12679     \string\makeglossaries\space
12680     or \string\makenoidxglossaries}%
12681   }
12682   \define@choicekey{glossaries.sty}{seenindex}%
12683   [\gls@seenindex@val\gls@seenindex@nr]{error,warn,ignore}{%
12684     \ifcase\gls@seenindex@nr
12685       \renewcommand*{\@gls@see@noindex}{%

```

```

12686 \PackageError{glossaries}%
12687 {'\gls@xr@key' key may only be used after \string\makeglossaries\space
12688 or \string\makenoidxglossaries}%
12689 {You must use \string\makeglossaries\space
12690 or \string\makenoidxglossaries\space before defining
12691 any entries that have a '\gls@xr@key' key}%
12692 }%
12693 \or
12694 \renewcommand*{\@gls@see@noindex}{%
12695 \GlossariesWarning{'\gls@xr@key' key ignored}%
12696 }%
12697 \or
12698 \renewcommand*{\@gls@see@noindex}{}%
12699 \fi
12700 }
12701 \define@boolkey{glossaries.sty}[gls]{toc}[true]{}
12702 \define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
12703 \ifcsundef{chapter}%
12704 {\newcommand*{\@@glossarysec}{section}}%
12705 {\newcommand*{\@@glossarysec}{chapter}}
12706 \define@choicekey{glossaries.sty}{section}{part,chapter,section,%
12707 subsection,subsubsection,paragraph,subparagraph}[section]{}
12708 \renewcommand*{\@@glossarysec}{#1}}
12709 \newcommand*{\@@glossarysecstar}{*}
12710 \newcommand*{\@@glossaryseclabel}{}
12711 \newcommand*{\glsautoprefix}{}
12712 \define@choicekey{glossaries.sty}{numberedsection}%
12713 [\gls@numberedsection@val\gls@numberedsection@nr]{}%
12714 false,nolabel,autolabel,nameref}[nolabel]{}
12715 \ifcase\gls@numberedsection@nr\relax
12716 \renewcommand*{\@@glossarysecstar}{*}%
12717 \renewcommand*{\@@glossaryseclabel}{}%
12718 \or
12719 \renewcommand*{\@@glossarysecstar}{}%
12720 \renewcommand*{\@@glossaryseclabel}{}%
12721 \or
12722 \renewcommand*{\@@glossarysecstar}{}%
12723 \renewcommand*{\@@glossaryseclabel}{%
12724 \label{\glsautoprefix\@glo@type}}%
12725 \or
12726 \renewcommand*{\@@glossarysecstar}{*}%
12727 \renewcommand*{\@@glossaryseclabel}{%
12728 \protected@edef\@currentlabelname{\glossarytoctitle}%
12729 \label{\glsautoprefix\@glo@type}}%
12730 \fi
12731 }
12732 \@ifpackageloaded{classicthesis}
12733 {\newcommand*{\@glossary@default@style}{index}}
12734 {\newcommand*{\@glossary@default@style}{list}}

```

```

12735 \define@key{glossaries.sty}{style}{%
12736   \def\@glossary@default@style{#1}%
12737 }
12738 \newcommand*\@gls@declareoption}[2]{%
12739   \DeclareOptionX{#1}{#2}%
12740   \DeclareOption{#1}{#2}%
12741 }
12742 \newcommand*\@glossaryentrynumbers}[1]{#1\@gls@save@numberlist{#1}}
12743 \@gls@declareoption{nonumberlist}{%
12744   \renewcommand*\@glossaryentrynumbers}[1]{\@gls@save@numberlist{#1}}%
12745 }
12746 \define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
12747   \glssavenumberlistfalse
12748 \newcommand*\@glo@seeautonumberlist{%
12749   \@gls@declareoption{seeautonumberlist}{%
12750     \renewcommand*\@glo@seeautonumberlist{%
12751       \def\@glo@prefix{\glsnextpages}%
12752     }%
12753 }
12754 \define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
12755   \glsclocationstrue
12756 \newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
12757 \@gls@declareoption{nolong}{\renewcommand*\@gls@loadlong{}}
12758 \IfFileExists{supertabular.sty}{%
12759   \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}%
12760 \newcommand*\@gls@loadsuper{}
12761 \@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}
12762 \newcommand*\@gls@loadlist{\RequirePackage{glossary-list}}
12763 \@gls@declareoption{nolist}{%
12764   \renewcommand*\@gls@loadlist{%
12765     \ifdefstring{\@glossary@default@style}{list}%
12766     {\let\@glossary@default@style\relax}%
12767   }%
12768 }%
12769 }
12770 \newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}}
12771 \@gls@declareoption{notree}{\renewcommand*\@gls@loadtree{}}
12772 \@gls@declareoption{nostyles}{%
12773   \renewcommand*\@gls@loadlong}{%
12774   \renewcommand*\@gls@loadsuper}{%
12775   \renewcommand*\@gls@loadlist}{%
12776   \renewcommand*\@gls@loadtree}{%
12777   \let\@glossary@default@style\relax
12778 }
12779 \newcommand*\@gls@postdescription{%
12780   \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
12781 }
12782 \define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
12783   \glsnopostdotfalse

```

```

12784 \define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
12785 \glsnogroupskipfalse
12786 \define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
12787 \@ifclassloaded{memoir}
12788 {%
12789   \glsucmarktrue
12790 }%
12791 {%
12792   \glsucmarkfalse
12793 }
12794 \newcommand*{\@gls@define@glossaryentrycounter}{%
12795   \ifgl Sentrycounter
12796     \ifundef\c@glossaryentry
12797     {%
12798       \ifx\@gls@counterwithin\@empty
12799         \newcounter{glossaryentry}%
12800       \else
12801         \newcounter{glossaryentry}[\@gls@counterwithin]%
12802       \fi
12803       \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
12804     }%
12805   {}%
12806 \fi
12807 }
12808 \define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
12809 \gl Sentrycounterfalse
12810 \define@key{glossaries.sty}{counterwithin}{%
12811   \renewcommand*{\@gls@counterwithin}{#1}%
12812   \gl Sentrycountertrue
12813   \@gls@define@glossaryentrycounter
12814 }
12815 \newcommand*{\@gls@counterwithin}{}
12816 \newcommand{\@gls@define@glossarysubentrycounter}{%
12817   \ifundef\c@glossarysubentry
12818   {%
12819     \ifglssubentrycounter
12820     \ifgl Sentrycounter
12821       \newcounter{glossarysubentry}[glossaryentry]%
12822     \else
12823       \newcounter{glossarysubentry}%
12824     \fi
12825     \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
12826   \fi
12827 }%
12828 {}%
12829 }
12830 \define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
12831 \glssubentrycounterfalse
12832 \newcommand*{\@glo@default@sorttype}{standard}

```



```

12833 \define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
12834   \renewcommand*{\@glo@default@sorttype}{#1}%
12835   \csname @gls@setupsort@#1\endcsname
12836 }
12837 \newcommand*{\glsprestandardsort}[3]{%
12838   \glsdosanitizesort
12839 }
12840 \newcommand*{\@glo@check@sortallowed}[1]{%
12841 \newcommand*{\@gls@setupsort@standard}{%
12842   \def\do@glo@storeentry{\@glo@storeentry}%
12843   \def\@gls@defsortcount##1{%
12844     \def\@gls@defsort##1##2{%
12845       \ifx\@glo@sort\@glsdefaultsort
12846         \let\@glo@sort\@glo@name
12847       \fi
12848       \let\glsdosanitizesort\@gls@sanitizesort
12849       \glsprestandardsort{\@glo@sort}{##1}{##2}%
12850       \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
12851     }%
12852     \def\@gls@setsort##1{%
12853       \let\@glo@check@sortallowed\@gobble
12854     }
12855     \@gls@setupsort@standard
12856 \newcommand*{\glssortnumberfmt}[1]{%
12857   \ifnum#1<100000 0\fi
12858   \ifnum#1<10000 0\fi
12859   \ifnum#1<1000 0\fi
12860   \ifnum#1<100 0\fi
12861   \ifnum#1<10 0\fi
12862   \number#1%
12863 }
12864 \newcommand*{\@gls@setupsort@def}{%
12865   \def\do@glo@storeentry{\@glo@storeentry}%
12866   \def\@gls@defsortcount##1{%
12867     \expandafter\global
12868     \expandafter\newcount\csname glossary@##1@sortcount\endcsname
12869   }%
12870   \def\@gls@defsort##1##2{%
12871     \ifcsundef{glossary@##1@sortcount}%
12872     {\@gls@defsortcount{##1}}%
12873     {}%
12874     \expandafter\global\expandafter
12875     \advance\csname glossary@##1@sortcount\endcsname by 1\relax
12876     \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
12877       \expandafter\glssortnumberfmt
12878       {\csname glossary@##1@sortcount\endcsname}}%
12879   }%
12880   \def\@gls@setsort##1{%
12881     \let\@glo@check@sortallowed\@gobble

```

```

12882 }
12883 \newcommand*{\@gls@setupsort@use}{%
12884   \let\do@glo@storeentry\@gobble
12885   \def\@gls@defsortcount##1{%
12886     \expandafter\global
12887     \expandafter\newcount\csname glossary@##1@sortcount\endcsname
12888   }%
12889   \def\@gls@defsort##1##2{%
12890     \expandafter\gdef\csname glo@##2@sort\endcsname{%}%
12891   }%
12892   \def\@gls@setsort##1{%
12893     \edef\@glo@parent{\csname glo@##1@parent\endcsname}%
12894     \ifx\@glo@parent\@empty
12895       \else
12896         \expandafter\@gls@setsort\expandafter{\@glo@parent}%
12897       \fi
12898     \edef\@glo@type{\csname glo@##1@type\endcsname}%
12899     \edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
12900     \ifx\@gls@tmp\@empty
12901       \expandafter\global\expandafter
12902       \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
12903       \expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
12904         \expandafter\glssortnumberfmt
12905         {\csname glossary@\@glo@type @sortcount\endcsname}}%
12906       \@glo@storeentry{##1}%
12907     \fi
12908   }%
12909   \let\@glo@check@sortallowed\@gobble
12910 }
12911 \newcommand*{\@gls@setupsort@none}{%
12912   \def\do@glo@storeentry##1{}%
12913   \def\@gls@defsortcount##1{}%
12914   \def\@gls@defsort##1##2{%
12915     \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
12916   }%
12917   \def\@gls@setsort##1{}%
12918   \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
12919     {Option sort=none not allowed with \string##1}%
12920     {(Use sort=def instead)}}%
12921 }
12922 \newcommand*{\glsdefmain}{%
12923   \if@gls@docloaded
12924     \newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
12925   \else
12926     \newglossary{main}{gls}{glo}{\glossaryname}%
12927   \fi
12928   \newcommand*{\gls@tr@set@main@toctitle}{%
12929     \translatelet{\glossarytoctitle}{Glossary}%
12930   }%

```

```

12931 }
12932 \newcommand*{\glsdefaulttype}{main}
12933 \newcommand*{\acronymtype}{\glsdefaulttype}
12934 \@gls@declareoption{nomain}{%
12935   \let\glsdefaulttype\relax
12936   \renewcommand*{\glsdefmain}{}}%
12937 }
12938 \define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
12939   \ifglsacronym
12940     \renewcommand*{\@gls@do@acronymsdef}{%
12941       \DeclareAcronymList{acronym}%
12942       \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
12943       \renewcommand*{\acronymtype}{acronym}%
12944       \newcommand*{\gls@tr@set@acronym@toctitle}{%
12945         \translatelet{\glossarytoctitle}{Acronyms}%
12946       }%
12947     }%
12948   \else
12949     \let\@gls@do@acronymsdef\relax
12950   \fi
12951 }
12952 \AtBeginDocument{%
12953   \ifglsacronym
12954     \ifbool{glscompatible-3.07}{%
12955       {}%
12956     }{%
12957       \providecommand*{\printacronyms}[1][1]{%
12958         \printglossary[type=\acronymtype,#1]}%
12959     }%
12960   \fi
12961 }
12962 \newcommand*{\@gls@do@acronymsdef}{}
12963 \@gls@declareoption{acronyms}{%
12964   \glsacronymtrue
12965   \def\@gls@do@acronymsdef{%
12966     \DeclareAcronymList{acronym}%
12967     \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
12968     \renewcommand*{\acronymtype}{acronym}%
12969     \newcommand*{\gls@tr@set@acronym@toctitle}{%
12970       \translatelet{\glossarytoctitle}{Acronyms}%
12971     }%
12972   }%
12973 }
12974 \newcommand*{\@glsacronymlists}{}
12975 \newcommand*{\@addtoacronymlists}[1]{%
12976   \ifx\@glsacronymlists\@empty
12977     \protected@xdef\@glsacronymlists{#1}%
12978   \else
12979     \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%

```

```

12980 \fi
12981 }
12982 \newcommand*\DeclareAcronymList[1]{%
12983   \glsIfListOfAcronyms{#1}{-}{\@addtoacronymlists{#1}}%
12984 }
12985 \newcommand{\glsIfListOfAcronyms}[1]{%
12986   \edef\@do@gl@s@islistofacronyms{%
12987     \noexpand\@gl@s@islistofacronyms{#1}{\@gl@sacronymlists}}%
12988   \@do@gl@s@islistofacronyms
12989 }
12990 \newcommand{\@gl@s@islistofacronyms}[4]{%
12991   \def\gl@s@islistofacronyms##1,#1,##2\end@gl@s@islistofacronyms{%
12992     \def\@gl@s@before{##1}\def\@gl@s@after{##2}}%
12993   \gl@s@islistofacronyms,#2,#1,\@nil\end@gl@s@islistofacronyms
12994   \ifx\@gl@s@after\@nnil
12995     #4%
12996   \else
12997     #3%
12998   \fi
12999 }
13000 \newif\if@gl@sisacronymlist
13001 \newcommand*\@gl@s@checkisacronymlist[1]{%
13002   \glsIfListOfAcronyms{#1}%
13003   {\@gl@sisacronymlisttrue}{\@gl@sisacronymlistfalse}%
13004 }
13005 \newcommand*\@SetAcronymLists[1]{%
13006   \renewcommand*\@gl@sacronymlists{#1}%
13007 }
13008 \define@key{glossaries.sty}{acronymlists}{%
13009   \DeclareAcronymList{#1}%
13010 }
13011 \newcommand{\glscounter}{page}
13012 \define@key{glossaries.sty}{counter}{%
13013   \renewcommand*\glscounter{#1}%
13014 }
13015 \newcommand*\@gl@s@nohyperlist{}
13016 \newcommand*\GlsDeclareNoHyperList[1]{%
13017   \ifdefempty\@gl@s@nohyperlist
13018     {%
13019       \renewcommand*\@gl@s@nohyperlist{#1}%
13020     }%
13021     {%
13022       \appto\@gl@s@nohyperlist{,#1}%
13023     }%
13024 }
13025 \define@key{glossaries.sty}{nohypertypes}{%
13026   \GlsDeclareNoHyperList{#1}%
13027 }
13028 \newcommand*\GlossariesWarning[1]{%

```

```

13029 \PackageWarning{glossaries}{#1}%
13030 }
13031 \newcommand*\GlossariesWarningNoLine[1]{%
13032 \PackageWarningNoLine{glossaries}{#1}%
13033 }
13034 \newcommand{\glosortentrieswarning}{%
13035 \typeout{Using TeX to sort glossary entries---this may
13036 take a while}%
13037 }
13038 \@gls@declareoption{nowarn}{%
13039 \if@gls@debug
13040 \GlossariesWarning{Warnings can't be suppressed in debug mode}%
13041 \else
13042 \renewcommand*\GlossariesWarning[1]{}%
13043 \renewcommand*\GlossariesWarningNoLine[1]{}%
13044 \renewcommand*\glosortentrieswarning{}%
13045 \renewcommand*\@gls@missinglang@warn[2]{}%
13046 \fi
13047 }
13048 \newcommand*\@gls@missinglang@warn[2]{%
13049 \PackageWarningNoLine{glossaries}%
13050 {No language module detected for '#1'.\MessageBreak
13051 Language modules need to be installed separately.\MessageBreak
13052 Please check on CTAN for a bundle called\MessageBreak
13053 'glossaries-#2' or similar}%
13054 }
13055 \@gls@declareoption{nolangwarn}{%
13056 \renewcommand*\@gls@missinglang@warn[2]{}%
13057 }
13058 \newcommand*\@gls@warnonglossdefined{%
13059 \GlossariesWarning{Overriding \string\printglossary}%
13060 }
13061 \newcommand*\@gls@warnontheGLOSSdefined{%
13062 \GlossariesWarning{Overriding 'theGLOSS' environment}%
13063 }
13064 \@gls@declareoption{noredefwarn}{%
13065 \renewcommand*\@gls@warnonglossdefined{}%
13066 \renewcommand*\@gls@warnontheGLOSSdefined{}%
13067 }
13068 \newcommand*\@gls@sanitizedesc{%
13069 }
13070 \newcommand*\glsssetexpandfield[1]{%
13071 \csdef{gls@assign@#1@field}##1##2{%
13072 \@@gls@expand@field{##1}{#1}{##2}%
13073 }%
13074 }
13075 \newcommand*\glsssetnoexpandfield[1]{%
13076 \csdef{gls@assign@#1@field}##1##2{%
13077 \@@gls@noexpand@field{##1}{#1}{##2}%

```

```

13078 }%
13079 }
13080 \glssetexpandfield{type}
13081 \glssetnoexpandfield{desc}
13082 \glssetnoexpandfield{descplural}
13083 \newcommand*{\@gls@sanitizename}{%
13084 \glssetnoexpandfield{name}
13085 \newcommand*{\@gls@sanitizesymbol}{%
13086 \glssetnoexpandfield{symbol}
13087 \glssetnoexpandfield{symbolplural}
13088 \newcommand*{\@gls@sanitizesort}{%
13089 \ifglssanitizesort
13090 \@@gls@sanitizesort
13091 \else
13092 \@@gls@nosanitizesort
13093 \fi
13094 }
13095 \newcommand*{\@@gls@sanitizesort{%
13096 \@onelevel@sanitize\@glo@sort
13097 }
13098 \newcommand*{\@@gls@nosanitizesort}{%
13099 \newcommand*{\@gls@noidx@sanitizesort{%
13100 \ifdefvoid\@glo@sort
13101 }%
13102 {%
13103 \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
13104 }%
13105 }
13106 \def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
13107 \def\@glo@sort{#1#2}%
13108 \@onelevel@sanitize\@glo@sort
13109 }
13110 \newcommand*{\@@gls@noidx@nosanitizesort}{%
13111 \ifdefvoid\@glo@sort
13112 }%
13113 {%
13114 \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
13115 }%
13116 }
13117 \def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
13118 \bgroup
13119 \glsnoidxstripaccents
13120 \protected@xdef\@@glo@sort{#1#2}%
13121 \egroup
13122 \let\@glo@sort\@@glo@sort
13123 }
13124 \newcommand*\glsnoidxstripaccents{%
13125 \let\IeC\@firstofone
13126 \let\add@accent@\@secondoftwo

```

```

13127 \let\@text@composite@x\@secondoftwo
13128 \let\@tabacckludge\@secondoftwo
13129 \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
13130 \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
13131 \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
13132 \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
13133 \let\'\'@firstofone
13134 \let\'\'@firstofone
13135 \let\~\@firstofone
13136 \let\"\"@firstofone
13137 \let\u\"@firstofone
13138 \let\t\"@firstofone
13139 \let\d\"@firstofone
13140 \let\r\"@firstofone
13141 \let\=\"@firstofone
13142 \let\.\@firstofone
13143 \let\~\@firstofone
13144 \let\v\"@firstofone
13145 \let\H\"@firstofone
13146 \let\c\"@firstofone
13147 \let\b\"@firstofone
13148 \let\a\"@secondoftwo
13149 \def\AE{AE}%
13150 \def\ae{ae}%
13151 \def\OE{OE}%
13152 \def\oe{oe}%
13153 \def\AA{AA}%
13154 \def\aa{aa}%
13155 \def\L{L}%
13156 \def\l{l}%
13157 \def\O{O}%
13158 \def\o{o}%
13159 \def\SS{SS}%
13160 \def\ss{ss}%
13161 \def\th{th}%
13162 \def\TH{TH}%
13163 \def\dh{dh}%
13164 \def\DH{DH}%
13165 }
13166 \@ifl@t@r\fmtversion{2019/10/01}
13167 {%
13168 \appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
13169 }
13170 {}
13171 \define@boolkey[glS]{sanitize}{description}[true]{%
13172 \GlossariesWarning{sanitize={description} package option deprecated}%
13173 \ifglS@sanitize@description
13174 \glSsetnoexpandfield{desc}%
13175 \glSsetnoexpandfield{descplural}%

```

```

13176 \else
13177   \glsssetexpandfield{desc}%
13178   \glsssetexpandfield{descplural}%
13179 \fi
13180 }
13181 \define@boolkey[glS]{sanitize}{name}[true]{%
13182   \GlossariesWarning{sanitize={name} package option deprecated}%
13183   \ifglS@sanitize@name
13184     \glsssetnoexpandfield{name}%
13185   \else
13186     \glsssetexpandfield{name}%
13187   \fi
13188 }
13189 \define@boolkey[glS]{sanitize}{symbol}[true]{%
13190   \GlossariesWarning{sanitize={symbol} package option deprecated}%
13191   \ifglS@sanitize@symbol
13192     \glsssetnoexpandfield{symbol}%
13193     \glsssetnoexpandfield{symbolplural}%
13194   \else
13195     \glsssetexpandfield{symbol}%
13196     \glsssetexpandfield{symbolplural}%
13197   \fi
13198 }
13199 \define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
13200   \ifglsssanitizesort
13201     \glsssetnoexpandfield{sortvalue}%
13202     \renewcommand*{\@glS@noidx@setsanitizesort}{%
13203       \glsssanitizesorttrue
13204       \glsssetnoexpandfield{sortvalue}%
13205     }%
13206   \else
13207     \glsssetexpandfield{sortvalue}%
13208     \renewcommand*{\@glS@noidx@setsanitizesort}{%
13209       \glsssanitizesortfalse
13210       \glsssetexpandfield{sortvalue}%
13211     }%
13212   \fi
13213 }
13214 \glsssanitizesorttrue
13215 \glsssetnoexpandfield{sortvalue}%
13216 \newcommand*{\@glS@noidx@setsanitizesort}{%
13217   \glsssanitizesortfalse
13218   \glsssetexpandfield{sortvalue}%
13219 }
13220 \define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
13221   \setbool{glsssanitizesort}{#1}%
13222   \ifglsssanitizesort
13223     \glsssetnoexpandfield{sortvalue}%
13224   \else

```



```

13225 \glssetexpandfield{sortvalue}%
13226 \fi
13227 \GlossariesWarning{sanitize={sort} package option
13228 deprecated. Use sanitizesort instead}%
13229 }
13230 \define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
13231 \ifthenelse{\equal{#1}{none}}{%
13232 {%
13233 \GlossariesWarning{sanitize package option deprecated}%
13234 \glssetexpandfield{name}%
13235 \glssetexpandfield{symbol}%
13236 \glssetexpandfield{symbolplural}%
13237 \glssetexpandfield{desc}%
13238 \glssetexpandfield{descplural}%
13239 }%
13240 {%
13241 \setkeys{gls}{sanitize}{#1}%
13242 }%
13243 }
13244 \newif\ifglstranslate
13245 \newcommand*\@gls@usetranslator{%
13246 \@ifpackageloaded{polyglossia}%
13247 {%
13248 \let\glsifusetranslator\@secondoftwo
13249 }%
13250 {%
13251 \@ifpackageloaded{babel}%
13252 {%
13253 \IfFileExists{translator.sty}%
13254 {%
13255 \RequirePackage{translator}%
13256 \let\glsifusetranslator\@firstoftwo
13257 }%
13258 {}%
13259 }%
13260 {}%
13261 }%
13262 }
13263 \newcommand{\glsifusedtranslatordict}[3]{%
13264 \glsifusetranslator
13265 {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
13266 {#3}%
13267 }
13268 \@gls@declareoption{nottranslate}{%
13269 \glstranslatefalse
13270 \let\@gls@usetranslator\relax
13271 \let\glsifusetranslator\@secondoftwo
13272 }
13273 \define@choicekey{glossaries.sty}{translate}%

```

```

13274 [\gls@translate@val\gls@translate@nr]%
13275 {true,false,babel}[true]%
13276 {%
13277   \ifcase\gls@translate@nr\relax
13278     \glstranslatetrue
13279     \renewcommand*\@gls@usetranslator{%
13280       \@ifpackageloaded{polyglossia}%
13281       {%
13282         \let\glsifusetranslator\@secondoftwo
13283       }%
13284       {%
13285         \@ifpackageloaded{babel}%
13286         {%
13287           \IfFileExists{translator.sty}%
13288           {%
13289             \RequirePackage{translator}%
13290             \let\glsifusetranslator\@firstoftwo
13291           }%
13292           {}%
13293         }%
13294         {}%
13295       }%
13296     }%
13297   \or
13298     \glstranslatefalse
13299     \let\@gls@usetranslator\relax
13300     \let\glsifusetranslator\@secondoftwo
13301   \or
13302     \glstranslatetrue
13303     \let\@gls@usetranslator\relax
13304     \let\glsifusetranslator\@secondoftwo
13305   \fi
13306 }
13307 \glstranslatefalse
13308 \let\glsifusetranslator\@secondoftwo
13309 \@ifpackageloaded{translator}%
13310 {%
13311   \glstranslatetrue
13312   \let\glsifusetranslator\@firstoftwo
13313 }%
13314 {%
13315   \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
13316   {
13317     \@ifpackageloaded{\gls@thissty}%
13318     {%
13319       \glstranslatetrue
13320       \@endfortrue
13321     }%
13322     {}%

```

```

13323 }
13324 }
13325 \define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
13326 \glsindexonlyfirstfalse
13327 \define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
13328 \glshyperfirsttrue
13329 \newcommand*{\@gls@setacrstyle}{}
13330 \define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
13331 \ifbool{glsacrdescription}%
13332 {}%
13333 {%
13334 \renewcommand*{\@gls@sanitizedesc}{}%
13335 }%
13336 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13337 }
13338 \define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
13339 \renewcommand*{\@gls@sanitizesymbol}{}%
13340 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13341 }
13342 \define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
13343 \renewcommand*{\@gls@sanitizesymbol}{}%
13344 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13345 }
13346 \define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
13347 \renewcommand*{\@gls@sanitizesymbol}{}%
13348 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13349 }
13350 \define@boolkey{glossaries.sty}[glsacr]{dua}[true]{}
13351 \renewcommand*{\@gls@sanitizesymbol}{}%
13352 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13353 }
13354 \define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
13355 \newcommand*{\glsorder}{word}
13356 \newcommand*{\@glsorder}[1]{}
13357 \define@choicekey{glossaries.sty}{order}{word,letter}{}
13358 \def\glsorder{#1}}
13359 \newif\ifglxindy
13360 \glxindyfalse
13361 \@gls@declareoption{makeindex}{\glxindyfalse}
13362 \define@boolkey[gls]{xindy}{glsnumbers}[true]{}
13363 \gls@xindy@glsnumberstrue
13364 \def\@xdy@main@language{\language}%
13365 \define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
13366 \ifcsundef{inputencodingname}{}%
13367 \def\gls@codepage{}{}%
13368 \def\gls@codepage{\inputencodingname}
13369 }
13370 \define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
13371 \define@key{glossaries.sty}{xindy}[]{}

```

```

13372 \glxindytrue
13373 \setkeys[glx]{indy}{#1}%
13374 }
13375 \@glx@declareoption{xindygloss}{%
13376 \glxindytrue
13377 }
13378 \@glx@declareoption{xindynoglsnumbers}{%
13379 \glxindytrue
13380 \glx@xindy@glxnumbersfalse
13381 }
13382 \providecommand{\@domakeglossaries}[1]{#1}
13383 \@glx@declareoption{disablemakegloss}{%
13384 \ifdefequal\makeglossaries\@no@makeglossaries
13385 {%
13386 \GlossariesWarning{Option ‘disablemakegloss’ has no effect
13387 (\string\makenoidxglossaries\space has already been used)}%
13388 }%
13389 {%
13390 \ifdefequal\@makeglossary\@gobble
13391 {%
13392 \GlossariesWarning{Option ‘disablemakegloss’ has no effect
13393 (\string\makeglossaries\space has already been used)}%
13394 }%
13395 {%
13396 \renewcommand{\@domakeglossaries}[1]{%
13397 \PackageInfo{glossaries}{\string\makeglossaries\space and
13398 \string\makenoidxglossaries\space have been disabled}%
13399 }%
13400 }%
13401 }%
13402 }
13403 \@glx@declareoption{restoremakegloss}{%
13404 \ifdefequal\makeglossaries\@no@makeglossaries
13405 {%
13406 \GlossariesWarning{Option ‘restoremakegloss’ has no effect
13407 (\string\makenoidxglossaries\space has already been used)}%
13408 }%
13409 {%
13410 \ifdefequal\@makeglossary\@gobble
13411 {%
13412 \GlossariesWarning{Option ‘restoremakegloss’ has no effect
13413 (\string\makeglossaries\space has already been used)}%
13414 }%
13415 {%
13416 \PackageInfo{glossaries}{\string\makeglossaries\space and
13417 \string\makenoidxglossaries\space have been restored}%
13418 \let\@domakeglossaries\@firstofone
13419 }%
13420 }%

```

```

13421 }
13422 \newcommand*{\@do@write@glslabels}{%
13423   \AtEndDocument{\@do@write@glslabels}%
13424   \let\@do@write@glslabels\relax
13425 }
13426 \newcommand*{\@do@do@write@glslabels}{%
13427   \newwrite\@glsl@labelsfile
13428   \immediate\openout\@glsl@labelsfile=\jobname.glslabels
13429   \forallglstentries[\@glo@types,\@ignored@glossaries]{\@glstentry}%
13430   {\ifdefempty{\@glstentry}{\immediate\write\@glsl@labelsfile{\@glstentry}}}%
13431   \immediate\closeout\@glsl@labelsfile
13432 }
13433 \@glsl@declareoption{writeglslabels}{\@do@write@glslabels}
13434 \newif\ifglslautomake
13435 \newcommand{\glsl@automake@nr}{1}
13436 \define@choicekey{glossaries.sty}{automake}%
13437   [\glsl@automake@val\glsl@automake@nr]{true,false,immediate}[true]{%
13438   \ifnum\glsl@automake@nr=1\relax
13439     \glslautomakefalse
13440   \else
13441     \glslautomaketrue
13442   \fi
13443   \ifglslautomake
13444     \renewcommand*{\@glsl@doautomake}{%
13445       \PackageError{glossaries}{You must use
13446       \string\makeglossaries\space with automake=true}
13447       {%
13448         Either remove the automake=true setting or
13449         add \string\makeglossaries\space to your document preamble.%
13450       }%
13451     }%
13452   \else
13453     \renewcommand*{\@glsl@doautomake}{}%
13454   \fi
13455 }
13456 \glslautomakefalse
13457 \newcommand*{\@glsl@doautomake}{}
13458 \AtEndDocument{\@glsl@doautomake}
13459 \define@boolkey{glossaries.sty}[glsl]{savewrites}[true]{%
13460   \ifglssavewrites
13461     \renewcommand*{\glslwritefiles}{\@glslwritefiles}%
13462   \else
13463     \let\glslwritefiles\@empty
13464   \fi
13465 }
13466 \glssavewritesfalse
13467 \let\glslwritefiles\@empty
13468 \define@boolkey{glossaries.sty}[glsl]{compatible-3.07}[true]{}
13469 \boolfalse{glslcompatible-3.07}

```

```

13470 \define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
13471   \ifbool{glscompatible-2.07}%
13472   {%
13473     \booltrue{glscompatible-3.07}%
13474   }%
13475   {}%
13476 }
13477 \boolfalse{glscompatible-2.07}
13478 \let\gls@original@makeglossary\makeglossary
13479 \let\gls@original@glossary\glossary
13480 \def\makeglossary{%
13481   \GlossariesWarning{Use of \string\makeglossary\space with
13482     glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
13483     instead. If you \MessageBreak need the original definition of
13484     \string\makeglossary\space use \MessageBreak the package options
13485     kernelglossredefs=false (to \MessageBreak restore the former definition of
13486     \string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
13487     conflict)}}%
13488   \makeglossaries
13489 }
13490 \newcommand*{\@gls@override@glossary}[1][main]{%
13491   \GlossariesWarning{Use of \string\glossary\space with
13492     glossaries.sty is deprecated. \MessageBreak Indexing should be performed
13493     with the user level \MessageBreak commands, such as \string\gls\space or
13494     \string\glsadd. If you need the \MessageBreak original definition of
13495     \string\glossary\space use the package \MessageBreak options
13496     kernelglossredefs=false (to restore the \MessageBreak former definition of
13497     \string\glossary) and nomain (if the \MessageBreak file extensions cause a
13498     conflict)}}%
13499   \gls@glossary{#1}%
13500 }
13501 \if@gls@docloaded
13502 \else
13503   \def\glossary{\@gls@override@glossary}
13504 \fi
13505 \define@choicekey{glossaries.sty}{kernelglossredefs}%
13506   [\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
13507 {%
13508   \ifcase\gls@debug@nr\relax
13509     \def\glossary{\@gls@override@glossary}%
13510     \def\makeglossary{%
13511       \GlossariesWarning{Use of \string\makeglossary\space with
13512         glossaries.sty is deprecated. Use \string\makeglossaries\space
13513         instead. If you need the original definition of
13514         \string\makeglossary\space use the package options
13515         kernelglossredefs=false (to prevent redefinition of
13516         \string\makeglossary) and nomain (if the file extensions cause a
13517         conflict)}}%
13518       \makeglossaries

```

```

13519 }%
13520 \or
13521 \let\glossary\gls@original@glossary
13522 \let\makeglossary\gls@original@makeglossary
13523 \or
13524 \def\makeglossary{\makeglossaries}%
13525 \renewcommand*{\@gls@override@glossary}[1][main]{%
13526 \gls@glossary{##1}%
13527 }%
13528 \fi
13529 }
13530 \@gls@declareoption{symbols}{%
13531 \let\@gls@do@symbolsdef\@gls@symbolsdef
13532 }
13533 \newcommand*{\@gls@do@symbolsdef}{%
13534 \newcommand*{\@gls@symbolsdef}{%
13535 \newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
13536 \newcommand*{\printsymbols}[1][ ]{\printglossary[type=symbols,##1]}%
13537 \newcommand*{\gls@tr@set@symbols@toctitle}{%
13538 \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
13539 }%
13540 }%
13541 \@gls@declareoption{numbers}{%
13542 \let\@gls@do@numbersdef\@gls@numbersdef
13543 }
13544 \newcommand*{\@gls@do@numbersdef}{%
13545 \newcommand*{\@gls@numbersdef}{%
13546 \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
13547 \newcommand*{\printnumbers}[1][ ]{\printglossary[type=numbers,##1]}%
13548 \newcommand*{\gls@tr@set@numbers@toctitle}{%
13549 \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
13550 }%
13551 }%
13552 \@gls@declareoption{index}{%
13553 \ifx\@gls@do@indexdef\@empty
13554 \let\@gls@do@indexdef\@gls@indexdef
13555 \fi
13556 }
13557 \@gls@declareoption{noglossaryindex}{%
13558 \let\@gls@do@indexdef\relax
13559 }
13560 \newcommand*{\@gls@do@indexdef}{%
13561 \newcommand*{\@gls@indexdef}{%
13562 \newglossary[ilg]{index}{ind}{idx}{\indexname}%
13563 \newcommand*{\printindex}[1][ ]{\printglossary[type=index,##1]}%
13564 \newcommand*{\newterm}[2][ ]{%
13565 \newglossaryentry{##2}%
13566 {type={index},name={##2},description={\nopostdesc},##1}}
13567 \let\@gls@do@indexdef\relax

```

```

13568 }%
13569 \@for\CurrentOption := \@declaredoptions\do{%
13570   \ifx\CurrentOption\@empty
13571   \else
13572     \@expandtwoargs
13573     \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
13574     \ifin@
13575     \@use@ption
13576     \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
13577   \fi
13578 \fi
13579 }
13580 \ProcessOptionsX
13581 \RequirePackage{glossaries-compatible-307}
13582 \disable@keys{glossaries.sty}{compatible-2.07,%
13583 xindy,xindygloss,xindynoglsnumbers,makeindex,%
13584 acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
13585 nomain,noglossaryindex}
13586 \newcommand*{\setupglossaries}[1]{%
13587   \renewcommand*{@gls@setacrstyle}{}%
13588   \ifglsacrshortcuts
13589     \def@gls@setupshortcuts{glsacrshortcutstrue}%
13590   \else
13591     \def@gls@setupshortcuts{%
13592       \ifglsacrshortcuts
13593         \DefineAcronymSynonyms
13594       \fi
13595     }%
13596   \fi
13597   \glsacrshortcutsfalse
13598   \let@gls@do@numbersdef\relax
13599   \let@gls@do@symbolssdef\relax
13600   \let@gls@do@indexdef\relax
13601   \let@gls@do@acronymsdef\relax
13602   \ifgl Sentrycounter
13603     \let@gls@doentrycounterdef\relax
13604   \else
13605     \let@gls@doentrycounterdef@gls@define@glossaryentrycounter
13606   \fi
13607   \ifglssubentrycounter
13608     \let@gls@dosubentrycounterdef\relax
13609   \else
13610     \let@gls@dosubentrycounterdef@gls@define@glossarysubentrycounter
13611   \fi
13612   \setkeys{glossaries.sty}{#1}%
13613   \@gls@setacrstyle
13614   \@gls@setupshortcuts
13615   \@gls@do@acronymsdef
13616   \@gls@do@numbersdef

```



```

13617 \@gls@do@symbolssdef
13618 \@gls@do@indexdef
13619 \@gls@doentrycounterdef
13620 \@gls@dosubentrycounterdef
13621 }
13622 \ifthenelse{\equal{\@glscounter}{section}}{%
13623 {%
13624 \ifcsundef{chapter}{}%
13625 {%
13626 \let\@gls@old@chapter\@chapter
13627 \def\@chapter[#1]#2{\@gls@old@chapter[#1]{#2}%
13628 \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}}%
13629 }%
13630 }%
13631 {}
13632 \newcommand*\@gls@onlypremakeg{}
13633 \newcommand*\@onlypremakeg[1]{%
13634 \ifx\@gls@onlypremakeg\@empty
13635 \def\@gls@onlypremakeg{#1}%
13636 \else
13637 \expandafter\toks\expandafter{\@gls@onlypremakeg}%
13638 \edef\@gls@onlypremakeg{\the\toks,\noexpand#1}%
13639 \fi
13640 }
13641 \newcommand*\@disable@onlypremakeg{%
13642 \@for\@thiscs:=\@gls@onlypremakeg\do{%
13643 \expandafter\@disable@premakecs\@thiscs%
13644 }}
13645 \newcommand*\@disable@premakecs[1]{%
13646 \def#1{\PackageError{glossaries}{\string#1\space may only be
13647 used before \string\makeglossaries}{You can't use
13648 \string#1\space after \string\makeglossaries}}%
13649 }
13650 \providecommand*\glossaryname{Glossary}
13651 \providecommand*\acronymname{Acronyms}
13652 \newcommand*\glssettocitle[1]{%
13653 \def\glossarytocitle{\csname @glotype@#1@title\endcsname}}
13654 \providecommand*\entryname{Notation}
13655 \providecommand*\descriptionname{Description}
13656 \providecommand*\symbolname{Symbol}
13657 \providecommand*\pagelistname{Page List}
13658 \providecommand*\glsymbolsgroupname{Symbols}
13659 \providecommand*\glsnumbersgroupname{Numbers}
13660 \newcommand*\glspluralsuffix{s}
13661 \newcommand*\glsacrpluralsuffix{\glspluralsuffix}
13662 \newcommand*\glsupacrpluralsuffix{\glsacrpluralsuffix}
13663 \providecommand*\seename{see}
13664 \providecommand*\andname{&}
13665 \newcommand*\RequireGlossariesLang[1]{%

```

```

13666 \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
13667 }
13668 \newcommand*{\ProvidesGlossariesLang}[1]{%
13669   \ProvidesFile{glossaries-#1.ldf}%
13670 }
13671 \newcommand*{\addglossarytocaptions}[1]{%
13672   \ifglstranslate
13673     \RequirePackage{tracklang}
13674     \@gls@usetranslator
13675     \ifpackageloaded{translator}
13676     {%
13677       \ifboolexpr
13678       {
13679         test {\ifdefstring{\trans@languages}{English}}
13680         and not
13681         test {\ifdefstring{bbl@loaded}{english}}
13682       }
13683       {%
13684         \let\glsifusetranslator\@secondoftwo
13685       }%
13686       {%
13687         \usedictionary{glossaries-dictionary}%
13688         \renewcommand*{\addglossarytocaptions}[1]{%
13689           \ifcsundef{captions#1}{}%
13690           {%
13691             \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
13692             \expandafter\toks@\expandafter{\@gls@tmp
13693               \renewcommand*{\glossaryname}{\translate{Glossary}}}%
13694             }%
13695             \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
13696           }%
13697         }%
13698       }%
13699     }%
13700   }{}%
13701   \AnyTrackedLanguages
13702   {%
13703     \ForEachTrackedDialect{\this@dialect}{%
13704       \IfTrackedLanguageFileExists{\this@dialect}%
13705       {glossaries-}% prefix
13706       {.ldf}%
13707       {%
13708         \RequireGlossariesLang{\CurrentTrackedTag}%
13709       }%
13710     }%
13711     \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
13712   }%
13713 }%
13714 }%

```

```

13715 {}%
13716 \glsifusetranslator
13717 {%
13718   \renewcommand*{\glssettoctitle}[1]{%
13719     \ifcsdef{gls@tr@set@#1@toctitle}%
13720     {%
13721       \csuse{gls@tr@set@#1@toctitle}%
13722     }%
13723     {%
13724       \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
13725     }%
13726   }%
13727   \renewcommand*{\glossaryname}{\translate{Glossary}}%
13728   \renewcommand*{\acronymname}{\translate{Acronyms}}%
13729   \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
13730   \renewcommand*{\descriptionname}{%
13731     \translate{Description (glossaries)}}%
13732   \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
13733   \renewcommand*{\pagelistname}{%
13734     \translate{Page List (glossaries)}}%
13735   \renewcommand*{\glssymbolsgroupname}{%
13736     \translate{Symbols (glossaries)}}%
13737   \renewcommand*{\glsnumbersgroupname}{%
13738     \translate{Numbers (glossaries)}}%
13739   }{}%
13740 \fi
13741 \DeclareRobustCommand*{\nopostdesc}{%
13742 \newcommand*{\@nopostdesc}{%
13743   \let\org@glspostdescription\glspostdescription
13744   \def\glspostdescription{%
13745     \let\glspostdescription\org@glspostdescription}%
13746 }
13747 \newcommand*{\@no@post@desc}{\nopostdesc}
13748 \newcommand{\glspar}{\par}
13749 \newcommand{\setStyleFile}[1]{%
13750   \renewcommand*{\gls@istfilebase}{#1}%
13751   \ifglsxindy
13752     \def\istfilename{\gls@istfilebase.xdy}
13753   \else
13754     \def\istfilename{\gls@istfilebase.ist}
13755   \fi
13756 }
13757 \@onlypremakeg\setStyleFile
13758 \ifglsxindy
13759   \def\istfilename{\gls@istfilebase.xdy}
13760 \else
13761   \def\istfilename{\gls@istfilebase.ist}
13762 \fi
13763 \newcommand*{\gls@istfilebase}{\jobname}

```

```

13764 \newcommand*{\@istfilename}[1]{}
13765 \newcommand*{\glscompositor}{.}
13766 \newcommand*{\glsSetCompositor}[1]{%
13767   \renewcommand*{\glscompositor}{#1}}
13768 \@onlypremakeg\glsSetCompositor
13769 \newcommand*{\@glsAlphacompositor}{\glscompositor}
13770 \ifglxsindy
13771   \newcommand*{\glsSetAlphaCompositor}[1]{%
13772     \renewcommand*{\@glsAlphacompositor}{#1}}
13773 \else
13774   \newcommand*{\glsSetAlphaCompositor}[1]{%
13775     \glsnoxywarning\glsSetAlphaCompositor}
13776 \fi
13777 \@onlypremakeg\glsSetAlphaCompositor
13778 \newcommand*{\gls@suffixF}{}
13779 \newcommand*{\glsSetSuffixF}[1]{%
13780   \renewcommand*{\gls@suffixF}{#1}}
13781 \@onlypremakeg\glsSetSuffixF
13782 \newcommand*{\gls@suffixFF}{}
13783 \newcommand*{\glsSetSuffixFF}[1]{%
13784   \renewcommand*{\gls@suffixFF}{#1}}
13785 }
13786 \ifcsundef{hyperlink}%
13787 {%
13788   \newcommand*{\glsnumberformat}[1]{#1}%
13789 }%
13790 {%
13791   \newcommand*{\glsnumberformat}[1]{\glshypernumber{#1}}%
13792 }
13793 \newcommand{\delimN}{, }
13794 \newcommand{\delimR}{--}
13795 \newcommand*{\glossarypreamble}{%
13796   \csuse{@glossarypreamble@\currentglossary}%
13797 }
13798 \newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
13799   \ifglossaryexists*{#1}%
13800     {\csgdef{@glossarypreamble@#1}{#2}}%
13801     {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
13802 }
13803 \newcommand*{\glossarypostamble}{}
13804 \newcommand*{\glossarysection}[2][\@gls@title]{%
13805   \def\@gls@title{#2}%
13806   \ifcsundef{phantomsection}%
13807     {%
13808       \@glossarysection{#1}{#2}%
13809     }%
13810     {%
13811       \@pglossarysection{#1}{#2}%
13812     }%

```

```

13813 \glsglossarymark{\glossarytoctitle}%
13814 }
13815 \ifcsundef{glossarymark}%
13816 {%
13817 \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
13818 }%
13819 {%
13820 \@ifclassloaded{memoir}
13821 {%
13822 \newcommand{\glsglossarymark}[1]{%
13823 \ifglsucmark
13824 \markboth{\memUHead{#1}}{\memUHead{#1}}%
13825 \else
13826 \markboth{#1}{#1}%
13827 \fi
13828 }
13829 }%
13830 {%
13831 \newcommand{\glsglossarymark}[1]{%
13832 \ifglsucmark
13833 \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
13834 \else
13835 \@mkboth{#1}{#1}%
13836 \fi
13837 }
13838 }
13839 }
13840 \providecommand{\glossarymark}[1]{%
13841 \ifglsucmark
13842 \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
13843 \else
13844 \@mkboth{#1}{#1}%
13845 \fi
13846 }
13847 \newcommand*{\setglossarysection}[1]{%
13848 \setkeys{glossaries.sty}{section=#1}}
13849 \newcommand*{\@glossarysection}[2]{%
13850 \ifdefempty\@glossarysecstar
13851 {%
13852 \csname\@glossarysec\endcsname[#1]{#2}%
13853 }%
13854 {%
13855 \csname\@glossarysec\endcsname*{#2}%
13856 \@gls@toc{#1}{\@glossarysec}%
13857 }%
13858 \@glossaryseclabel
13859 }
13860 \newcommand*{\@p@glossarysection}[2]{%
13861 \glsclearpage

```

```

13862 \phantomsection
13863 \ifdefempty\@@glossarysecstar
13864 {%
13865   \csname\@@glossarysec\endcsname{#2}%
13866 }%
13867 {%
13868   \@gls@toc{#1}{\@@glossarysec}%
13869   \csname\@@glossarysec\endcsname*{#2}%
13870 }%
13871 \@@glossaryseclabel
13872 }
13873 \newcommand*\@gls@docclearpage{%
13874   \ifthenelse{\equal{\@@glossarysec}{chapter}}%
13875   {%
13876     \ifcsundef{cleardoublepage}%
13877     {%
13878       \clearpage
13879     }%
13880     {%
13881       \ifcsdef{if@openright}%
13882       {%
13883         \if@openright
13884           \cleardoublepage
13885         \else
13886           \clearpage
13887         \fi
13888       }%
13889       {%
13890         \cleardoublepage
13891       }%
13892     }%
13893   }%
13894   {}%
13895 }
13896 \newcommand*\@gls@clearpage{\@gls@docclearpage}
13897 \newcommand*\@gls@toc[2]{%
13898   \ifglstoc
13899     \ifglslnumberline
13900       \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
13901     \else
13902       \addcontentsline{toc}{#2}{#1}%
13903     \fi
13904   \fi
13905 }
13906 \newcommand*\@glsnoxindywarning[1]{%
13907   \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
13908 }
13909 \newcommand*\@glsnomakeindexwarning[1]{%
13910   \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%

```

```

13911 }
13912 \ifglxsindy
13913   \edef\@xdyattributes{\string"default\string"}%
13914 \fi
13915 \ifglxsindy
13916   \edef\@xdyattributelist{}%
13917 \fi
13918 \ifglxsindy
13919   \def\@xdylocref{}
13920 \fi
13921 \newcommand*{\@gls@ifinlist}[4]{%
13922   \def\@do@ifinlist##1,##2\end@ifinlist{%
13923     \def\@gls@listsuffix{##2}%
13924     \ifx\@gls@listsuffix\@empty
13925       #4%
13926     \else
13927       #3%
13928     \fi
13929   }%
13930   \@do@ifinlist,##1,##2\end@ifinlist
13931 }
13932 \ifglxsindy
13933   \newcommand*{\@xdycounters}{\glscounter}
13934   \newcommand*\GlsAddXdyCounters[1]{%
13935     \@for\@gls@ctr:=#1\do{%
13936       \edef\@do@addcounter{%
13937         \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
13938         {%
13939           \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
13940             \noexpand\@gls@ctr}%
13941         }%
13942       }%
13943       \@do@addcounter
13944     }
13945   }
13946   \@onlypremakeg\GlsAddXdyCounters
13947 \else
13948   \newcommand*\GlsAddXdyCounters[1]{%
13949     \glsnoxywarning\GlsAddXdyAttribute
13950   }
13951 \fi
13952 \newcommand*\@disabled@glssaddxdycounters{%
13953   \PackageError{glossaries}{\string\GlsAddXdyCounters\space
13954     can't be used after \string\GlsAddXdyAttribute}{Move all
13955     occurrences of \string\GlsAddXdyCounters\space before the first
13956     instance of \string\GlsAddXdyAttribute}%
13957 }
13958 \ifglxsindy
13959   \newcommand*\@glssaddxdyattribute[2]{%

```

```

13960 \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
13961 \string"#2#1\string"}%
13962 \expandafter\toks@\expandafter{\@xdylocref}%
13963 \edef\@xdylocref{\the\toks@ ^^J%
13964 (markup-locref
13965 :open \string"\glstildechar n%
13966 \expandafter\string\csname glsX#2X#1\endcsname
13967 \string" ^^J
13968 :close \string"\string" ^^J
13969 :attr \string"#2#1\string")}%
13970 \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
13971 \setentrycounter[##1][##2]\csname #1\endcsname{##2}%
13972 }%
13973 }
13974 \newcommand*\GlsAddXdyAttribute[1]{%
13975 \ifx\@xdyattributelist\@empty
13976 \edef\@xdyattributelist{#1}%
13977 \else
13978 \edef\@xdyattributelist{\@xdyattributelist,#1}%
13979 \fi
13980 \@for\@this@counter:=\@xdycounters\do{%
13981 \protected@edef\gls@do@addxdyattribute{%
13982 \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
13983 }
13984 \gls@do@addxdyattribute
13985 }%
13986 \let\GlsAddXdyCounters\@disabled@glsaddxdycounters
13987 }
13988 \@onlypremakeg\GlsAddXdyAttribute
13989 \else
13990 \newcommand*\GlsAddXdyAttribute[1]{%
13991 \glsnoxywarning\GlsAddXdyAttribute}
13992 \fi
13993 \ifglsxindy
13994 \newcommand*{\@gls@addpredefinedattributes}{%
13995 \GlsAddXdyAttribute{glsnumberformat}
13996 \GlsAddXdyAttribute{textrm}
13997 \GlsAddXdyAttribute{textsf}
13998 \GlsAddXdyAttribute{texttt}
13999 \GlsAddXdyAttribute{textbf}
14000 \GlsAddXdyAttribute{textmd}
14001 \GlsAddXdyAttribute{textit}
14002 \GlsAddXdyAttribute{textup}
14003 \GlsAddXdyAttribute{textsl}
14004 \GlsAddXdyAttribute{textsc}
14005 \GlsAddXdyAttribute{emph}
14006 \GlsAddXdyAttribute{glshypernumber}
14007 \GlsAddXdyAttribute{hyperrm}
14008 \GlsAddXdyAttribute{hypersf}

```



```

14009 \GlsAddXdyAttribute{hypertt}
14010 \GlsAddXdyAttribute{hyperbf}
14011 \GlsAddXdyAttribute{hypermd}
14012 \GlsAddXdyAttribute{hyperit}
14013 \GlsAddXdyAttribute{hyperup}
14014 \GlsAddXdyAttribute{hypersl}
14015 \GlsAddXdyAttribute{hypersc}
14016 \GlsAddXdyAttribute{hyperemph}
14017 \GlsAddXdyAttribute{glsignore}
14018 }
14019 \else
14020 \let\@gls@addpredefinedattributes\relax
14021 \fi
14022 \def\@xdyuseralphabets{}
14023 \ifglsxindy
14024 \newcommand*{\GlsAddXdyAlphabet}[2]{%
14025 \edef\@xdyuseralphabets{%
14026 \@xdyuseralphabets ^^J
14027 (define-alphabet "#1" (#2))}}
14028 \else
14029 \newcommand*{\GlsAddXdyAlphabet}[2]{%
14030 \glsnoxindywarning\GlsAddXdyAlphabet}
14031 \fi
14032 \ifglsxindy
14033 \newcommand*{\@gls@xdy@locationlist}{%
14034 roman-page-numbers,%
14035 Roman-page-numbers,%
14036 arabic-page-numbers,%
14037 alpha-page-numbers,%
14038 Alpha-page-numbers,%
14039 Appendix-page-numbers,%
14040 arabic-section-numbers%
14041 }
14042 \protected@edef\@gls@roman{\@roman{0}\string"
14043 \string"roman-numbers-lowercase\string" :sep \string"}}%
14044 \@onelevel@sanitize\@gls@roman
14045 \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
14046 :sep \string"}%
14047 \@onelevel@sanitize\@tmp
14048 \ifx\@tmp\@gls@roman
14049 \expandafter
14050 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
14051 \string"roman-numbers-lowercase\string"%
14052 }%
14053 \else
14054 \expandafter
14055 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
14056 :sep \string"\@gls@roman\string"%
14057 }%

```

```

14058 \fi
14059 \expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
14060   \string"roman-numbers-uppercase\string"%
14061 }%
14062 \expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
14063   \string"arabic-numbers\string"%
14064 }%
14065 \expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
14066   \string"alpha\string"%
14067 }%
14068 \expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
14069   \string"ALPHA\string"%
14070 }%
14071 \expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
14072   \string"ALPHA\string"
14073   :sep \string"@glsAlphacompositor\string"
14074   \string"arabic-numbers\string"%
14075 }
14076 \expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
14077   \string"arabic-numbers\string"
14078   :sep \string"glscocompositor\string"
14079   \string"arabic-numbers\string"%
14080 }%
14081 \def\@xdyuserlocationdefs{}
14082 \def\@xdyuserlocationnames{}
14083 \fi
14084 \ifglsxindy
14085 \newcommand\@xdycrossrefhook{}
14086 \fi
14087 \ifglsxindy
14088 \newcommand*\GlsAddXdyLocation}[3][[]]{%
14089   \def\@gls@tmp{#1}%
14090   \ifx\@gls@tmp\@empty
14091     \edef\@xdyuserlocationdefs{%
14092       \@xdyuserlocationdefs ^^J%
14093       (define-location-class \string"#2\string"^^J\space\space
14094       \space(:sep \string"{}\glssopenbrace\string" #3
14095       :sep \string"\glsclosebrace\string"))
14096   }%
14097   \else
14098     \edef\@xdyuserlocationdefs{%
14099       \@xdyuserlocationdefs ^^J%
14100       (define-location-class \string"#2\string"^^J\space\space
14101       \space(:sep "\glssopenbrace"
14102       #1
14103       :sep "\glsclosebrace\glssopenbrace" #3
14104       :sep "\glsclosebrace"))
14105   }%
14106 \fi

```

```

14107     \edef\@xdyuserlocationnames{%
14108         \@xdyuserlocationnames^^J\space\space\space
14109         \string"#2\string"}%
14110     }
14111     \@onlypremakeg\GlsAddXdyLocation
14112 \else
14113     \newcommand*\GlsAddXdyLocation[2]{%
14114         \glsnnoxindywarning\GlsAddXdyLocation}
14115 \fi
14116 \ifglxindy
14117     \def\@xdylocationclassorder{^^J\space\space\space
14118         \string"roman-page-numbers\string"^^J\space\space\space
14119         \string"arabic-page-numbers\string"^^J\space\space\space
14120         \string"arabic-section-numbers\string"^^J\space\space\space
14121         \string"alpha-page-numbers\string"^^J\space\space\space
14122         \string"Roman-page-numbers\string"^^J\space\space\space
14123         \string"Alpha-page-numbers\string"^^J\space\space\space
14124         \string"Appendix-page-numbers\string"
14125         \@xdyuserlocationnames^^J\space\space\space
14126         \string"see\string"
14127     }
14128 \fi
14129 \ifglxindy
14130     \newcommand*\GlsSetXdyLocationClassOrder[1]{%
14131         \def\@xdylocationclassorder{#1}}
14132 \else
14133     \newcommand*\GlsSetXdyLocationClassOrder[1]{%
14134         \glsnnoxindywarning\GlsSetXdyLocationClassOrder}
14135 \fi
14136 \ifglxindy
14137     \def\@xdysortrules{}
14138 \fi
14139 \ifglxindy
14140     \newcommand*\GlsAddSortRule[2]{%
14141         \expandafter\toks@\expandafter{\@xdysortrules}%
14142         \protected@edef\@xdysortrules{\the\toks@ ^^J
14143             (sort-rule \string"#1\string" \string"#2\string")}%
14144     }
14145 \else
14146     \newcommand*\GlsAddSortRule[2]{%
14147         \glsnnoxindywarning\GlsAddSortRule}
14148 \fi
14149 \ifglxindy
14150     \def\@xdyrequiredstyles{tex}
14151 \fi
14152 \ifglxindy
14153     \newcommand*\GlsAddXdyStyle[1]{%
14154         \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
14155 \else

```

```

14156 \newcommand*\GlsAddXdyStyle[1]{%
14157   \glsnoxywarning\GlsAddXdyStyle}
14158 \fi
14159 \ifglxindy
14160   \newcommand*\GlsSetXdyStyles[1]{%
14161     \edef\@xdyrequiredstyles{#1}}
14162 \else
14163   \newcommand*\GlsSetXdyStyles[1]{%
14164     \glsnoxywarning\GlsSetXdyStyles}
14165 \fi
14166 \newcommand*\findrootlanguage{}
14167 \def\@xdy@language#1#2{}
14168 \ifglxindy
14169   \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
14170     \ifglossaryexists{#1}{%
14171       \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
14172     }{%
14173       \PackageError{glossaries}{Can't set language type for
14174         glossary type '#1' --- no such glossary}{%
14175         You have specified a glossary type that doesn't exist}}
14176 \else
14177   \newcommand*\GlsSetXdyLanguage[2][]{%
14178     \glsnoxywarning\GlsSetXdyLanguage}
14179 \fi
14180 \def\@gls@codepage#1#2{}
14181 \ifglxindy
14182   \newcommand*\GlsSetXdyCodePage[1]{%
14183     \renewcommand*\@gls@codepage{#1}%
14184   }
14185   \AtBeginDocument{%
14186     \ifx\@gls@codepage\@empty
14187       \@ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
14188     \fi
14189   }
14190 \else
14191   \newcommand*\GlsSetXdyCodePage[1]{%
14192     \glsnoxywarning\GlsSetXdyCodePage}
14193 \fi
14194 \ifglxindy
14195   \ifglxindy@glxnumbers
14196     \def\@xdylettergroups{(define-letter-group
14197       \string"glxnumbers\string"^^J\space\space\space
14198       :prefixes (\string"0\string" \string"1\string"
14199       \string"2\string" \string"3\string" \string"4\string"
14200       \string"5\string" \string"6\string" \string"7\string"
14201       \string"8\string" \string"9\string")^^J\space\space\space
14202       \@xdynumbergrouporder)}
14203   \else
14204     \def\@xdylettergroups{}

```

```

14205 \fi
14206 \fi
14207 \newcommand*\GlsAddLetterGroup[2]{%
14208   \expandafter\toks@\expandafter{\@xdylettergroups}%
14209   \protected@edef\@xdylettergroups{\the\toks@^^J%
14210     (define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
14211 }%
14212 \newcommand*\forallglossaries}[3][\@glo@types]{%
14213   \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
14214 }
14215
14216 \newcommand*\forallacronyms}[2]{%
14217   \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
14218 }
14219 \newcommand*\forglsentries}[3][\glsdefaulttype]{%
14220   \edef\@glo@list{\csname glolist@#1\endcsname}%
14221   \@for#2:=\@glo@list\do
14222   {%
14223     \ifdefempty{#2}{#3}%
14224   }%
14225 }
14226 \newcommand*\forallglsentries}[3][\@glo@types]{%
14227   \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
14228   {%
14229     \forglsentries[\@this@glo@]{#2}{#3}%
14230   }%
14231 }
14232 \newcommand{\ifglossaryexists}{%
14233   \@ifstar\s@ifglossaryexists\@ifglossaryexists
14234 }
14235 \newcommand{\@ifglossaryexists}[3]{%
14236   \ifcsundef{@glo@type@#1@out}{#3}{#2}%
14237 }
14238 \newcommand{\s@ifglossaryexists}[3]{%
14239   \ifcsundef{glolist@#1}{#3}{#2}%
14240 }
14241 \newcommand*\glsdetoklabel}[1]{#1}
14242 \newcommand{\ifglsentryexists}[3]{%
14243   \ifcsundef{glo@glsdetoklabel{#1}@name}{#3}{#2}%
14244 }
14245 \newcommand*\ifglsused}[3]{%
14246   \ifbool{glo@glsdetoklabel{#1}@flag}{#2}{#3}%
14247 }
14248 \newcommand{\glsdoifexists}[2]{%
14249   \ifglsentryexists{#1}{#2}{%
14250     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
14251       has not been defined}{You need to define a glossary entry before you
14252       can use it.}}%
14253 }

```

```

14254 \newcommand{\glsdoifnoexists}[2]{%
14255   \ifglentryexists{#1}{%
14256     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
14257       been defined}{-}{#2}%
14258 }
14259 \newcommand{\glsdoifexistsorwarn}[2]{%
14260   \ifglentryexists{#1}{#2}{%
14261     \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
14262       has not been defined}%
14263   }%
14264 }
14265 \newcommand{\glsdoifexistsordo}[3]{%
14266   \ifglentryexists{#1}{#2}{%
14267     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
14268       has not been defined}{You need to define a glossary entry before you
14269       can use it.}%
14270     #3%
14271   }%
14272 }
14273 \newcommand{\doifglossarynoexistsordo}[3]{%
14274   \ifglossaryexists*{#1}%
14275   {%
14276     \PackageError{glossaries}{Glossary type ‘#1’ already exists}{-}%
14277     #3%
14278   }%
14279   {#2}%
14280 }
14281 \newrobustcmd{\ifglshaschildren}[3]{%
14282   \glsdoifexists{#1}%
14283   {%
14284     \def\do@glshaschildren{#3}%
14285     \edef\@gls@thislabel{\glsdetoklabel{#1}}%
14286     \expandafter\forlentries\expandafter
14287       [\csname glo@\@gls@thislabel @type\endcsname]
14288     {\glo@label}%
14289     {%
14290       \letcs\glo@parent{glo@\glo@label @parent}%
14291       \ifdefequal\@gls@thislabel\glo@parent
14292       {%
14293         \def\do@glshaschildren{#2}%
14294         \@endfortrue
14295       }%
14296     }%
14297   }%
14298   \do@glshaschildren
14299 }%
14300 }
14301 \newcommand{\ifglshasparent}[3]{%
14302   \glsdoifexists{#1}%

```

```

14303  {%
14304    \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
14305  }%
14306 }
14307 \newcommand*{\ifglshasdesc}[3]{%
14308   \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
14309   {#3}%
14310   {#2}%
14311 }
14312 \newcommand*{\ifglshasdescsuppressed}[3]{%
14313   \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
14314   {#2}%
14315   {#3}%
14316 }
14317 \newrobustcmd*{\ifglshassymbol}[3]{%
14318   \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
14319   \ifdefempty\@glo@symbol
14320   {#3}%
14321   {%
14322     \ifdefequal\@glo@symbol\@gls@default@value
14323     {#3}%
14324     {#2}%
14325   }%
14326 }
14327 \newrobustcmd*{\ifglshaslong}[3]{%
14328   \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
14329   \ifdefempty\@glo@long
14330   {#3}%
14331   {%
14332     \ifdefequal\@glo@long\@gls@default@value
14333     {#3}%
14334     {#2}%
14335   }%
14336 }
14337 \newrobustcmd*{\ifglshasshort}[3]{%
14338   \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
14339   \ifdefempty\@glo@short
14340   {#3}%
14341   {%
14342     \ifdefequal\@glo@short\@gls@default@value
14343     {#3}%
14344     {#2}%
14345   }%
14346 }
14347 \newrobustcmd*{\ifglshasfield}[4]{%
14348   \glsdoifexists{#2}%
14349   {%
14350     \letcs{\@glo@thisvalue}{glo@\glsdetoklabel{#2}@#1}%
14351     \ifdef\@glo@thisvalue

```

```

14352 {%
14353     \ifdefempty\@glo@thisvalue
14354     {%
14355         #4%
14356     }%
14357     {%
14358         \ifdefequal\@glo@thisvalue\@gls@default@value
14359         {%
14360             #4%
14361         }%
14362         {%
14363             \let\glscurrentfieldvalue\@glo@thisvalue
14364             #3%
14365         }%
14366     }%
14367 }%
14368 {%
14369     \@gls@fetchfield{\@gls@thisfield}{#1}%
14370     \ifdef\@gls@thisfield
14371     {%
14372         \letcs{\@glo@thisvalue}{glo\@glsdetoklabel{#2}\@gls@thisfield}%
14373         \ifdefempty\@glo@thisvalue
14374         {%
14375             #4%
14376         }%
14377         {%
14378             \ifdefequal\@glo@thisvalue\@gls@default@value
14379             {%
14380                 #4%
14381             }%
14382             {%
14383                 \let\glscurrentfieldvalue\@glo@thisvalue
14384                 #3%
14385             }%
14386         }%
14387     }%
14388     {%
14389         \GlossariesWarning{Unknown entry field '#1'}%
14390         #4%
14391     }%
14392 }%
14393 }%
14394 }
14395 \newcommand*{\glscurrentfieldvalue}{}
14396 \newcommand*{\@glo@types}{,}
14397 \newcommand*\@gls@provide@newglossary{%
14398     \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
14399     \let\@gls@provide@newglossary\relax
14400 }

```



```

14401 \newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
14402   \csgdef{gls@#1@entryfmt}{#2}%
14403 }
14404 \newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
14405 \newcommand*{\@gls@forbidtext}[1]{%
14406   \ifboolexpr{test {\ifdefstring{#1}{tex}}
14407     or test {\ifdefstring{#1}{TEX}}}
14408   {%
14409     \def#1{nottex}%
14410     \PackageError{glossaries}%
14411       {Forbidden '.tex' extension replaced with '.nottex'}%
14412       {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
14413         Don't use '.tex' as an extension for a temporary file.}%
14414   }%
14415   {%
14416   }%
14417 }
14418 \newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
14419 \def\@gls@gobbleopt[#1]{}
14420 \newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
14421 \newcommand*{\s@newglossary}[2]{%
14422   \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
14423 }
14424 \newcommand*{\ns@newglossary}[5][glg]{%
14425   \doifglossarynoexistsordo{#2}%
14426   {%
14427     \ifundef\glsdefaulttype
14428     {%
14429       \gdef\glsdefaulttype{#2}%
14430     }{}%
14431     \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
14432     \expandafter\gdef\csname glolist@#2\endcsname{,}%
14433     \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
14434     \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
14435     \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
14436     \expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
14437     \expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
14438     \expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname
14439     \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
14440     \@gls@provide@newglossary
14441     \protected@write\auxout{{\string\@newglossary{#2}{#1}{#3}{#4}}}%
14442     \ifcsundef{gls@#2@entryfmt}%
14443     {%
14444       \defglsentryfmt[#2]{\glsentryfmt}%
14445     }%
14446     {}%
14447     \@gls@defsortcount{#2}%
14448     \@ifnextchar[{\@gls@setcounter{#2}}%
14449     {\@gls@setcounter{#2}[\glscounter]}%

```

```

14450 }%
14451 {%
14452   \gls@gobbleopt
14453 }%
14454 }
14455 \newcommand*{\altnewglossary}[3]{%
14456   \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
14457 }
14458 \@onlypreamble{\newglossary}
14459 \@onlypremakeg\newglossary
14460 \newcommand*{\@newglossary}[4]{%
14461   \def\@gls@setcounter#1[#2]{%
14462     \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
14463     \ifglxindy
14464       \GlsAddXdyCounters{#2}%
14465     \fi
14466   }
14467 \newcommand*{\@gls@getcounter}[1]{%
14468   \csname @glotype@#1@counter\endcsname
14469 }
14470 \glsdefmain
14471 \@gls@do@acronymsdef
14472 \@gls@do@symbolsdef
14473 \@gls@do@numbersdef
14474 \@gls@do@indexdef
14475 \newcommand*{\newignoredglossary}[1]{%
14476   \ifdefempty\@ignored@glossaries
14477   {%
14478     \edef\@ignored@glossaries{#1}%
14479   }%
14480   {%
14481     \eappto\@ignored@glossaries{, #1}%
14482   }%
14483   \csgdef{glolist@#1}{,}%
14484   \ifcsundef{gls@#1@entryfmt}%
14485   {%
14486     \defglsentryfmt[#1]{\glsentryfmt}%
14487   }%
14488   {}%
14489   \ifdefempty\@gls@nohyperlist
14490   {%
14491     \renewcommand*{\@gls@nohyperlist}{#1}%
14492   }%
14493   {%
14494     \eappto\@gls@nohyperlist{, #1}%
14495   }%
14496 }
14497 \newcommand*{\@ignored@glossaries}{}
14498 \newcommand*{\ifignoredglossary}[3]{%

```

```

14499 \edef\@gls@igtype{#1}%
14500 \expandafter\DTLifinlist\expandafter
14501   {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
14502 }
14503 \define@key{glossentry}{name}{%
14504 \def\@glo@name{#1}%
14505 }
14506 \define@key{glossentry}{description}{%
14507 \def\@glo@desc{#1}%
14508 }
14509 \define@key{glossentry}{descriptionplural}{%
14510 \def\@glo@descplural{#1}%
14511 }
14512 \define@key{glossentry}{sort}{%
14513 \def\@glo@sort{#1}}
14514 \define@key{glossentry}{text}{%
14515 \def\@glo@text{#1}%
14516 }
14517 \define@key{glossentry}{plural}{%
14518 \def\@glo@plural{#1}%
14519 }
14520 \define@key{glossentry}{first}{%
14521 \def\@glo@first{#1}%
14522 }
14523 \define@key{glossentry}{firstplural}{%
14524 \def\@glo@firstplural{#1}%
14525 }
14526 \newcommand*{\@gls@default@value}{\relax}
14527 \define@key{glossentry}{symbol}{%
14528 \def\@glo@symbol{#1}%
14529 }
14530 \define@key{glossentry}{symbolplural}{%
14531 \def\@glo@symbolplural{#1}%
14532 }
14533 \define@key{glossentry}{type}{%
14534 \def\@glo@type{#1}}
14535 \define@key{glossentry}{counter}{%
14536 \ifcsundef{c@#1}%
14537   {%
14538     \PackageError{glossaries}%
14539     {There is no counter called ‘#1’}%
14540     {%
14541       The counter key should have the name of a valid counter
14542       as its value%
14543     }%
14544   }%
14545   {%
14546     \def\@glo@counter{#1}%
14547   }%

```

```

14548 }
14549 \define@key{glossentry}{see}{%
14550   \gls@set@xr@key{see}{\@glo@see}{#1}%
14551 }
14552 \newcommand*{\gls@set@xr@key}[3]{%
14553   \renewcommand*{\gls@xr@key}{#1}%
14554   \gls@checkseeallowed
14555   \def#2{#3}%
14556   \@glo@seeautonumberlist
14557 }
14558 \newcommand*{\gls@xr@key}{see}
14559 \newcommand*{\gls@checkseeallowed}{%
14560   \@gls@see@noindex
14561 }
14562 \newcommand*{\gls@checkseeallowed@preambleonly}{%
14563   \GlossariesWarning{glossaries}%
14564   {'\gls@xr@key' key doesn't have any effect when used in the document
14565     environment. Move the definition to the preamble
14566     after \string\makeglossaries\space
14567     or \string\makenoidxglossaries}%
14568 }
14569 \define@key{glossentry}{parent}{%
14570 \def\@glo@parent{#1}}
14571 \define@choicekey{glossentry}{nonumberlist}{%
14572   [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
14573 }%
14574   \ifcase\gls@nonumberlist@nr\relax
14575     \def\@glo@prefix{\glsnonextpages}%
14576     \@gls@savenonumberlist{true}%
14577   \else
14578     \def\@glo@prefix{\glsnextpages}%
14579     \@gls@savenonumberlist{false}%
14580   \fi
14581 }
14582 \newcommand*{\@gls@savenonumberlist}[1]{%
14583 \newcommand*{\@gls@initnonumberlist}{}%
14584 \newcommand*{\@gls@storenonumberlist}[1]{%
14585 \newcommand*{\@gls@enablesavenonumberlist}{%
14586   \renewcommand*{\@gls@initnonumberlist}{%
14587     \undef\@glo@nonumberlist
14588   }%
14589   \renewcommand*{\@gls@savenonumberlist}[1]{%
14590     \def\@glo@nonumberlist{##1}%
14591   }%
14592   \renewcommand*{\@gls@storenonumberlist}[1]{%
14593     \ifdef\@glo@nonumberlist
14594       {%
14595         \cslet{glo@glstetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
14596       }%

```

```

14597     {}%
14598   }%
14599   \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
14600 }
14601 \define@key{glossentry}{user1}{%
14602   \def\@glo@useri{#1}%
14603 }
14604 \define@key{glossentry}{user2}{%
14605   \def\@glo@userii{#1}%
14606 }
14607 \define@key{glossentry}{user3}{%
14608   \def\@glo@useriii{#1}%
14609 }
14610 \define@key{glossentry}{user4}{%
14611   \def\@glo@useriv{#1}%
14612 }
14613 \define@key{glossentry}{user5}{%
14614   \def\@glo@userv{#1}%
14615 }
14616 \define@key{glossentry}{user6}{%
14617   \def\@glo@uservi{#1}%
14618 }
14619 \define@key{glossentry}{short}{%
14620   \def\@glo@short{#1}%
14621 }
14622 \define@key{glossentry}{shortplural}{%
14623   \def\@glo@shortpl{#1}%
14624 }
14625 \define@key{glossentry}{long}{%
14626   \def\@glo@long{#1}%
14627 }
14628 \define@key{glossentry}{longplural}{%
14629   \def\@glo@longpl{#1}%
14630 }
14631 \newcommand*{\@glsnoname}{%
14632   \PackageError{glossaries}{name key required in
14633     \string\newglossaryentry\space for entry '\@glo@label'}{You
14634     haven't specified the entry name}}
14635 \newcommand*{\@glsnodelong}{%
14636   \PackageError{glossaries}
14637     {%
14638       description key required in \string\newglossaryentry\space
14639       for entry '\@glo@label'%
14640     }%
14641   {%
14642     You haven't specified the entry description%
14643   }%
14644 }%
14645 \newcommand*{\@glsdefaultplural}{%

```

```

14646 \newcommand*{\@gls@missingnumberlist}[1]{%
14647   ??%
14648   \ifglssavenumberlist
14649     \GlossariesWarning{Missing number list for entry ‘#1’.
14650     Maybe makeglossaries + rerun required}%
14651   \else
14652     \PackageError{glossaries}%
14653     {Package option ‘savenumberlist=true’ required}%
14654     {%
14655       You must use the ‘savenumberlist’ package option
14656       to reference location lists.%
14657     }%
14658   \fi
14659 }
14660 \newcommand*{\@gls@defaultsort}{\@glo@name}
14661 \newcount\gls@level
14662 \newcommand{\@gls@noexpand@field}[3]{%
14663   \expandafter\global\expandafter
14664   \let\csname glo@#1@#2\endcsname#3%
14665 }
14666 \newcommand{\@gls@noexpand@fields}[4]{%
14667   \ifcsdef{gls@assign@#3@field}
14668   {%
14669     \ifdefequal{#4}{\@gls@default@value}%
14670     {%
14671       \edef\@gls@value{\expandonce{#1}}%
14672       \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
14673     }%
14674     {%
14675       \csuse{gls@assign@#3@field}{#2}{#4}%
14676     }%
14677   }%
14678   {%
14679     \ifdefequal{#4}{\@gls@default@value}%
14680     {%
14681       \edef\@gls@value{\expandonce{#1}}%
14682       \@gls@noexpand@field{#2}{#3}{\@gls@value}%
14683     }%
14684     {%
14685       \@gls@noexpand@field{#2}{#3}{#4}%
14686     }%
14687   }%
14688 }
14689 \newcommand{\@gls@expand@field}[3]{%
14690   \expandafter
14691   \protected@xdef\csname glo@#1@#2\endcsname{#3}%
14692 }
14693 \newcommand{\@gls@expand@fields}[4]{%
14694   \ifcsdef{gls@assign@#3@field}

```

```

14695 {%
14696   \ifdefequal{#4}{\@gls@default@value}%
14697   {%
14698     \edef\@gls@value{\expandonce{#1}}%
14699     \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
14700   }%
14701   {%
14702     \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
14703     {%
14704       \@@gls@expand@field{#2}{#3}{#4}%
14705     }%
14706     {%
14707       \csuse{gls@assign@#3@field}{#2}{#4}%
14708     }%
14709   }%
14710 }%
14711 {%
14712   \ifdefequal{#4}{\@gls@default@value}%
14713   {%
14714     \@@gls@expand@field{#2}{#3}{#1}%
14715   }%
14716   {%
14717     \@@gls@expand@field{#2}{#3}{#4}%
14718   }%
14719 }%
14720 }
14721 \def\@gls@expandonce{\expandonce}
14722 \def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
14723   \def\@gls@tmp{#1}%
14724   \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
14725 }
14726 \let\gls@assign@field\@gls@expand@fields
14727 \newcommand*\@gls@expandfields{%
14728   \let\gls@assign@field\@gls@expand@fields
14729 }
14730 \newcommand*\@gls@noexpandfields{%
14731   \let\gls@assign@field\@gls@noexpand@fields
14732 }
14733 \newrobustcmd{\newglossaryentry}[2]{%
14734   \glsdoifnoexists{#1}%
14735   {%
14736     \gls@defglossaryentry{#1}{#2}%
14737   }%
14738 }
14739 \newcommand*\@gls@defdocnewglossaryentry{%
14740   \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
14741   \let\newglossaryentry\newglossaryentry
14742 }
14743 \newrobustcmd{\provideglossaryentry}[2]{%

```

```

14744 \ifglentryexists{#1}%
14745 {}%
14746 {%
14747   \gls@defglossaryentry{#1}{#2}%
14748 }%
14749 }
14750 \onlypreamble{\provideglossaryentry}
14751 \newrobustcmd{\new@glossaryentry}[2]{%
14752   \ifundef\@gls@deffile
14753   {%
14754     \global\newwrite\@gls@deffile
14755     \immediate\openout\@gls@deffile=\jobname.glsdefs
14756   }%
14757   {}%
14758   \ifglentryexists{#1}{}%
14759   {%
14760     \gls@defglossaryentry{#1}{#2}%
14761   }%
14762   \@gls@writedef{#1}%
14763 }
14764 \AtBeginDocument{\gls@begindocdefs}
14765 \AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
14766 \newcommand*\@gls@begindocdefs{%
14767   \@gls@enablesavenonumberlist
14768   \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
14769   \makeatletter
14770   \InputIfFileExists{\jobname.glsdefs}{\@gls@restoreat}{\@gls@restoreat}
14771   \undef\@gls@restoreat
14772   \gls@defdocnewglossaryentry
14773 }
14774 }
14775 \newcommand*\@gls@writedef[1]{%
14776   \immediate\write\@gls@deffile
14777   {%
14778     \string\ifglentryexists{#1}{}\glspercentchar^^J%
14779     \expandafter\@gobble\string{\glspercentchar^^J%
14780       \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
14781       \expandafter\@gobble\string{\glspercentchar%
14782     }%
14783     \@for\@gls@map:=\@gls@keymap\do
14784     {%
14785       \letcs\glo@value{glo\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
14786       \ifdef\glo@value
14787       {%
14788         \@onelevel@sanitize\glo@value
14789         \immediate\write\@gls@deffile
14790         {%
14791           \expandafter\@firstoftwo\@gls@map
14792           =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%

```



```

14793         \glspercentchar
14794     }%
14795 }%
14796 {}%
14797 }%
14798 \glswritedefhook
14799 \immediate\write\@gls@deffile
14800 {%
14801     \glspercentchar^^J%
14802     \expandafter\@gobble\string\}\glspercentchar^^J%
14803     \expandafter\@gobble\string\}\glspercentchar%
14804 }%
14805 }
14806 \newcommand*{\@gls@keymap}{%
14807     {name}{name},%
14808     {sort}{sortvalue},% unescaped sort value
14809     {type}{type},%
14810     {first}{first},%
14811     {firstplural}{firstpl},%
14812     {text}{text},%
14813     {plural}{plural},%
14814     {description}{desc},%
14815     {descriptionplural}{descplural},%
14816     {symbol}{symbol},%
14817     {symbolplural}{symbolplural},%
14818     {user1}{useri},%
14819     {user2}{userii},%
14820     {user3}{useriii},%
14821     {user4}{useriv},%
14822     {user5}{userv},%
14823     {user6}{uservi},%
14824     {long}{long},%
14825     {longplural}{longpl},%
14826     {short}{short},%
14827     {shortplural}{shortpl},%
14828     {counter}{counter},%
14829     {parent}{parent}%
14830 }
14831 \newcommand*{\@gls@fetchfield}[2]{%
14832     \edef\@gls@thisval{#2}%
14833     \@for\@gls@map:=\@gls@keymap\do{%
14834         \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
14835         \ifdefequal{\@this@key}{\@gls@thisval}%
14836             {%
14837                 \edef#1{\expandafter\@secondoftwo\@gls@map}%
14838                 \@endfortrue
14839             }%
14840         }%
14841     }%

```

```

14842 }
14843 \newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
14844 \newcommand*{\@sglsaddstoragekey}[1]{%
14845   \key@ifundefined{glossentry}{#1}%
14846   {%
14847     \expandafter\newcommand\expandafter*\expandafter
14848       {\csname gls@assign@#1@field\endcsname}[2]{%
14849         \@gls@expand@field{##1}{#1}{##2}%
14850       }%
14851   }%
14852   {}%
14853   \@glsaddstoragekey{#1}%
14854 }
14855 \newcommand*{\@glsaddstoragekey}[3]{%
14856   \key@ifundefined{glossentry}{#1}%
14857   {%
14858     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
14859     \appto\@gls@keymap{,{#1}{#1}}%
14860     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
14861     \appto\@newglossaryentryposthook{%
14862       \letcs{\@glo@tmp}{@glo@#1}%
14863       \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
14864     }%
14865     \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
14866   }%
14867   {%
14868     \PackageError{glossaries}{Key ‘#1’ already exists}{}%
14869   }%
14870 }
14871 \newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
14872 \newcommand*{\@sglsaddkey}[1]{%
14873   \key@ifundefined{glossentry}{#1}%
14874   {%
14875     \expandafter\newcommand\expandafter*\expandafter
14876       {\csname gls@assign@#1@field\endcsname}[2]{%
14877         \@gls@expand@field{##1}{#1}{##2}%
14878       }%
14879     }%
14880     {}%
14881     \@glsaddkey{#1}%
14882 }
14883 \newcommand*{\@glsaddkey}[7]{%
14884   \key@ifundefined{glossentry}{#1}%
14885   {%
14886     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
14887     \appto\@gls@keymap{,{#1}{#1}}%
14888     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
14889     \appto\@newglossaryentryposthook{%
14890       \letcs{\@glo@tmp}{@glo@#1}%

```

```

14891 \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
14892 }%
14893 \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
14894 \newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
14895 \ifcsdef{@gls@user@#1@}%
14896 {%
14897 \PackageError{glossaries}%
14898 {Can't define '\string#5' as helper command
14899 '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
14900 }%
14901 }%
14902 {%
14903 \expandafter\newcommand\expandafter*\expandafter
14904 {\csname @gls@user@#1\endcsname}[2][1]{%
14905 \new@ifnextchar[%
14906 {\csuse{@gls@user@#1@}{##1}{##2}}%
14907 {\csuse{@gls@user@#1@}{##1}{##2}[1]}}%
14908 \csdef{@gls@user@#1@}##1##2[##3]{%
14909 \@gls@field@link{##1}{##2}{#3{##2}##3}%
14910 }%
14911 \newrobustcmd*{#5}{%
14912 \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
14913 }%
14914 \ifcsdef{@Gls@user@#1@}%
14915 {%
14916 \PackageError{glossaries}%
14917 {Can't define '\string#6' as helper command
14918 '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
14919 }%
14920 }%
14921 {%
14922 \expandafter\newcommand\expandafter*\expandafter
14923 {\csname @Gls@user@#1\endcsname}[2][1]{%
14924 \new@ifnextchar[%
14925 {\csuse{@Gls@user@#1@}{##1}{##2}}%
14926 {\csuse{@Gls@user@#1@}{##1}{##2}[1]}}%
14927 \csdef{@Gls@user@#1@}##1##2[##3]{%
14928 \@gls@field@link{##1}{##2}{#4{##2}##3}%
14929 }%
14930 \newrobustcmd*{#6}{%
14931 \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
14932 }%
14933 \ifcsdef{@GLS@user@#1@}%
14934 {%
14935 \PackageError{glossaries}%
14936 {Can't define '\string#7' as helper command
14937 '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
14938 }%
14939 }%

```

```

14940   {%
14941     \expandafter\newcommand\expandafter*\expandafter
14942       {\csname @GLS@user@#1\endcsname}[2][ ]{%
14943         \new@ifnextchar[%
14944           {\csuse{@GLS@user@#1@}{##1}{##2}}}%
14945         {\csuse{@GLS@user@#1@}{##1}{##2}[ ]}}}%
14946     \csdef{@GLS@user@#1@}##1##2[##3]{%
14947       \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}}%
14948   }%
14949   \newrobustcmd*{#7}{%
14950     \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
14951   }%
14952 }%
14953 {%
14954   \PackageError{glossaries}{Key ‘#1’ already exists}{}%
14955 }%
14956 }
14957 \newcommand{\glsfieldxddef}[3]{%
14958   \glsdoifexists{#1}%
14959   {%
14960     \edef\@glo@label{\glsdetoklabel{#1}}%
14961     \ifcsdef{glo@\@glo@label @#2}%
14962     {%
14963       \protected@csxdef{glo@\@glo@label @#2}{#3}%
14964     }%
14965     {%
14966       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
14967     }%
14968   }%
14969 }
14970 \newcommand{\glsfielddedef}[3]{%
14971   \glsdoifexists{#1}%
14972   {%
14973     \edef\@glo@label{\glsdetoklabel{#1}}%
14974     \ifcsdef{glo@\@glo@label @#2}%
14975     {%
14976       \protected@csedef{glo@\@glo@label @#2}{#3}%
14977     }%
14978     {%
14979       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
14980     }%
14981   }%
14982 }
14983 \newcommand{\glsfieldgdef}[3]{%
14984   \glsdoifexists{#1}%
14985   {%
14986     \edef\@glo@label{\glsdetoklabel{#1}}%
14987     \ifcsdef{glo@\@glo@label @#2}%
14988     {%

```

```

14989     \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
14990 }%
14991 {%
14992     \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
14993 }%
14994 }%
14995 }
14996 \newcommand{\glsfielddef}[3]{%
14997   \glsdoifexists{#1}%
14998   {%
14999     \edef\@glo@label{\glsdetoklabel{#1}}%
15000     \ifcsdef{glo@\@glo@label @#2}%
15001     {%
15002       \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
15003     }%
15004     {%
15005       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15006     }%
15007   }%
15008 }
15009 \newcommand{\glsfieldfetch}[3]{%
15010   \glsdoifexists{#1}%
15011   {%
15012     \edef\@glo@label{\glsdetoklabel{#1}}%
15013     \ifcsdef{glo@\@glo@label @#2}%
15014     {%
15015       \letcs#3{glo@\@glo@label @#2}%
15016     }%
15017     {%
15018       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15019     }%
15020   }%
15021 }
15022 \newcommand{\ifglsfieldeq}[5]{%
15023   \glsdoifexists{#1}%
15024   {%
15025     \edef\@glo@label{\glsdetoklabel{#1}}%
15026     \ifcsdef{glo@\@glo@label @#2}%
15027     {%
15028       \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
15029     }%
15030     {%
15031       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15032     }%
15033   }%
15034 }
15035 \newcommand{\ifglsfielddefeq}[5]{%
15036   \glsdoifexists{#1}%
15037   {%

```

```

15038 \edef\@glo@label{\glsdetoklabel{#1}}%
15039 \ifcsdef{glo@\@glo@label @#2}%
15040 {%
15041     \expandafter\ifdefstrequal
15042     \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
15043 }%
15044 {%
15045     \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15046 }%
15047 }%
15048 }
15049 \newcommand{\ifglsfieldcseq}[5]{%
15050 \glsdoifexists{#1}%
15051 {%
15052     \edef\@glo@label{\glsdetoklabel{#1}}%
15053     \ifcsdef{glo@\@glo@label @#2}%
15054     {%
15055         \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
15056     }%
15057     {%
15058         \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15059     }%
15060 }%
15061 }
15062 \newcommand*\@gls@writedefhook{}
15063 \newcommand*\@gls@assign@desc}[1]{%
15064 \gls@assign@field{#1}{desc}{\@glo@desc}%
15065 \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
15066 }
15067 \newcommand{\longnewglossaryentry}[3]{%
15068 \glsdoifnoexists{#1}%
15069 {%
15070     \bgroup
15071     \let\@org@newglossaryentryprehook\@newglossaryentryprehook
15072     \long\def\@newglossaryentryprehook{%
15073         \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
15074         \@org@newglossaryentryprehook
15075     }%
15076     \renewcommand*\@gls@assign@desc}[1]{%
15077         \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
15078         \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
15079     }
15080     \gls@defglossaryentry{#1}{#2}%
15081     \egroup
15082 }
15083 }
15084 \@onlypreamble{\longnewglossaryentry}
15085 \newcommand{\longprovideglossaryentry}[3]{%
15086 \ifglsentryexists{#1}{}%

```

```

15087 {\longnewglossaryentry{#1}{#2}{#3}}%
15088 }
15089 \@onlypreamble{\longprovideglossaryentry}
15090 \newcommand{\gls@defglossaryentry}[2]{%
15091   \let\GlsSetQuote\gls@nosetquote
15092   \edef\@glo@label{\glsdetoklabel{#1}}%
15093   \let\glslabel\@glo@label
15094   \let\@glo@name\@glsnname
15095   \let\@glo@desc\@glsndesc
15096   \let\@glo@descplural\@gls@default@value
15097   \let\@glo@type\@gls@default@value
15098   \let\@glo@symbol\@gls@default@value
15099   \let\@glo@symbolplural\@gls@default@value
15100   \let\@glo@text\@gls@default@value
15101   \let\@glo@plural\@gls@default@value
15102   \let\@glo@first\@gls@default@value
15103   \let\@glo@firstplural\@gls@default@value
15104   \let\@glo@sort\@gls@default@value
15105   \let\@glo@counter\@gls@default@value
15106   \def\@glo@see{}%
15107   \def\@glo@parent{}%
15108   \def\@glo@prefix{}%
15109   \@gls@initnonumberlist
15110   \def\@glo@useri{}%
15111   \def\@glo@userii{}%
15112   \def\@glo@useriii{}%
15113   \def\@glo@useriv{}%
15114   \def\@glo@userv{}%
15115   \def\@glo@uservi{}%
15116   \def\@glo@short{}%
15117   \def\@glo@shortpl{}%
15118   \def\@glo@long{}%
15119   \def\@glo@longpl{}%
15120   \@newglossaryentryprehook
15121   \setkeys{glossentry}{#2}%
15122   \ifundef\glsdefaulttype
15123   {%
15124     \PackageError{glossaries}%
15125     {No default glossary type (have you used ‘nomain’ by mistake?)}%
15126     {If you use package option ‘nomain’ you must define
15127      a new glossary before you can define entries}%
15128   }%
15129   {%
15130     \gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
15131     \edef\@glo@type{\glsentrytype{\@glo@label}}%
15132     \ifcsundef{glolist@\@glo@type}%
15133     {%
15134       \PackageError{glossaries}%
15135       {Glossary type ‘\@glo@type’ has not been defined}%

```

```

15136      {You need to define a new glossary type, before making entries
15137      in it}%
15138  }%
15139  {%
15140    \ifignoredglossary\@glo@type
15141    {%
15142      \ifx\@glo@desc\@glsnodesc
15143      \let\@glo@desc\@empty
15144    \fi
15145    }%
15146    {%
15147    }%
15148    \protected@edef\@glo@list@\csname glolist@\@glo@type\endcsname}%
15149    \expandafter\xdef\csname glolist@\@glo@type\endcsname{%
15150      \@glo@list@\@glo@label},}%
15151  }%
15152  \gls@level=0\relax
15153  \ifx\@glo@parent\@empty
15154    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15155  \else
15156    \ifdefequal\@glo@label\@glo@parent%
15157    {%
15158      \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
15159      \def\@glo@parent{}%
15160      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15161    }%
15162    {%
15163      \ifglsentryexists{\@glo@parent}%
15164      {%
15165        \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
15166          \@glo@parent}%
15167        \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
15168        \advance\gls@level by 1\relax
15169        \ifx\@glo@name\@glsnoname
15170          \expandafter\let\expandafter\@glo@name
15171          \csname glo@\@glo@parent @name\endcsname
15172          \ifx\@glo@plural\@gls@default@value
15173            \expandafter\let\expandafter\@glo@plural
15174            \csname glo@\@glo@parent @plural\endcsname
15175          \fi
15176        \fi
15177      }%
15178    {%
15179      \PackageError{glossaries}%
15180      {%
15181        Invalid parent ‘\@glo@parent’
15182        for entry ‘\@glo@label’ - parent doesn’t exist%
15183      }%
15184    {%

```



```

15185         Parent entries must be defined before their children%
15186     }%
15187     \def\@glo@parent{%
15188         \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
15189     }%
15190 }%
15191 \fi
15192 \expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
15193 \gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
15194 \letcs\@glo@sort{glo@\@glo@label @sortvalue}%
15195 \gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
15196 \expandafter\gls@assign@field\expandafter
15197     {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
15198     {\@glo@label}{plural}{\@glo@plural}%
15199 \expandafter\gls@assign@field\expandafter
15200     {\csname glo@\@glo@label @text\endcsname}%
15201     {\@glo@label}{first}{\@glo@first}%
15202 \ifx\@glo@first\@gls@default@value
15203     \expandafter\gls@assign@field\expandafter
15204         {\csname glo@\@glo@label @plural\endcsname}%
15205         {\@glo@label}{firstpl}{\@glo@firstplural}%
15206 \else
15207     \expandafter\gls@assign@field\expandafter
15208         {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
15209         {\@glo@label}{firstpl}{\@glo@firstplural}%
15210 \fi
15211 \ifcsundef{@glotype@\@glo@type @counter}%
15212 {%
15213     \def\@glo@defaultcounter{\glscounter}%
15214 }%
15215 {%
15216     \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
15217 }%
15218 \gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
15219 \gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
15220 \gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
15221 \gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
15222 \gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
15223 \gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
15224 \gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
15225 \gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
15226 \gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
15227 \gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
15228 \gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
15229 \ifx\@glo@name\@gls@name
15230     \@gls@name
15231     \let\@glo@name\@gls@default@value
15232 \fi
15233 \gls@assign@field{}{\@glo@label}{name}{\@glo@name}%

```

```

15234 \ifcsundef{glo@\@glo@label @numberlist}%
15235 {%
15236   \csxdef{glo@\@glo@label @numberlist}{%
15237     \noexpand\@gls@missingnumberlist{\@glo@label}}%
15238   }%
15239   {%
15240     \@gls@storenonumberlist{\@glo@label}%
15241     \def\@glo@@desc{\@glo@first}%
15242     \ifx\@glo@desc\@glo@@desc
15243       \let\@glo@desc\@glo@first
15244     \fi
15245     \ifx\@glo@desc\@glsnodesc
15246       \@glsnodesc
15247       \let\@glo@desc\@gls@default@value
15248     \fi
15249     \@gls@assign@desc{\@glo@label}%
15250     \@gls@defsort{\@glo@type}{\@glo@label}%
15251     \def\@glo@@symbol{\@glo@text}%
15252     \ifx\@glo@symbol\@glo@@symbol
15253       \let\@glo@symbol\@glo@text
15254     \fi
15255     \@gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
15256     \expandafter
15257       \@gls@assign@field\expandafter
15258         {\csname glo@\@glo@label @symbol\endcsname}
15259         {\@glo@label}{symbolplural}{\@glo@symbolplural}%
15260     \expandafter\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
15261       \noexpand\global
15262         \noexpand\let\expandafter\noexpand
15263         \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
15264     }%
15265     \expandafter\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
15266       \noexpand\global
15267         \noexpand\let\expandafter\noexpand
15268         \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
15269     }%
15270     \csname glo@\@glo@label @flagfalse\endcsname
15271     \@glo@autosee
15272     \ifignoredglossary\@glo@type
15273     {%
15274       \csdef{glo@\@glo@label @index}{}%
15275     }
15276     {%
15277       \do@glo@storeentry{\@glo@label}%
15278     }%
15279     \@newglossaryentry@defcounters
15280     \@newglossaryentryposthook
15281 }
15282 \newcommand*{\@glo@autosee}{%

```

```

15283 \ifdefvoid\@glo@see{}%
15284 {%
15285     \protected@edef\@do@glsee{%
15286         \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
15287         \noexpand\expandafter\noexpand\@glsee\noexpand\@glo@list{\@glo@label}}%
15288     \@do@glsee
15289 }%
15290 \@glo@autoseehook
15291 }%
15292 \newcommand*{\@glo@autoseehook}{%}
15293 \newcommand*{\@newglossaryentryprehook}{%}
15294 \newcommand*{\@newglossaryentryposthook}{%}
15295 \newcommand*{\@newglossaryentry@defcounters}{%}
15296 \newcommand*{\glsmoveentry}[2]{%
15297     \edef\@glo@thislabel{\glsetoklabel{#1}}%
15298     \edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
15299     \def\glo@list{,%}
15300     \for\glseentries[\glo@type]{\glo@label}%
15301     {%
15302         \ifdefequal\@glo@thislabel\glo@label
15303             {\eappto\glo@list{\glo@label,}}%
15304         }%
15305     \cslet\glolist@\glo@type{\glo@list}%
15306     \csdef\glo@\@glo@thislabel @type{#2}%
15307 }
15308 \ifglxindy
15309     \newcommand*{\@glossaryentryfield}{\string\glossentry}
15310 \else
15311     \newcommand*{\@glossaryentryfield}{\string\glossentry}
15312 \fi
15313 \ifglxindy
15314     \newcommand*{\@glossarysubentryfield}{%
15315         \string\subglossentry}
15316 \else
15317     \newcommand*{\@glossarysubentryfield}{%
15318         \string\subglossentry}
15319 \fi
15320 \newcommand{\@glo@storeentry}[1]{%
15321     \edef\@glo@esclabel{#1}%
15322     \@gls@checkmkidxchars\@glo@esclabel
15323     \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
15324     \@gls@checkmkidxchars\@glo@sort
15325     \@gls@checkmkidxchars\@glo@prefix
15326     \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
15327     \ifglxindy
15328         \ifx\@glo@parent\@empty
15329             \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15330                 (\string"\@glo@sort\string" %
15331                 \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %

```

```

15332     }%
15333   \else
15334     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15335       \csname glo@\@glo@parent @index\endcsname
15336       (\string"\@glo@sort\string" %
15337       \string"\@glo@prefix\@glossarysubentryfield
15338       {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
15339     }%
15340   \fi
15341 \else
15342   \ifx\@glo@parent\@empty
15343     \@onelevel@sanitize\@glo@prefix
15344     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15345       \@glo@sort\@gls@actualchar\@glo@prefix
15346       \@glossaryentryfield{\@glo@esclabel}%
15347     }%
15348   \else
15349     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15350       \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
15351       \@glo@sort\@gls@actualchar\@glo@prefix
15352       \@glossarysubentryfield
15353       {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
15354     }%
15355   \fi
15356 \fi
15357 }
15358 \AtBeginDocument{%
15359   \@ifpackageloaded{amsmath}%
15360   {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
15361   {}%
15362 }
15363 \newcommand*{\@gls@ifnotmeasuring}[1]{%
15364   \ifmeasuring@
15365   \else
15366     #1%
15367   \fi
15368 }
15369 \newcommand*\gls@ifnotmeasuring[1]{#1}
15370 \def\@gls@patchtabularx#1\hbox#2#3!!{%
15371   \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
15372 }
15373 \newcommand*\glspatchtabularx{%
15374   \ifdef\TX@trial
15375   {%
15376     \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
15377     \let\glspatchtabularx\relax
15378   }%
15379   {}%
15380 }

```

```

15381 \newcommand*{\glsreset}[1]{%
15382   \gls@ifnotmeasuring
15383   {%
15384     \glsdoifexists{#1}%
15385     {%
15386       \@glsreset{#1}%
15387     }%
15388   }%
15389 }
15390 \newcommand*{\glslocalreset}[1]{%
15391   \gls@ifnotmeasuring
15392   {%
15393     \glsdoifexists{#1}%
15394     {%
15395       \@glslocalreset{#1}%
15396     }%
15397   }%
15398 }
15399 \newcommand*{\glsunset}[1]{%
15400   \gls@ifnotmeasuring
15401   {%
15402     \glsdoifexists{#1}%
15403     {%
15404       \@glsunset{#1}%
15405     }%
15406   }%
15407 }
15408 \newcommand*{\glslocalunset}[1]{%
15409   \gls@ifnotmeasuring
15410   {%
15411     \glsdoifexists{#1}%
15412     {%
15413       \@glslocalunset{#1}%
15414     }%
15415   }%
15416 }
15417 \newcommand*{\@glslocalunset}{\@@glslocalunset}
15418 \newcommand*{\@@glslocalunset}[1]{%
15419   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
15420 }
15421 \newcommand*{\@glsunset}{\@@glsunset}
15422 \newcommand*{\@@glsunset}[1]{%
15423   \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
15424 }
15425 \newcommand*{\@glslocalreset}{\@@glslocalreset}
15426 \newcommand*{\@@glslocalreset}[1]{%
15427   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
15428 }
15429 \newcommand*{\@glsreset}{\@@glsreset}

```

```

15430 \newcommand*{\@glsreset}[1]{%
15431   \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
15432 }
15433 \newcommand*{\glsresetall}[1][\@glo@types]{%
15434   \forallglsentries[#1]{\@glsentry}%
15435   {%
15436     \glsreset{\@glsentry}%
15437   }%
15438 }
15439 \newcommand*{\glslocalresetall}[1][\@glo@types]{%
15440   \forallglsentries[#1]{\@glsentry}%
15441   {%
15442     \glslocalreset{\@glsentry}%
15443   }%
15444 }
15445 \newcommand*{\glsunsetall}[1][\@glo@types]{%
15446   \forallglsentries[#1]{\@glsentry}%
15447   {%
15448     \glsunset{\@glsentry}%
15449   }%
15450 }
15451 \newcommand*{\glslocalunsetall}[1][\@glo@types]{%
15452   \forallglsentries[#1]{\@glsentry}%
15453   {%
15454     \glslocalunset{\@glsentry}%
15455   }%
15456 }
15457 \newcommand*{\@@newglossaryentry@defcounters}{%
15458   \csdef{glo@\@glo@label @currcount}{0}%
15459   \csdef{glo@\@glo@label @prevcount}{0}%
15460 }
15461 \newcommand*{\glsenableentrycount}{%
15462   \let\@newglossaryentry@defcounters\@@newglossaryentry@defcounters
15463   \renewcommand*{\gls@defdocnewglossaryentry}{%
15464     \renewcommand*\newglossaryentry[2]{%
15465       \PackageError{glossaries}{\string\newglossaryentry\space
15466         may only be used in the preamble when entry counting has
15467         been activated}{If you use \string\glsenableentrycount\space
15468         you must place all entry definitions in the preamble not in
15469         the document environment}%
15470     }%
15471   }%
15472   \newcommand*{\glsentrycurrcount}[1]{%
15473     \ifcsundef{glo@glsdetoklabel{##1}@currcount}%
15474     {0}{\@gls@entry@field{##1}{currcount}}%
15475   }%
15476   \newcommand*{\glsentryprevcount}[1]{%
15477     \ifcsundef{glo@glsdetoklabel{##1}@prevcount}%
15478     {0}{\@gls@entry@field{##1}{prevcount}}%

```

```

15479 }%
15480 \renewcommand*{\@glsunset}[1]{%
15481   \@@glsunset{##1}%
15482   \@gls@increment@currcount{##1}%
15483 }%
15484 \renewcommand*{\@glslocalunset}[1]{%
15485   \@@glslocalunset{##1}%
15486   \@gls@local@increment@currcount{##1}%
15487 }%
15488 \renewcommand*{\@glsreset}[1]{%
15489   \@@glsreset{##1}%
15490   \csgdef{glo@glsdetoklabel{##1}@currcount}{0}%
15491 }%
15492 \renewcommand*{\@glslocalreset}[1]{%
15493   \@@glslocalreset{##1}%
15494   \csdef{glo@glsdetoklabel{##1}@currcount}{0}%
15495 }%
15496 \def\@cgl@s@##1##2[##3]{%
15497   \ifnum\glsentryprevcount{##2}=1\relax
15498     \cgl@sformat{##2}{##3}%
15499     \glsunset{##2}%
15500   \else
15501     \@gls@{##1}{##2}[##3]%
15502   \fi
15503 }%
15504 \def\@cgl spl@##1##2[##3]{%
15505   \ifnum\glsentryprevcount{##2}=1\relax
15506     \cgl splformat{##2}{##3}%
15507     \glsunset{##2}%
15508   \else
15509     \@glspl@{##1}{##2}[##3]%
15510   \fi
15511 }%
15512 \def\@cGls@##1##2[##3]{%
15513   \ifnum\glsentryprevcount{##2}=1\relax
15514     \cGlsformat{##2}{##3}%
15515     \glsunset{##2}%
15516   \else
15517     \@Gls@{##1}{##2}[##3]%
15518   \fi
15519 }%
15520 \def\@cGlspl@##1##2[##3]{%
15521   \ifnum\glsentryprevcount{##2}=1\relax
15522     \cGlsplformat{##2}{##3}%
15523     \glsunset{##2}%
15524   \else
15525     \@Glspl@{##1}{##2}[##3]%
15526   \fi
15527 }%

```

```

15528 \AtEndDocument{\@gls@write@entrycounts}%
15529 \renewcommand*{\@gls@entry@count}[2]{%
15530   \csgdef{glo@glstdetoklabel{##1}@prevcount}{##2}%
15531 }%
15532 \let\glsenableentrycount\relax
15533 }
15534 \@onlypreamble\glsenableentrycount
15535 \newcommand*{\@gls@increment@currcount}[1]{%
15536   \csxdef{glo@glstdetoklabel{##1}@currcount}{%
15537     \number\numexpr\glsentrycurrcount{##1}+1}%
15538 }
15539 \newcommand*{\@gls@local@increment@currcount}[1]{%
15540   \csedef{glo@glstdetoklabel{##1}@currcount}{%
15541     \number\numexpr\glsentrycurrcount{##1}+1}%
15542 }
15543 \newcommand*{\@gls@write@entrycounts}{%
15544   \immediate\write\@auxout
15545     {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
15546   \forallglsentries{\@glsentry}{%
15547     \ifglsused{\@glsentry}%
15548     {\immediate\write\@auxout
15549       {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
15550   }%
15551 }%
15552 }
15553 \newcommand*{\@gls@entry@count}[2]{%
15554 \newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
15555 \newcommand*{\@cgl}[2][{}]{%
15556   \new@ifnextchar[{\@cgl@{##1}{##2}}{\@cgl@{##1}{##2}[]}%
15557 }
15558 \def\@cgl@#1#2[#3]{%
15559   \GlossariesWarning{\string\cgl\space is defaulting to
15560     \string\gls\space since you haven't enabled entry counting}%
15561   \@gls@{##1}{##2}[##3]}
15562 }
15563 \newcommand*{\cglformat}[2]{%
15564   \ifglshaslong{##1}{\glsentrylong{##1}}{\glsentryfirst{##1}}#2%
15565 }
15566 \newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
15567 \newcommand*{\@cGls}[2][{}]{%
15568   \new@ifnextchar[{\@cGls@{##1}{##2}}{\@cGls@{##1}{##2}[]}%
15569 }
15570 \def\@cGls@#1#2[#3]{%
15571   \GlossariesWarning{\string\cGls\space is defaulting to
15572     \string\Gls\space since you haven't enabled entry counting}%
15573   \@Gls@{##1}{##2}[##3]}
15574 }
15575 \newcommand*{\cGlsformat}[2]{%
15576   \ifglshaslong{##1}{\Glsentrylong{##1}}{\Glsentryfirst{##1}}#2%

```



```

15577 }
15578 \newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
15579 \newcommand*{\@cglspl}[2][]{%
15580   \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}[]}%
15581 }
15582 \def\@cglspl@#1#2[#3]{%
15583   \GlossariesWarning{\string\cglspl\space is defaulting to
15584     \string\glspl\space since you haven't enabled entry counting}%
15585   \@glspl@{#1}{#2}[#3]%
15586 }
15587 \newcommand*{\cglsplformat}[2]{%
15588   \ifglshaslong{#1}{\glentrylongpl{#1}}{\glentryfirstplural{#1}}#2%
15589 }
15590 \newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
15591 \newcommand*{\@cGlspl}[2][]{%
15592   \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[]}%
15593 }
15594 \def\@cGlspl@#1#2[#3]{%
15595   \GlossariesWarning{\string\cGlspl\space is defaulting to
15596     \string\Glspl\space since you haven't enabled entry counting}%
15597   \@Glspl@{#1}{#2}[#3]%
15598 }
15599 \newcommand*{\cGlsplformat}[2]{%
15600   \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
15601 }
15602 \newcommand*{\loadglsentries}[2][\@gls@default]{%
15603   \let\@gls@default\glsdefaulttype
15604   \def\glsdefaulttype{#1}\input{#2}%
15605   \let\glsdefaulttype\@gls@default
15606 }
15607 \@onlypreamble{\loadglsentries}
15608 \newcommand*{\glstextformat}[1]{#1}
15609 \newcommand*{\glentryfmt}{%
15610   \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
15611 }
15612 \newcommand*{\@@gls@default@entryfmt}[2]{%
15613   \ifdefempty\glscustomtext
15614     {%
15615       \glsifplural
15616       {%
15617         \glscapscase
15618         {%
15619           \ifglused\glslabel
15620             {%
15621               #2{\glentryplural{\glslabel}}%
15622               {\glentrydescplural{\glslabel}}%
15623               {\glentrysymbolplural{\glslabel}}{\glsinsert}%
15624             }%
15625             {%

```

```

15626      #1{\glsentryfirstplural{\glslabel}}}%
15627      {\glsentrydescplural{\glslabel}}}%
15628      {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15629  }%
15630 }%
15631 {%
15632   \ifglused\glslabel
15633   {%
15634     \ifbool{glscompatible-3.07}%
15635     {%
15636       \protected@edef\@glo@etext{%
15637         #2{\glsentryplural{\glslabel}}%
15638         {\glsentrydescplural{\glslabel}}%
15639         {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15640       \xmakefirstuc\@glo@etext
15641     }%
15642     {%
15643       #2{\Glsentryplural{\glslabel}}%
15644       {\glsentrydescplural{\glslabel}}%
15645       {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15646     }%
15647   }%
15648   {%
15649     \ifbool{glscompatible-3.07}%
15650     {%
15651       \protected@edef\@glo@etext{%
15652         #1{\glsentryfirstplural{\glslabel}}%
15653         {\glsentrydescplural{\glslabel}}%
15654         {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15655       \xmakefirstuc\@glo@etext
15656     }%
15657     {%
15658       #1{\Glsentryfirstplural{\glslabel}}%
15659       {\glsentrydescplural{\glslabel}}%
15660       {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15661     }%
15662   }%
15663 }%
15664 {%
15665   \ifglused\glslabel
15666   {%
15667     \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
15668     {\glsentrydescplural{\glslabel}}%
15669     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15670   }%
15671   {%
15672     \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
15673     {\glsentrydescplural{\glslabel}}%
15674     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%

```

```

15675     }%
15676 }%
15677 }%
15678 {%
15679     \glscapscase
15680 {%
15681     \ifglused\glslabel
15682     {%
15683         #2{\glsentrytext{\glslabel}}%
15684         {\glsentrydesc{\glslabel}}%
15685         {\glsentrysymbol{\glslabel}}{\glsinsert}%
15686     }%
15687     {%
15688         #1{\glsentryfirst{\glslabel}}%
15689         {\glsentrydesc{\glslabel}}%
15690         {\glsentrysymbol{\glslabel}}{\glsinsert}%
15691     }%
15692 }%
15693 {%
15694     \ifglused\glslabel
15695     {%
15696         \ifbool{glscompatible-3.07}%
15697         {%
15698             \protected@edef\@glo@etext{%
15699                 #2{\glsentrytext{\glslabel}}%
15700                 {\glsentrydesc{\glslabel}}%
15701                 {\glsentrysymbol{\glslabel}}{\glsinsert}}%
15702             \xmakefirstuc\@glo@etext
15703         }%
15704         {%
15705             #2{\Glsentrytext{\glslabel}}%
15706             {\glsentrydesc{\glslabel}}%
15707             {\glsentrysymbol{\glslabel}}{\glsinsert}%
15708         }%
15709     }%
15710     {%
15711         \ifbool{glscompatible-3.07}%
15712         {%
15713             \protected@edef\@glo@etext{%
15714                 #1{\glsentryfirst{\glslabel}}%
15715                 {\glsentrydesc{\glslabel}}%
15716                 {\glsentrysymbol{\glslabel}}{\glsinsert}}%
15717             \xmakefirstuc\@glo@etext
15718         }%
15719         {%
15720             #1{\Glsentryfirst{\glslabel}}%
15721             {\glsentrydesc{\glslabel}}%
15722             {\glsentrysymbol{\glslabel}}{\glsinsert}%
15723         }%

```

```

15724     }%
15725 }%
15726 {%
15727     \ifglused\glslabel
15728     {%
15729         \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}}%
15730         {\glsentrydesc{\glslabel}}}%
15731         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
15732     }%
15733     {%
15734         \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}}%
15735         {\glsentrydesc{\glslabel}}}%
15736         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
15737     }%
15738 }%
15739 }%
15740 }%
15741 {%
15742     \ifglused{\glslabel}%
15743     {%
15744         #2{\glscustomtext}%
15745         {\glsentrydesc{\glslabel}}}%
15746         {\glsentrysymbol{\glslabel}}{}%
15747     }%
15748     {%
15749         #1{\glscustomtext}%
15750         {\glsentrydesc{\glslabel}}}%
15751         {\glsentrysymbol{\glslabel}}{}%
15752     }%
15753 }%
15754 }
15755 \newcommand*{\glsgenentryfmt}{%
15756     \ifdefempty\glscustomtext
15757     {%
15758         \glsifplural
15759         {%
15760             \glscapscase
15761             {%
15762                 \ifglused\glslabel
15763                 {%
15764                     \glsentryplural{\glslabel}\glsinsert
15765                 }%
15766                 {%
15767                     \glsentryfirstplural{\glslabel}\glsinsert
15768                 }%
15769             }%
15770         }%
15771         \ifglused\glslabel
15772         {%

```

```

15773         \Glsentryplural{\glslabel}\glsinsert
15774     }%
15775     {%
15776         \Glsentryfirstplural{\glslabel}\glsinsert
15777     }%
15778 }%
15779 {%
15780     \ifglused\glslabel
15781     {%
15782         \mfirstucMakeUppercase
15783         {\Glsentryplural{\glslabel}\glsinsert}%
15784     }%
15785     {%
15786         \mfirstucMakeUppercase
15787         {\Glsentryfirstplural{\glslabel}\glsinsert}%
15788     }%
15789 }%
15790 }%
15791 {%
15792     \glscapscase
15793     {%
15794         \ifglused\glslabel
15795         {%
15796             \Glsentrytext{\glslabel}\glsinsert
15797         }%
15798         {%
15799             \Glsentryfirst{\glslabel}\glsinsert
15800         }%
15801     }%
15802     {%
15803         \ifglused\glslabel
15804         {%
15805             \Glsentrytext{\glslabel}\glsinsert
15806         }%
15807         {%
15808             \Glsentryfirst{\glslabel}\glsinsert
15809         }%
15810     }%
15811     {%
15812         \ifglused\glslabel
15813         {%
15814             \mfirstucMakeUppercase{\Glsentrytext{\glslabel}\glsinsert}%
15815         }%
15816         {%
15817             \mfirstucMakeUppercase{\Glsentryfirst{\glslabel}\glsinsert}%
15818         }%
15819     }%
15820 }%
15821 }%

```

```

15822 {%
15823     \glscustomtext\glsinsert
15824 }%
15825 }
15826 \newcommand*{\glsgenacfmt}{%
15827     \ifdefempty\glscustomtext
15828     {%
15829         \ifglssused\glslabel
15830         {%
15831             \glsifplural
15832             {%
15833                 \glscapscase
15834                 {%
15835                     \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
15836                 }%
15837                 {%
15838                     \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
15839                 }%
15840                 {%
15841                     \mfirstucMakeUppercase
15842                     {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
15843                 }%
15844             }%
15845             {%
15846                 \glscapscase
15847                 {%
15848                     \acronymfont{\glsentryshort{\glslabel}}\glsinsert
15849                 }%
15850                 {%
15851                     \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
15852                 }%
15853                 {%
15854                     \mfirstucMakeUppercase
15855                     {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
15856                 }%
15857             }%
15858         }%
15859         {%
15860             \glsifplural
15861             {%
15862                 \glscapscase
15863                 {%
15864                     \genplacrfullformat{\glslabel}{\glsinsert}%
15865                 }%
15866                 {%
15867                     \Genplacrfullformat{\glslabel}{\glsinsert}%
15868                 }%
15869                 {%
15870                     \mfirstucMakeUppercase

```

```

15871         {\genplacrfullformat{\glslabel}{\glsinsert}}}%
15872     }%
15873 }%
15874 {%
15875     \glscapscase
15876     {%
15877         \genacrfullformat{\glslabel}{\glsinsert}%
15878     }%
15879     {%
15880         \Genacrfullformat{\glslabel}{\glsinsert}%
15881     }%
15882     {%
15883         \mfirstucMakeUppercase
15884         {\genacrfullformat{\glslabel}{\glsinsert}}}%
15885     }%
15886 }%
15887 }%
15888 }%
15889 {%
15890     \glscustomtext
15891 }%
15892 }
15893 \newcommand*{\genacrfullformat}[2]{%
15894     \glsentrylong{#1}#2\space
15895     (\protect\firstacronymfont{\glsentryshort{#1}})%
15896 }
15897 \newcommand*{\Genacrfullformat}[2]{%
15898     \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
15899     \xmakefirstuc\gls@text
15900 }
15901 \newcommand*{\genplacrfullformat}[2]{%
15902     \glsentrylongpl{#1}#2\space
15903     (\protect\firstacronymfont{\glsentryshortpl{#1}})%
15904 }
15905 \newcommand*{\Genplacrfullformat}[2]{%
15906     \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
15907     \xmakefirstuc\gls@text
15908 }
15909 \newcommand*{\glsdisplayfirst}[4]{#1#4}
15910 \newcommand*{\glsdisplay}[4]{#1#4}
15911 \newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
15912     \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
15913     Use \string\defglsentryfmt\space instead}%
15914     \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
15915     \edef\@gls@doentrydef{%
15916         \noexpand\defglsentryfmt[#1]{%
15917             \noexpand\ifcsdef{gls@#1@displayfirst}%
15918             {%
15919                 \noexpand\@@gls@default@entryfmt

```

```

15920         {\noexpand\csuse{gls@#1@displayfirst}}%
15921         {\noexpand\csuse{gls@#1@display}}%
15922     }%
15923     {%
15924         \noexpand\@gls@default@entryfmt
15925         {\noexpand\glsdisplayfirst}%
15926         {\noexpand\csuse{gls@#1@display}}%
15927     }%
15928 }%
15929 }%
15930 \@gls@doentrydef
15931 }
15932 \newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
15933 \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
15934 Use \string\defglsentryfmt\space instead}%
15935 \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
15936 \edef\@gls@doentrydef{%
15937     \noexpand\defglsentryfmt[#1]{%
15938         \noexpand\ifcsdef{gls@#1@display}%
15939         {%
15940             \noexpand\@gls@default@entryfmt
15941             {\noexpand\csuse{gls@#1@displayfirst}}%
15942             {\noexpand\csuse{gls@#1@display}}%
15943         }%
15944         {%
15945             \noexpand\@gls@default@entryfmt
15946             {\noexpand\csuse{gls@#1@displayfirst}}%
15947             {\noexpand\glsdisplay}%
15948         }%
15949     }%
15950 }%
15951 \@gls@doentrydef
15952 }
15953 \define@key{glslink}{counter}{%
15954 \ifcsundef{c@#1}%
15955 {%
15956     \PackageError{glossaries}%
15957     {There is no counter called ‘#1’}%
15958     {%
15959         The counter key should have the name of a valid counter
15960         as its value%
15961     }%
15962 }%
15963 {%
15964     \def\@gls@counter{#1}%
15965 }%
15966 }
15967 \define@key{glslink}{format}{%
15968 \def\@glsnumberformat{#1}}

```



```

15969 \define@boolkey{glslink}{hyper}[true]{}
15970 \ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
15971 \define@boolkey{glslink}{local}[true]{}
15972 \newcommand*{\glslinkvar}[3]{#1}
15973 \newcommand*{\glsifhyper}[2]{%
15974   \glslinkvar{#1}{#2}{#1}%
15975   \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
15976     you mean \string\glsifhyperon\space or \string\glslinkvar?}%
15977 }
15978 \newcommand*{\@gls@hyp@opt}[1]{%
15979   \let\glslinkvar\@firstofthree
15980   \let\@gls@hyp@opt@cs#1\relax
15981   \@ifstar{\s@gls@hyp@opt}%
15982   {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
15983 }
15984 \newcommand*{\s@gls@hyp@opt}[1][]{%
15985   \let\glslinkvar\@secondofthree
15986   \@gls@hyp@opt@cs[hyper=false,#1]}
15987 \newcommand*{\p@gls@hyp@opt}[1][]{%
15988   \let\glslinkvar\@thirdofthree
15989   \@gls@hyp@opt@cs[hyper=true,#1]}
15990 \newrobustcmd*{\glslink}{%
15991   \@gls@hyp@opt\@gls@link
15992 }
15993 \newcommand*{\@gls@link}[3][]{%
15994   \glsdoifexistsordo{#2}%
15995   {%
15996     \let\do@gls@link@checkfirsthyper\relax
15997     \@gls@link[#1]{#2}{#3}%
15998   }{%
15999     \glstextformat{#3}%
16000   }%
16001   \glspostlinkhook
16002 }
16003 \newcommand*{\glspostlinkhook}{}
16004 \newcommand*{\@gls@link@checkfirsthyper}{%
16005   \ifglssused{\glslabel}%
16006   {%
16007   }%
16008   {%
16009     \gls@checkisacronymlist\glstype
16010     \ifglshyperfirst
16011       \if@glsisacronymlist
16012         \ifglssacrfootnote
16013           \KV@glslink@hyperfalse
16014           \fi
16015         \fi
16016       \else
16017         \KV@glslink@hyperfalse

```

```

16018 \fi
16019 }%
16020 \glslinkcheckfirsthyperhook
16021 }
16022 \newcommand*{\glslinkcheckfirsthyperhook}{}
16023 \newcommand*{\glslinkpostsetkeys}{}
16024 \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
16025 \newcommand*{\do@glsglssdisablehyperinlist}{}%
16026 \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
16027 {\KV@glslink@hyperfalse}{}%
16028 }
16029 \newcommand*{\@gls@setdefault@glslink@opts}{}
16030 \def\@gls@link[#1]#2#3{%
16031 \leavevmode
16032 \edef\glslabel{\glsdetoklabel{#2}}%
16033 \def\@gls@link@opts{#1}%
16034 \let\@gls@link@label\glslabel
16035 \def\@glsnumberformat{glsnumberformat}%
16036 \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
16037 \edef\glstype{\csname glo@\glslabel @type\endcsname}%
16038 \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
16039 \@gls@setdefault@glslink@opts
16040 \do@glsglssdisablehyperinlist
16041 \do@gls@link@checkfirsthyper
16042 \setkeys{glslink}{#1}%
16043 \glslinkpostsetkeys
16044 \@gls@saveentrycounter
16045 \@gls@setsort{\glslabel}%
16046 \@do@wrglossary{#2}%
16047 \ifKV@glslink@hyper
16048 \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
16049 \else
16050 \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
16051 \fi
16052 \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
16053 }
16054 \newcommand*{\glolinkprefix}{glo:}
16055 \def\glsentrycounter{\glscounter}%
16056 \newcommand*{\@gls@saveentrycounter}{}%
16057 \def\@gls@Hcounter{}%
16058 \ifthenelse{\equal{\@gls@counter}{equation}}%
16059 {
16060 \ifcsundef{xatlevel@}%
16061 {%
16062 \edef\theglsentrycounter{\expandafter\noexpand
16063 \csname the\@gls@counter\endcsname}%
16064 }%
16065 {%
16066 \ifx\xatlevel@\@empty

```

```

16067     \edef\theglsentrycounter{\expandafter\noexpand
16068         \csname the\@gls@counter\endcsname}%
16069     \else
16070         \savecounters@
16071         \advance\c@equation by 1\relax
16072         \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
16073         \ifcsundef{theH\@gls@counter}%
16074             {%
16075                 \def\@gls@Hcounter{\theglsentrycounter}%
16076             }%
16077             {%
16078                 \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
16079             }%
16080         \protected@edef\theHglentrycounter{\@gls@Hcounter}%
16081         \restorecounters@
16082     \fi
16083 }%
16084 }%
16085 {%
16086     \edef\theglsentrycounter{\expandafter\noexpand
16087         \csname the\@gls@counter\endcsname}%
16088 }%
16089 \ifx\@gls@Hcounter\@empty
16090     \ifcsundef{theH\@gls@counter}%
16091         {%
16092             \def\theHglentrycounter{\theglsentrycounter}%
16093         }%
16094         {%
16095             \protected@edef\theHglentrycounter{\expandafter\noexpand
16096                 \csname theH\@gls@counter\endcsname}%
16097             }%
16098         \fi
16099 }
16100 \def\@set@glo@numformat#1#2#3#4{%
16101     \expandafter\@glo@check@mkidxrangechar#3\@nil
16102     \protected@edef#1{%
16103         \@glo@prefix setentrycounter[#4]{#2}%
16104         \expandafter\string\csname\@glo@suffix\endcsname
16105     }%
16106     \@gls@checkmkidxchars#1%
16107 }
16108 \def\@glo@check@mkidxrangechar#1#2\@nil{%
16109 \if#1(\relax
16110     \def\@glo@prefix{(%
16111     \if\relax#2\relax
16112         \def\@glo@suffix{glsnumberformat}%
16113     \else
16114         \def\@glo@suffix{#2}%
16115     \fi

```

```

16116 \else
16117   \if#1)\relax
16118     \def\@glo@prefix{}}}%
16119     \if\relax#2\relax
16120       \def\@glo@suffi{x{glsnumberformat}}}%
16121       \else
16122         \def\@glo@suffi{x{#2}}}%
16123     \fi
16124   \else
16125     \def\@glo@prefix{}}\def\@glo@suffi{x{#1#2}}}%
16126   \fi
16127 \fi}
16128 \newcommand*{\@gls@escbsdq}[1]{%
16129   \def\@gls@checkedmkidx{}%
16130   \let\gls@xdystring=#1\relax
16131   \@onelevel@sanitize\gls@xdystring
16132   \edef\do@gls@xdycheckbackslash{%
16133     \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
16134     \@backslashchar\@backslashchar\noexpand\null}%
16135   \do@gls@xdycheckbackslash
16136   \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
16137   \def\@gls@checkedmkidx{}%
16138   \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
16139   \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
16140   \@for\@gls@tmp:=\gls@protected@pagefmts\do
16141   {%
16142     \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
16143     \@onelevel@sanitize\@gls@sanitized@tmp
16144     \edef\gls@dostubst{%
16145       \noexpand\DTLsubstituteall\noexpand\gls@xdystring
16146       {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
16147     }%
16148     \gls@dostubst
16149   }%
16150   \let#1=\gls@xdystring
16151 }
16152 \newcommand{\@gls@checkmkidxchars}[1]{%
16153   \ifglxindy
16154     \@gls@escbsdq{#1}%
16155   \else
16156     \def\@gls@checkedmkidx{}%
16157     \expandafter\@gls@checkquote#1\@nil""\null
16158     \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16159     \def\@gls@checkedmkidx{}%
16160     \expandafter\@gls@checkescquote#1\@nil\""\null
16161     \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16162     \def\@gls@checkedmkidx{}%
16163     \expandafter\@gls@checkescactual#1\@nil\?\?\null
16164     \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%

```

```

16165 \def\@gls@checkedmkidx{%
16166 \expandafter\@gls@checkactual#1\@nil??\null
16167 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16168 \def\@gls@checkedmkidx{%
16169 \expandafter\@gls@checkbar#1\@nil||\null
16170 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16171 \def\@gls@checkedmkidx{%
16172 \expandafter\@gls@checkescbar#1\@nil\\|\null
16173 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16174 \def\@gls@checkedmkidx{%
16175 \expandafter\@gls@checklevel#1\@nil!!\null
16176 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16177 \fi
16178 }
16179 \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
16180 \newtoks\@gls@tmpb
16181 \def\@gls@checkquote#1"#2"#3\null{%
16182 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16183 \toks@={#1}%
16184 \ifx\null#2\null
16185 \ifx\null#3\null
16186 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16187 \def\@gls@checkquote{\relax}%
16188 \else
16189 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16190 \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
16191 \def\@gls@checkquote{\@gls@checkquote#3\null}%
16192 \fi
16193 \else
16194 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16195 \@gls@quotechar\@gls@quotechar}%
16196 \ifx\null#3\null
16197 \def\@gls@checkquote{\@gls@checkquote#2""\null}%
16198 \else
16199 \def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
16200 \fi
16201 \fi
16202 \@gls@checkquote
16203 }
16204 \def\@gls@checkescquote#1"#2"#3\null{%
16205 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16206 \toks@={#1}%
16207 \ifx\null#2\null
16208 \ifx\null#3\null
16209 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16210 \def\@gls@checkescquote{\relax}%
16211 \else
16212 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16213 \@gls@quotechar\string"\@gls@quotechar

```

```

16214     \@gls@quotechar\string\" \@gls@quotechar}%
16215     \def\@@gls@checkescquote{\@gls@checkescquote#3\null}%
16216     \fi
16217   \else
16218     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16219       \@gls@quotechar\string\" \@gls@quotechar}%
16220     \ifx\null#3\null
16221       \def\@@gls@checkescquote{\@gls@checkescquote#2\" \"\null}%
16222     \else
16223       \def\@@gls@checkescquote{\@gls@checkescquote#2\"#3\null}%
16224     \fi
16225   \fi
16226 \@@gls@checkescquote
16227 }
16228 \def\@gls@checkescactual#1\?#2\?#3\null{%
16229   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16230   \toks@={#1}%
16231   \ifx\null#2\null
16232     \ifx\null#3\null
16233       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16234       \def\@@gls@checkescactual{\relax}%
16235     \else
16236       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16237         \@gls@quotechar\string\" \@gls@actualchar
16238         \@gls@quotechar\string\" \@gls@actualchar}%
16239       \def\@@gls@checkescactual{\@gls@checkescactual#3\null}%
16240     \fi
16241   \else
16242     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16243       \@gls@quotechar\string\" \@gls@actualchar}%
16244     \ifx\null#3\null
16245       \def\@@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
16246     \else
16247       \def\@@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
16248     \fi
16249   \fi
16250 \@@gls@checkescactual
16251 }
16252 \def\@gls@checkescbar#1\|#2\|#3\null{%
16253   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16254   \toks@={#1}%
16255   \ifx\null#2\null
16256     \ifx\null#3\null
16257       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16258       \def\@@gls@checkescbar{\relax}%
16259     \else
16260       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16261         \@gls@quotechar\string\" \@gls@encapchar
16262         \@gls@quotechar\string\" \@gls@encapchar}%

```

```

16263 \def\@gls@checkesbar{\@gls@checkesbar#3\null}%
16264 \fi
16265 \else
16266 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16267 \@gls@quotechar\string"\@gls@encapchar}%
16268 \ifx\null#3\null
16269 \def\@gls@checkesbar{\@gls@checkesbar#2\|\|\null}%
16270 \else
16271 \def\@gls@checkesbar{\@gls@checkesbar#2\|#3\null}%
16272 \fi
16273 \fi
16274 \@gls@checkesbar
16275 }
16276 \def\@gls@checkesclevel#1\!#2\!#3\null{%
16277 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16278 \toks@={#1}%
16279 \ifx\null#2\null
16280 \ifx\null#3\null
16281 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16282 \def\@gls@checkesclevel{\relax}%
16283 \else
16284 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16285 \@gls@quotechar\string"\@gls@levelchar
16286 \@gls@quotechar\string"\@gls@levelchar}%
16287 \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
16288 \fi
16289 \else
16290 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16291 \@gls@quotechar\string"\@gls@levelchar}%
16292 \ifx\null#3\null
16293 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
16294 \else
16295 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
16296 \fi
16297 \fi
16298 \@gls@checkesclevel
16299 }
16300 \def\@gls@checkbar#1|#2|#3\null{%
16301 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16302 \toks@={#1}%
16303 \ifx\null#2\null
16304 \ifx\null#3\null
16305 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16306 \def\@gls@checkbar{\relax}%
16307 \else
16308 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16309 \@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
16310 \def\@gls@checkbar{\@gls@checkbar#3\null}%
16311 \fi

```

```

16312 \else
16313 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16314 \@gls@quotechar\@gls@encapchar}%
16315 \ifx\null#3\null
16316 \def\@gls@checkbar{\@gls@checkbar#2||\null}%
16317 \else
16318 \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
16319 \fi
16320 \fi
16321 \@gls@checkbar
16322 }
16323 \def\@gls@checklevel#1!#2!#3\null{%
16324 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16325 \toks@={#1}%
16326 \ifx\null#2\null
16327 \ifx\null#3\null
16328 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16329 \def\@gls@checklevel{\relax}%
16330 \else
16331 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16332 \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
16333 \def\@gls@checklevel{\@gls@checklevel#3\null}%
16334 \fi
16335 \else
16336 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16337 \@gls@quotechar\@gls@levelchar}%
16338 \ifx\null#3\null
16339 \def\@gls@checklevel{\@gls@checklevel#2!!\null}%
16340 \else
16341 \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
16342 \fi
16343 \fi
16344 \@gls@checklevel
16345 }
16346 \def\@gls@checkactual#1?#2?#3\null{%
16347 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16348 \toks@={#1}%
16349 \ifx\null#2\null
16350 \ifx\null#3\null
16351 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16352 \def\@gls@checkactual{\relax}%
16353 \else
16354 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16355 \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
16356 \def\@gls@checkactual{\@gls@checkactual#3\null}%
16357 \fi
16358 \else
16359 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16360 \@gls@quotechar\@gls@actualchar}%

```



```

16361 \ifx\null#3\null
16362 \def\@gls@checkactual{\@gls@checkactual#2??\null}%
16363 \else
16364 \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
16365 \fi
16366 \fi
16367 \@gls@checkactual
16368 }
16369 \def\@gls@xdycheckquote#1"#2"#3\null{%
16370 \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16371 \toks@={#1}%
16372 \ifx\null#2\null
16373 \ifx\null#3\null
16374 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16375 \def\@gls@xdycheckquote{\relax}%
16376 \else
16377 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16378 \string"\string"}%
16379 \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
16380 \fi
16381 \else
16382 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16383 \string"}%
16384 \ifx\null#3\null
16385 \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
16386 \else
16387 \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
16388 \fi
16389 \fi
16390 \@gls@xdycheckquote
16391 }
16392 \edef\def\@gls@xdycheckbackslash{%
16393 \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
16394 ##2\@backslashchar##3\noexpand\null{%
16395 \noexpand\@gls@tmpb=\noexpand\expandafter
16396 {\noexpand\@gls@checkedmkidx}%
16397 \noexpand\toks@={##1}%
16398 \noexpand\ifx\noexpand\null##2\noexpand\null
16399 \noexpand\ifx\noexpand\null##3\noexpand\null
16400 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16401 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
16402 \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
16403 \noexpand\else
16404 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16405 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
16406 \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
16407 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16408 \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
16409 \noexpand\fi

```

```

16410 \noexpand\else
16411 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16412 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
16413 \@backslashchar\@backslashchar}%
16414 \noexpand\ifx\noexpand\null##3\noexpand\null
16415 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16416 \noexpand\@gls@xdycheckbackslash##2\@backslashchar
16417 \@backslashchar\noexpand\null}%
16418 \noexpand\else
16419 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16420 \noexpand\@gls@xdycheckbackslash##2\@backslashchar
16421 ##3\noexpand\null}%
16422 \noexpand\fi
16423 \noexpand\fi
16424 \noexpand\@gls@xdycheckbackslash
16425 }%
16426 }
16427 \def@gls@xdycheckbackslash
16428
16429 \newlength@gls@tmplen
16430 \newcommand*\@glsdohypertarget}[2]{%
16431 \@glsshowtarget{#1}%
16432 \settoheight@gls@tmplen{#2}%
16433 \raisebox@gls@tmplen{\hypertarget{#1}}{#2}%
16434 }
16435 \newcommand*\@glsdohyperlink}[2]{%
16436 \@glsshowtarget{#1}%
16437 \hyperlink{#1}{#2}%
16438 }
16439 \newcommand*\@glsdonohyperlink}[2]{#2}
16440 \ifcsundef{hyperlink}%
16441 {%
16442 \let@glslink@glsdonohyperlink
16443 }%
16444 {%
16445 \let@glslink@glsdohyperlink
16446 }
16447 \ifcsundef{hypertarget}%
16448 {%
16449 \let@glstarget\@secondoftwo
16450 }%
16451 {%
16452 \let@glstarget@glsdohypertarget
16453 }
16454 \newcommand*\@glsdisablehyper}{%
16455 \KV@glslink@hyperfalse
16456 \let@glslink@glsdonohyperlink
16457 \let@glstarget\@secondoftwo
16458 }

```

```

16459 \newcommand{\glsenablehyper}{%
16460   \KV@glslink@hypertrue
16461   \let\@glslink\glsdohyperlink
16462   \let\@glstarget\glsdohypertarget
16463 }
16464 \providecommand{\@firstofthree}[3]{#1}
16465 \providecommand{\@secondofthree}[3]{#2}
16466 \newrobustcmd*{\@gls}{\@gls@hyp@opt\@gls}
16467 \newcommand*{\@gls}[2][{}]{%
16468   \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
16469 }
16470 \def\@gls@#1#2[#3]{%
16471   \glstoifexists{#2}%
16472   {%
16473     \let\do@glslink@checkfirsthyper\@glslink@checkfirsthyper
16474     \let\glsifplural\@secondoftwo
16475     \let\glscapscase\@firstofthree
16476     \let\glscustomtext\@empty
16477     \def\glsinsert{#3}%
16478     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16479     \@glslink[#1]{#2}{\@glo@text}%
16480     \ifKV@glslink@local
16481       \glslocalunset{#2}%
16482     \else
16483       \glsunset{#2}%
16484     \fi
16485   }%
16486   \glspostlinkhook
16487 }
16488 \newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
16489 \newcommand*{\@Gls}[2][{}]{%
16490   \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
16491 }
16492 \def\@Gls@#1#2[#3]{%
16493   \glstoifexists{#2}%
16494   {%
16495     \let\do@glslink@checkfirsthyper\@glslink@checkfirsthyper
16496     \let\glsifplural\@secondoftwo
16497     \let\glscapscase\@secondofthree
16498     \let\glscustomtext\@empty
16499     \def\glsinsert{#3}%
16500     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16501     \@glslink[#1]{#2}{\@glo@text}%
16502     \ifKV@glslink@local
16503       \glslocalunset{#2}%
16504     \else
16505       \glsunset{#2}%
16506     \fi
16507   }%

```

```

16508 \glspostlinkhook
16509 }
16510 \newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
16511 \newcommand*{\@GLS}[2] [] {%
16512   \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}]%
16513 }
16514 \def\@GLS@#1#2[#3] {%
16515   \glsdoifexists{#2}%
16516   {%
16517     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16518     \let\glsifplural\@secondoftwo
16519     \let\glscapscase\@thirdofthree
16520     \let\glscustomtext\@empty
16521     \def\glsinsert{#3}%
16522     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16523     \@gls@link[#1]{#2}{\@glo@text}%
16524     \ifKV@glslink@local
16525       \glslocalunset{#2}%
16526     \else
16527       \glsunset{#2}%
16528     \fi
16529   }%
16530   \glspostlinkhook
16531 }
16532 \newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
16533 \newcommand*{\@glspl}[2] [] {%
16534   \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}]%
16535 }
16536 \def\@glspl@#1#2[#3] {%
16537   \glsdoifexists{#2}%
16538   {%
16539     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16540     \let\glsifplural\@firstoftwo
16541     \let\glscapscase\@firstofthree
16542     \let\glscustomtext\@empty
16543     \def\glsinsert{#3}%
16544     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16545     \@gls@link[#1]{#2}{\@glo@text}%
16546     \ifKV@glslink@local
16547       \glslocalunset{#2}%
16548     \else
16549       \glsunset{#2}%
16550     \fi
16551   }%
16552   \glspostlinkhook
16553 }
16554 \newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
16555 \newcommand*{\@Glspl}[2] [] {%
16556   \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}]%

```

```

16557 }
16558 \def\@GLSpl@#1#2[#3]{%
16559   \glsdoifexists{#2}%
16560   {%
16561     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16562     \let\glsifplural\@firstoftwo
16563     \let\glscapscase\@secondofthree
16564     \let\glscustomtext\@empty
16565     \def\glsinsert{#3}%
16566     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16567     \@gl@link[#1]{#2}{\@glo@text}%
16568     \ifKV@gl@link@local
16569       \glslocalunset{#2}%
16570     \else
16571       \glsunset{#2}%
16572     \fi
16573   }%
16574   \glspostlinkhook
16575 }
16576 \newrobustcmd*{\GLSpl}{\@gl@hyp@opt\@GLSpl}
16577 \newcommand*{\@GLSpl}[2][{}]{%
16578   \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
16579 }
16580 \def\@GLSpl@#1#2[#3]{%
16581   \glsdoifexists{#2}%
16582   {%
16583     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16584     \let\glsifplural\@firstoftwo
16585     \let\glscapscase\@thirdofthree
16586     \let\glscustomtext\@empty
16587     \def\glsinsert{#3}%
16588     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16589     \@gl@link[#1]{#2}{\@glo@text}%
16590     \ifKV@gl@link@local
16591       \glslocalunset{#2}%
16592     \else
16593       \glsunset{#2}%
16594     \fi
16595   }%
16596   \glspostlinkhook
16597 }
16598 \newrobustcmd*{\glsdisp}{\@gl@hyp@opt\@glsdisp}
16599 \newcommand*{\@glsdisp}[3][{}]{%
16600   \glsdoifexists{#2}{%
16601     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16602     \let\glsifplural\@secondoftwo
16603     \let\glscapscase\@firstofthree
16604     \def\glscustomtext{#3}%
16605     \def\glsinsert{}%

```

```

16606 \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16607 \@gls@link[#1]{#2}{\@glo@text}%
16608 \ifKV@gls@link@local
16609 \glslocalunset{#2}%
16610 \else
16611 \glsunset{#2}%
16612 \fi
16613 }%
16614 \gls@postlinkhook
16615 }
16616 \newcommand*{\@gls@link@nocheckfirsthyper}{%
16617 \newcommand{\@gls@field@link}[3]{%
16618 \glsdoifexists{#2}%
16619 {%
16620 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16621 \@gls@link[#1]{#2}{#3}%
16622 }%
16623 \gls@postlinkhook
16624 }
16625 \newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
16626 \newcommand*{\@glstext}[2][]{%
16627 \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
16628 \def\@glstext@#1#2[#3]{%
16629 \@gls@field@link{#1}{#2}{\glstentrytext{#2}#3}%
16630 }
16631 \newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
16632 \newcommand*{\@GLStext}[2][]{%
16633 \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
16634 \def\@GLStext@#1#2[#3]{%
16635 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrytext{#2}#3}%
16636 }
16637 \newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
16638 \newcommand*{\@Glstext}[2][]{%
16639 \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
16640 \def\@Glstext@#1#2[#3]{%
16641 \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
16642 }
16643 \newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
16644 \newcommand*{\@glsfirst}[2][]{%
16645 \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
16646 \def\@glsfirst@#1#2[#3]{%
16647 \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
16648 }
16649 \newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
16650 \newcommand*{\@Glsfirst}[2][]{%
16651 \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}}
16652 \def\@Glsfirst@#1#2[#3]{%
16653 \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
16654 }

```

```

16655 \newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
16656 \newcommand*{\@GLSfirst}[2] [] {%
16657   \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
16658 \def\@GLSfirst@#1#2[#3] {%
16659   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}{#3}}}%
16660 }
16661 \newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
16662 \newcommand*{\@glsplural}[2] [] {%
16663   \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
16664 \def\@glsplural@#1#2[#3] {%
16665   \@gls@field@link{#1}{#2}{\glsentryplural{#2}{#3}}%
16666 }
16667 \newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
16668 \newcommand*{\@Glsplural}[2] [] {%
16669   \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
16670 \def\@Glsplural@#1#2[#3] {%
16671   \@gls@field@link{#1}{#2}{\Glsentryplural{#2}{#3}}%
16672 }
16673 \newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
16674 \newcommand*{\@GLSplural}[2] [] {%
16675   \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
16676 \def\@GLSplural@#1#2[#3] {%
16677   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}{#3}}}%
16678 }
16679 \newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
16680 \newcommand*{\@glsfirstplural}[2] [] {%
16681   \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
16682 \def\@glsfirstplural@#1#2[#3] {%
16683   \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}{#3}}%
16684 }
16685 \newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
16686 \newcommand*{\@Glsfirstplural}[2] [] {%
16687   \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
16688 \def\@Glsfirstplural@#1#2[#3] {%
16689   \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}{#3}}%
16690 }
16691 \newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
16692 \newcommand*{\@GLSfirstplural}[2] [] {%
16693   \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
16694 \def\@GLSfirstplural@#1#2[#3] {%
16695   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}{#3}}}%
16696 }
16697 \newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
16698 \newcommand*{\@glsname}[2] [] {%
16699   \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
16700 \def\@glsname@#1#2[#3] {%
16701   \@gls@field@link{#1}{#2}{\glsentryname{#2}{#3}}%
16702 }
16703 \newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}

```

```

16704 \newcommand*{\@Glsname}[2] [] {%
16705   \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
16706 \def\@Glsname@#1#2[#3] {%
16707   \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
16708 }
16709 \newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
16710 \newcommand*{\@GLSname}[2] [] {%
16711   \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
16712 \def\@GLSname@#1#2[#3] {%
16713   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryname{#2}#3}}%
16714 }
16715 \newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
16716 \newcommand*{\@glsdesc}[2] [] {%
16717   \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
16718 \def\@glsdesc@#1#2[#3] {%
16719   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
16720 }
16721 \newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
16722 \newcommand*{\@GLSdesc}[2] [] {%
16723   \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
16724 \def\@GLSdesc@#1#2[#3] {%
16725   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
16726 }
16727 \newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
16728 \newcommand*{\@GLSdesc}[2] [] {%
16729   \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
16730 \def\@GLSdesc@#1#2[#3] {%
16731   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydesc{#2}#3}}%
16732 }
16733 \newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
16734 \newcommand*{\@glsdescplural}[2] [] {%
16735   \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
16736 \def\@glsdescplural@#1#2[#3] {%
16737   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
16738 }
16739 \newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
16740 \newcommand*{\@GLSdescplural}[2] [] {%
16741   \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
16742 \def\@GLSdescplural@#1#2[#3] {%
16743   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
16744 }
16745 \newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
16746 \newcommand*{\@GLSdescplural}[2] [] {%
16747   \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
16748 \def\@GLSdescplural@#1#2[#3] {%
16749   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydescplural{#2}#3}}%
16750 }
16751 \newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
16752 \newcommand*{\@glssymbol}[2] [] {%

```



```

16753 \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
16754 \def\@glssymbol@#1#2[#3]{%
16755 \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
16756 }
16757 \newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
16758 \newcommand*{\@Glsymbol}[2] [] {%
16759 \new@ifnextchar[{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
16760 \def\@Glsymbol@#1#2[#3]{%
16761 \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
16762 }
16763 \newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
16764 \newcommand*{\@GLSsymbol}[2] [] {%
16765 \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
16766 \def\@GLSsymbol@#1#2[#3]{%
16767 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrysymbol{#2}#3}}%
16768 }
16769 \newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
16770 \newcommand*{\@glssymbolplural}[2] [] {%
16771 \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}]
16772 \def\@glssymbolplural@#1#2[#3]{%
16773 \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
16774 }
16775 \newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
16776 \newcommand*{\@Glsymbolplural}[2] [] {%
16777 \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
16778 \def\@Glsymbolplural@#1#2[#3]{%
16779 \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
16780 }
16781 \newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
16782 \newcommand*{\@GLSsymbolplural}[2] [] {%
16783 \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
16784 \def\@GLSsymbolplural@#1#2[#3]{%
16785 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrysymbolplural{#2}#3}}%
16786 }
16787 \newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
16788 \newcommand*{\@glsuseri}[2] [] {%
16789 \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
16790 \def\@glsuseri@#1#2[#3]{%
16791 \@gls@field@link{#1}{#2}{\glstentryuseri{#2}#3}%
16792 }
16793 \newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
16794 \newcommand*{\@Glsuseri}[2] [] {%
16795 \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
16796 \def\@Glsuseri@#1#2[#3]{%
16797 \@gls@field@link{#1}{#2}{\glstentryuseri{#2}#3}%
16798 }
16799 \newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
16800 \newcommand*{\@GLSuseri}[2] [] {%
16801 \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]

```

```

16802 \def\@GLSuseri@#1#2[#3]{%
16803   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}%
16804 }
16805 \newrobustcmd*{\@glsuserii}{\@gls@hyp@opt\@glsuserii}
16806 \newcommand*{\@glsuserii}[2][{}]{%
16807   \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
16808 \def\@glsuserii@#1#2[#3]{%
16809   \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
16810 }
16811 \newrobustcmd*{\@Glsuserii}{\@gls@hyp@opt\@Glsuserii}
16812 \newcommand*{\@Glsuserii}[2][{}]{%
16813   \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
16814 \def\@Glsuserii@#1#2[#3]{%
16815   \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
16816 }
16817 \newrobustcmd*{\@GLSuserii}{\@gls@hyp@opt\@GLSuserii}
16818 \newcommand*{\@GLSuserii}[2][{}]{%
16819   \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}}
16820 \def\@GLSuserii@#1#2[#3]{%
16821   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}%
16822 }
16823 \newrobustcmd*{\@glsuseriii}{\@gls@hyp@opt\@glsuseriii}
16824 \newcommand*{\@glsuseriii}[2][{}]{%
16825   \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}}
16826 \def\@glsuseriii@#1#2[#3]{%
16827   \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
16828 }
16829 \newrobustcmd*{\@Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
16830 \newcommand*{\@Glsuseriii}[2][{}]{%
16831   \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
16832 \def\@Glsuseriii@#1#2[#3]{%
16833   \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
16834 }
16835 \newrobustcmd*{\@GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
16836 \newcommand*{\@GLSuseriii}[2][{}]{%
16837   \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}}
16838 \def\@GLSuseriii@#1#2[#3]{%
16839   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriii{#2}#3}}%
16840 }
16841 \newrobustcmd*{\@glsuseriv}{\@gls@hyp@opt\@glsuseriv}
16842 \newcommand*{\@glsuseriv}[2][{}]{%
16843   \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2}[]}}
16844 \def\@glsuseriv@#1#2[#3]{%
16845   \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
16846 }
16847 \newrobustcmd*{\@Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
16848 \newcommand*{\@Glsuseriv}[2][{}]{%
16849   \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}}
16850 \def\@Glsuseriv@#1#2[#3]{%

```

```

16851 \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
16852 }
16853 \newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
16854 \newcommand*{\@GLSuseriv}[2] [] {%
16855 \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
16856 \def\@GLSuseriv@#1#2[#3]{%
16857 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuseriv{#2}#3}}%
16858 }
16859 \newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
16860 \newcommand*{\@glsuserv}[2] [] {%
16861 \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
16862 \def\@glsuserv@#1#2[#3]{%
16863 \@gls@field@link{#1}{#2}{\glentryuserv{#2}#3}%
16864 }
16865 \newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
16866 \newcommand*{\@Glsuserv}[2] [] {%
16867 \new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}]
16868 \def\@Glsuserv@#1#2[#3]{%
16869 \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
16870 }
16871 \newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
16872 \newcommand*{\@GLSuserv}[2] [] {%
16873 \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
16874 \def\@GLSuserv@#1#2[#3]{%
16875 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuserv{#2}#3}}%
16876 }
16877 \newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
16878 \newcommand*{\@glsuservi}[2] [] {%
16879 \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
16880 \def\@glsuservi@#1#2[#3]{%
16881 \@gls@field@link{#1}{#2}{\glentryuservi{#2}#3}%
16882 }
16883 \newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
16884 \newcommand*{\@Glsuservi}[2] [] {%
16885 \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}]
16886 \def\@Glsuservi@#1#2[#3]{%
16887 \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
16888 }
16889 \newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
16890 \newcommand*{\@GLSuservi}[2] [] {%
16891 \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
16892 \def\@GLSuservi@#1#2[#3]{%
16893 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuservi{#2}#3}}%
16894 }
16895 \newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
16896 \newcommand*{\@ns@acrshort}[2] [] {%
16897 \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
16898 }
16899 \def\@acrshort#1#2[#3]{%

```

```

16900 \glsdoifexists{#2}%
16901 {%
16902   \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16903   \let\glsifplural\@secondoftwo
16904   \let\glsapscase\@firstofthree
16905   \let\glsinsert\@empty
16906   \def\glscustomtext{%
16907     \acronymfont{\glsentryshort{#2}}#3%
16908   }%
16909   \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
16910 }%
16911 \glspostlinkhook
16912 }
16913 \newrobustcmd*{\Acrshort}{\@gls@hyp@opt\@ns@Acrshort}
16914 \newcommand*{\ns@Acrshort}[2][]{%
16915   \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2}[]}%
16916 }
16917 \def\@Acrshort#1#2[#3]{%
16918   \glsdoifexists{#2}%
16919   {%
16920     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16921     \def\glslabel{#2}%
16922     \let\glsifplural\@secondoftwo
16923     \let\glsapscase\@secondofthree
16924     \let\glsinsert\@empty
16925     \def\glscustomtext{%
16926       \acronymfont{\Glsentryshort{#2}}#3%
16927     }%
16928     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
16929   }%
16930   \glspostlinkhook
16931 }
16932 \newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
16933 \newcommand*{\ns@ACRshort}[2][]{%
16934   \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2}[]}%
16935 }
16936 \def\@ACRshort#1#2[#3]{%
16937   \glsdoifexists{#2}%
16938   {%
16939     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16940     \def\glslabel{#2}%
16941     \let\glsifplural\@secondoftwo
16942     \let\glsapscase\@thirdofthree
16943     \let\glsinsert\@empty
16944     \def\glscustomtext{%
16945       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
16946     }%
16947     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
16948   }%

```

```

16949 \glspostlinkhook
16950 }
16951 \newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
16952 \newcommand*{\ns@acrshortpl}[2] [] {%
16953   \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
16954 }
16955 \def\@acrshortpl#1#2[#3]{%
16956   \glsdoifexists{#2}%
16957   {%
16958     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16959     \def\glslabel{#2}%
16960     \let\glsifplural\@firstoftwo
16961     \let\glsapscase\@firstofthree
16962     \let\glsinsert\@empty
16963     \def\glscustomtext{%
16964       \acronymfont{\glsentryshortpl{#2}}#3%
16965     }%
16966     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
16967   }%
16968   \glspostlinkhook
16969 }
16970 \newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
16971 \newcommand*{\ns@Acrshortpl}[2] [] {%
16972   \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
16973 }
16974 \def\@Acrshortpl#1#2[#3]{%
16975   \glsdoifexists{#2}%
16976   {%
16977     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16978     \def\glslabel{#2}%
16979     \let\glsifplural\@firstoftwo
16980     \let\glsapscase\@secondofthree
16981     \let\glsinsert\@empty
16982     \def\glscustomtext{%
16983       \acronymfont{\Glsentryshortpl{#2}}#3%
16984     }%
16985     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
16986   }%
16987   \glspostlinkhook
16988 }
16989 \newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
16990 \newcommand*{\ns@ACRshortpl}[2] [] {%
16991   \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
16992 }
16993 \def\@ACRshortpl#1#2[#3]{%
16994   \glsdoifexists{#2}%
16995   {%
16996     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16997     \def\glslabel{#2}%

```

```

16998 \let\glsifplural\@firstoftwo
16999 \let\glschapscase\@thirdofthree
17000 \let\glsinsert\@empty
17001 \def\glscustomtext{%
17002     \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
17003 }%
17004 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17005 }%
17006 \glspostlinkhook
17007 }
17008 \newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}
17009 \newcommand*{\ns@acrlong}[2] [] {%
17010     \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
17011 }
17012 \def\@acrlong#1#2[#3] {%
17013     \glsdoifexists{#2}%
17014     {%
17015         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17016         \def\glslabel{#2}%
17017         \let\glsifplural\@secondoftwo
17018         \let\glschapscase\@firstofthree
17019         \let\glsinsert\@empty
17020         \def\glscustomtext{%
17021             \glsentrylong{#2}#3%
17022         }%
17023         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17024     }%
17025     \glspostlinkhook
17026 }
17027 \newrobustcmd*{\AcrLong}{\@gls@hyp@opt\@ns@AcrLong}
17028 \newcommand*{\ns@AcrLong}[2] [] {%
17029     \new@ifnextchar[{\@AcrLong{#1}{#2}}{\@AcrLong{#1}{#2} []}%
17030 }
17031 \def\@AcrLong#1#2[#3] {%
17032     \glsdoifexists{#2}%
17033     {%
17034         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17035         \def\glslabel{#2}%
17036         \let\glsifplural\@secondoftwo
17037         \let\glschapscase\@secondofthree
17038         \let\glsinsert\@empty
17039         \def\glscustomtext{%
17040             \Glsentrylong{#2}#3%
17041         }%
17042         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17043     }%
17044     \glspostlinkhook
17045 }
17046 \newrobustcmd*{\ACRLong}{\@gls@hyp@opt\@ns@ACRLong}

```

```

17047 \newcommand*{\ns@ACRlong}[2][{}]{%
17048   \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}]%
17049 }
17050 \def\@ACRlong#1#2[#3]{%
17051   \glsdoifexists{#2}%
17052   {%
17053     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17054     \def\glslabel{#2}%
17055     \let\glsifplural\@secondoftwo
17056     \let\glscapscase\@thirdofthree
17057     \let\glsinsert\@empty
17058     \def\glscustomtext{%
17059       \mfirstucMakeUppercase{\glstrylong{#2}#3}%
17060     }%
17061     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17062   }%
17063   \glspostlinkhook
17064 }
17065 \newrobustcmd*{\acrlongpl}{\@gl@hyp@opt\ns@acrlongpl}
17066 \newcommand*{\ns@acrlongpl}[2][{}]{%
17067   \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}]%
17068 }
17069 \def\@acrlongpl#1#2[#3]{%
17070   \glsdoifexists{#2}%
17071   {%
17072     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17073     \def\glslabel{#2}%
17074     \let\glsifplural\@firstoftwo
17075     \let\glscapscase\@firstofthree
17076     \let\glsinsert\@empty
17077     \def\glscustomtext{%
17078       \glstrylongpl{#2}#3%
17079     }%
17080     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17081   }%
17082   \glspostlinkhook
17083 }
17084 \newrobustcmd*{\Acrlongpl}{\@gl@hyp@opt\ns@Acrlongpl}
17085 \newcommand*{\ns@Acrlongpl}[2][{}]{%
17086   \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}]%
17087 }
17088 \def\@Acrlongpl#1#2[#3]{%
17089   \glsdoifexists{#2}%
17090   {%
17091     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17092     \def\glslabel{#2}%
17093     \let\glsifplural\@firstoftwo
17094     \let\glscapscase\@secondofthree
17095     \let\glsinsert\@empty

```

```

17096 \def\glscustomtext{%
17097 \Glsentrylongpl{#2}#3%
17098 }%
17099 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17100 }%
17101 \glspostlinkhook
17102 }
17103 \newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
17104 \newcommand*{\ns@ACRlongpl}[2][{}]{%
17105 \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
17106 }
17107 \def\@ACRlongpl#1#2[#3]{%
17108 \glsdoifexists{#2}%
17109 {%
17110 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17111 \def\glslabel{#2}%
17112 \let\glsifplural\@firstoftwo
17113 \let\glscapscase\@thirdofthree
17114 \let\glsinsert\@empty
17115 \def\glscustomtext{%
17116 \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
17117 }%
17118 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17119 }%
17120 \glspostlinkhook
17121 }
17122 \newcommand*{\@gls@entry@field}[2]{%
17123 \csname glo@\glsdetoklabel{#1}@#2\endcsname
17124 }
17125 \newcommand*{\glsletentryfield}[3]{%
17126 \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
17127 }
17128 \newcommand*{\@Gls@entry@field}[2]{%
17129 \glsdoifexistsordo{#1}%
17130 {%
17131 \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
17132 \ifdef\@glo@text
17133 {%
17134 \xmakefirstuc{\@glo@text}%
17135 }%
17136 }%
17137 ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
17138 entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
17139 label and the field name}%
17140 }%
17141 }%
17142 {%
17143 ???%
17144 }%

```



```

17145 }
17146 \newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
17147 \newrobustcmd*{\Glsentryname}[1]{%
17148   \@Gls@entryname{#1}%
17149 }
17150 \newcommand*{\@Gls@entryname}[1]{%
17151   \@Gls@entry@field{#1}{name}%
17152 }
17153 \newcommand*{\@Gls@acentryname}[1]{%
17154   \ifglshaslong{#1}%
17155   {%
17156     \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
17157     \expandafter\@gls@getbody\@glo@text{}\@nil
17158     \expandafter\ifx\@gls@body\glsentrylong\relax
17159       \expandafter\Glsentrylong\@gls@rest
17160     \else
17161       \expandafter\ifx\@gls@body\glsentryshort\relax
17162         \expandafter\Glsentryshort\@gls@rest
17163       \else
17164         \expandafter\ifx\@gls@body\acronymfont\relax
17165         {%
17166           \let\glsentryshort\Glsentryshort
17167           \@glo@text
17168         }%
17169       \else
17170         \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
17171         {%
17172           \let\glsentryshort\Glsentryshort
17173           \@glo@text
17174         }%
17175       \else
17176         \xmakefirstuc{\@glo@text}%
17177       \fi
17178     \fi
17179   \fi
17180 }%
17181 }%
17182 {%
17183   \@Gls@entry@field{#1}{name}%
17184 }%
17185 }
17186 \newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
17187 \newrobustcmd*{\Glsentrydesc}[1]{%
17188   \@Gls@entry@field{#1}{desc}%
17189 }
17190 \newcommand*{\glsentrydescplural}[1]{%
17191   \@gls@entry@field{#1}{descplural}%
17192 }
17193 \newrobustcmd*{\Glsentrydescplural}[1]{%

```

```

17194 \@Gls@entry@field{#1}{descplural}%
17195 }
17196 \newcommand*{\glentrytext}[1]{\@Gls@entry@field{#1}{text}}
17197 \newrobustcmd*{\Glentrytext}[1]{%
17198   \@Gls@entry@field{#1}{text}%
17199 }
17200 \newcommand*{\glentryplural}[1]{%
17201   \@Gls@entry@field{#1}{plural}%
17202 }
17203 \newrobustcmd*{\Glentryplural}[1]{%
17204   \@Gls@entry@field{#1}{plural}%
17205 }
17206 \newcommand*{\glentrysymbol}[1]{%
17207   \@Gls@entry@field{#1}{symbol}%
17208 }
17209 \newrobustcmd*{\Glentrysymbol}[1]{%
17210   \@Gls@entry@field{#1}{symbol}%
17211 }
17212 \newcommand*{\glentrysymbolplural}[1]{%
17213   \@Gls@entry@field{#1}{symbolplural}%
17214 }
17215 \newrobustcmd*{\Glentrysymbolplural}[1]{%
17216   \@Gls@entry@field{#1}{symbolplural}%
17217 }
17218 \newcommand*{\glentryfirst}[1]{%
17219   \@Gls@entry@field{#1}{first}%
17220 }
17221 \newrobustcmd*{\Glentryfirst}[1]{%
17222   \@Gls@entry@field{#1}{first}%
17223 }
17224 \newcommand*{\glentryfirstplural}[1]{%
17225   \@Gls@entry@field{#1}{firstpl}%
17226 }
17227 \newrobustcmd*{\Glentryfirstplural}[1]{%
17228   \@Gls@entry@field{#1}{firstpl}%
17229 }
17230 \newrobustcmd*{\@glentrytitlecase}[2]{%
17231   \glstoifexists{#1}%
17232   {%
17233     \glsofieldfetch{#1}{#2}{\@Gls@value}%
17234     \xcapitalisewords{\@Gls@value}%
17235   }%
17236 }
17237 \ifdef\texorpdfstring
17238 {
17239   \newcommand*{\glentrytitlecase}[2]{%
17240     \texorpdfstring
17241       {\@glentrytitlecase{#1}{#2}}%
17242       {\@Gls@entry@field{#1}{#2}}%

```

```

17243 }
17244 }
17245 {
17246 \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
17247 }
17248 \newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
17249 \newcommand*{\glsentrysort}[1]{%
17250 \@gls@entry@field{#1}{sort}}%
17251 }
17252 \newcommand*{\glsentryparent}[1]{%
17253 \@gls@entry@field{#1}{parent}}%
17254 }
17255 \newcommand*{\glsentryuseri}[1]{%
17256 \@gls@entry@field{#1}{useri}}%
17257 }
17258 \newrobustcmd*{\Glsentryuseri}[1]{%
17259 \@Gls@entry@field{#1}{useri}}%
17260 }
17261 \newcommand*{\glsentryuserii}[1]{%
17262 \@gls@entry@field{#1}{userii}}%
17263 }
17264 \newrobustcmd*{\Glsentryuserii}[1]{%
17265 \@Gls@entry@field{#1}{userii}}%
17266 }
17267 \newcommand*{\glsentryuseriii}[1]{%
17268 \@gls@entry@field{#1}{useriii}}%
17269 }
17270 \newrobustcmd*{\Glsentryuseriii}[1]{%
17271 \@Gls@entry@field{#1}{useriii}}%
17272 }
17273 \newcommand*{\glsentryuseriv}[1]{%
17274 \@gls@entry@field{#1}{useriv}}%
17275 }
17276 \newrobustcmd*{\Glsentryuseriv}[1]{%
17277 \@Gls@entry@field{#1}{useriv}}%
17278 }
17279 \newcommand*{\glsentryuserv}[1]{%
17280 \@gls@entry@field{#1}{userv}}%
17281 }
17282 \newrobustcmd*{\Glsentryuserv}[1]{%
17283 \@Gls@entry@field{#1}{userv}}%
17284 }
17285 \newcommand*{\glsentryuservi}[1]{%
17286 \@gls@entry@field{#1}{uservi}}%
17287 }
17288 \newrobustcmd*{\Glsentryuservi}[1]{%
17289 \@Gls@entry@field{#1}{uservi}}%
17290 }
17291 \newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}

```

```

17292 \newrobustcmd*{\Glsentryshort}[1]{%
17293   \@Gls@entry@field{#1}{short}}%
17294 }
17295 \newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
17296 \newrobustcmd*{\Glsentryshortpl}[1]{%
17297   \@Gls@entry@field{#1}{shortpl}}%
17298 }
17299 \newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
17300 \newrobustcmd*{\Glsentrylong}[1]{%
17301   \@Gls@entry@field{#1}{long}}%
17302 }
17303 \newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
17304 \newrobustcmd*{\Glsentrylongpl}[1]{%
17305   \@Gls@entry@field{#1}{longpl}}%
17306 }
17307 \newcommand*{\glsentryfull}[1]{%
17308   \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
17309 }
17310 \newrobustcmd*{\Glsentryfull}[1]{%
17311   \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
17312 }
17313 \newcommand*{\glsentryfullpl}[1]{%
17314   \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
17315 }
17316 \newrobustcmd*{\Glsentryfullpl}[1]{%
17317   \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
17318 }
17319 \newcommand*{\glsentrynumberlist}[1]{%
17320   \glsdoifexists{#1}%
17321   {%
17322     \@gls@entry@field{#1}{numberlist}%
17323   }%
17324 }
17325 \@ifpackageloaded{hyperref} {%
17326   \newcommand*{\glsdisplaynumberlist}[1]{%
17327     \GlossariesWarning
17328     {%
17329       \string\glsdisplaynumberlist\space
17330       doesn't work with hyperref.^^JUsing
17331       \string\glsentrynumberlist\space instead%
17332     }%
17333     \glsentrynumberlist{#1}%
17334   }%
17335 }%
17336 {%
17337   \newcommand*{\glsdisplaynumberlist}[1]{%
17338     \glsdoifexists{#1}%
17339     {%
17340       \bgroup

```



```

17390 }
17391 \newcommand*{\@gls@adjustmode}{%
17392 \AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
17393 \newrobustcmd*{\glsaddall}[1][1]{%
17394   \edef\@glo@type{\@glo@types}%
17395   \setkeys{glossadd}{#1}%
17396   \forallglsentries[\@glo@type]{\@glo@entry}{%
17397     \glsadd[#1]{\@glo@entry}%
17398   }%
17399 }
17400 \newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
17401   \forallglsentries[#1]{\@glo@entry}%
17402   {%
17403     \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
17404   }%
17405 }
17406 \newcommand*{\glsignore}[1]{%
17407   \edef\glsopenbrace{\expandafter\@gobble\string\{ }
17408   \edef\glsclosebrace{\expandafter\@gobble\string\} }
17409   \edef\glsbackslash{\expandafter\@gobble\string\ }
17410   \edef\glsquote#1{\string"#1\string"}
17411   \edef\glspercentchar{\expandafter\@gobble\string\% }
17412   \edef\glstildechar{\string~ }
17413   \ifglxsindy
17414     \newcommand*{\@glsfirstletter}{A}
17415 \fi
17416 \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
17417   \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
17418 \ifglxsindy
17419   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
17420     \renewcommand*{\@glsfirstletter}{#1}}
17421   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
17422     \renewcommand*{\@glsfirstletter}{#1}%
17423     \@onelevel@sanitize\@glsfirstletter
17424   }
17425 \else
17426   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
17427     \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
17428   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
17429     \@GlsSetXdyFirstLetterAfterDigits
17430   }
17431 \fi
17432 \ifglxsindy
17433   \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
17434 \fi
17435 \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
17436   \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
17437 }
17438 \ifglxsindy

```

```

17439 \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
17440   \renewcommand*{\@xdynumbergrouporder}{#1}%
17441 }
17442 \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
17443   \renewcommand*{\@xdynumbergrouporder}{#1}%
17444   \@onelevel@sanitize\@xdynumbergrouporder
17445 }
17446 \else
17447   \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
17448     \glsnoindywarning\GlsSetXdyNumberGroupOrder}
17449   \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
17450     \@GlsSetXdyNumberGroupOrder}
17451 \fi
17452 \newcommand*{\@glsminrange}{2}
17453 \ifglsxindy
17454   \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
17455     \renewcommand*{\@glsminrange}{#1}}
17456 \else
17457   \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
17458     \glsnoindywarning\GlsSetXdyMinRangeLength}
17459 \fi
17460
17461 \ifglsxindy
17462   \def\writeist{%
17463     \ifundef{\glswrite}{\newwrite\glswrite}{}%
17464     \@gls@addpredefinedattributes
17465     \openout\glswrite=\istfilename
17466     \write\glswrite{;; xindy style file created by the glossaries
17467       package}%
17468     \write\glswrite{;; for document '\jobname' on
17469       \the\year-\the\month-\the\day}%
17470     \write\glswrite{^^J; required styles^^J}
17471     \@for\@xdystyle:=\@xdyrequiredstyles\do{%
17472       \ifx\@xdystyle\@empty
17473       \else
17474         \protected@write\glswrite{}{(require
17475           \string"\@xdystyle.xdy\string")}%
17476       \fi
17477     }%
17478     \write\glswrite{^^J%
17479       ; list of allowed attributes (number formats)^^J}%
17480     \write\glswrite{(define-attributes ((\@xdyattributes)))}%
17481     \write\glswrite{^^J; user defined alphabets^^J}%
17482     \write\glswrite{\@xdyuseralphabets}%
17483     \write\glswrite{^^J; location class definitions^^J}%
17484     \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
17485       \protected@write\glswrite{}{(define-location-class
17486         \string"\@gls@classI\string"^^J\space\space\space
17487         (

```

```

17488         :sep "{{"
17489         \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
17490         :sep "}"
17491     )
17492     ^^J\space\space\space
17493     :min-range-length \@glsminrange^^J%
17494 )
17495 }%
17496 {%
17497     \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
17498         \protected@write\glswrite{ }(define-location-class
17499         \string"\@gls@classII-\@gls@classI\string"
17500         ^^J\space\space\space
17501         (
17502             :sep "{"
17503             \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
17504             :sep "{{"
17505             \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
17506             :sep "}"
17507         )
17508         ^^J\space\space\space
17509         :min-range-length \@glsminrange^^J%
17510     )
17511 }%
17512 }%
17513 }%
17514 }%
17515 \write\glswrite{^^J; user defined location classes}%
17516 \write\glswrite{\@xdyuserlocationdefs}%
17517 \write\glswrite{^^J; define cross-reference class^^J}%
17518 \write\glswrite{(define-crossref-class \string"see\string"
17519     :unverified )}%
17520 \write\glswrite{(markup-crossref-list
17521     :class \string"see\string"^^J\space\space\space
17522     :open \string"\string\glsseeformat\string"
17523     :close \string"{ }\string")}%
17524 \@xdycrossrefhook
17525 \write\glswrite{^^J; define the order of the location classes}%
17526 \write\glswrite{(define-location-class-order
17527     (\@xdylocationclassorder))}%
17528 \write\glswrite{^^J; define the glossary markup^^J}%
17529 \write\glswrite{(markup-index^^J\space\space\space
17530     :open \string"\string
17531     \glossarysection[\string\glossarytoctitle]{\string
17532     \glossarytitle}\string\glossarypreamble}%
17533 \@for\@this@ctr:=\@xdycounters\do{%
17534     {%
17535         \@for\@this@attr:=\@xdyattributelist\do{%
17536             \protected@write\glswrite{ }\string\providecommand*%

```



```

17537         \expandafter\string
17538         \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
17539         {%
17540             \string\setentrycounter
17541             [\expandafter\@gobble\string\#1]{\@this@ctr}%
17542             \expandafter\string
17543             \csname\@this@attr\endcsname
17544             {\expandafter\@gobble\string\#2}%
17545         }%
17546     }%
17547 }%
17548 }%
17549 }%
17550 \write\glswrite{%
17551     \string\begin
17552     {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
17553     \space\space:close \string"\glspercentchar\glstildechar n\string
17554     \end{theglossary}\string\glossarypostamble
17555     \glstildechar n\string" ^^J\space\space\space
17556     :tree)}}%
17557 \write\glswrite{(markup-letter-group-list
17558     :sep \string"\string\glsgroupskip\glstildechar n\string"))%
17559 \write\glswrite{(markup-indexentry
17560     :open \string"\string\relax \string\glsresetentrylist
17561     \glstildechar n\string"))%
17562 \write\glswrite{(markup-locclass-list :open
17563     \string"\glsopenbrace\string\glossaryentrynumbers
17564     \glsopenbrace\string\relax\space \string"^^J\space\space\space
17565     :sep \string", \string"
17566     :close \string"\glsclosebrace\glsclosebrace\string"))%
17567 \write\glswrite{(markup-locref-list
17568     :sep \string"\string\delimN\space\string"))%
17569 \write\glswrite{(markup-range
17570     :sep \string"\string\delimR\space\string"))%
17571 \@onelevel@sanitize\gls@suffixF
17572 \@onelevel@sanitize\gls@suffixFF
17573 \ifx\gls@suffixF\@empty
17574 \else
17575     \write\glswrite{(markup-range
17576         :close "\gls@suffixF" :length 1 :ignore-end)}}%
17577 \fi
17578 \ifx\gls@suffixFF\@empty
17579 \else
17580     \write\glswrite{(markup-range
17581         :close "\gls@suffixFF" :length 2 :ignore-end)}}%
17582 \fi
17583 \write\glswrite{^^J; define format to use for locations^^J}%
17584 \write\glswrite{\@xdylocref}%
17585 \write\glswrite{^^J; define letter group list format^^J}%

```

```

17586 \write\glswrite{(markup-letter-group-list
17587 :sep \string"\string\glsgroupskip\glstildechar n\string")}%
17588 \write\glswrite{^^J; letter group headings^^J}%
17589 \write\glswrite{(markup-letter-group
17590 :open-head \string"\string\glsgroupheading
17591 \glsoopenbrace\string"^^J\space\space\space
17592 :close-head \string"\glsclosebrace\string")}%
17593 \write\glswrite{^^J; additional letter groups^^J}%
17594 \write\glswrite{@\xdylettergroups}%
17595 \write\glswrite{^^J; additional sort rules^^J}
17596 \write\glswrite{@\xdysortrules}%
17597 \@gls@writeisthook
17598 \closeout\glswrite
17599 \let\writeist\relax
17600 }
17601 \else
17602 \edef\@gls@actualchar{\string?}
17603 \edef\@gls@encapchar{\string|}
17604 \edef\@gls@levelchar{\string!}
17605 \edef\@gls@quotechar{\string"}%
17606 \let\GlsSetQuote\gls@nosetquote
17607 \def\writeist{\relax
17608 \ifundef{\glswrite}{\newwrite\glswrite}{\relax
17609 \openout\glswrite=\istfilename
17610 \write\glswrite{\glspersentchar\space makeindex style file
17611 created by the glossaries package}
17612 \write\glswrite{\glspersentchar\space for document
17613 'jobname' on \the\year-\the\month-\the\day}
17614 \write\glswrite{actual '@gls@actualchar'}
17615 \write\glswrite{encap '@gls@encapchar'}
17616 \write\glswrite{level '@gls@levelchar'}
17617 \write\glswrite{quote '@gls@quotechar'}
17618 \write\glswrite{keyword \string"\string\glossaryentry\string"}
17619 \write\glswrite{preamble \string"\string\glossarysection[\string
17620 \glossarytoctitle]{\string\glossarytitle}\string
17621 \glossarypreamble\string\n\string\begin{theglossary}\string
17622 \glossaryheader\string\n\string"}
17623 \write\glswrite{postamble \string"\string%\string\n\string
17624 \end{theglossary}\string\glossarypostamble\string\n
17625 \string"}
17626 \write\glswrite{group_skip \string"\string\glsgroupskip\string\n
17627 \string"}
17628 \write\glswrite{item_0 \string"\string%\string\n\string"}
17629 \write\glswrite{item_1 \string"\string%\string\n\string"}
17630 \write\glswrite{item_2 \string"\string%\string\n\string"}
17631 \write\glswrite{item_01 \string"\string%\string\n\string"}
17632 \write\glswrite{item_x1
17633 \string"\string\relax \string\glsresetentrylist\string\n
17634 \string"}

```

```

17635 \write\glswrite{item_12 \string"\string%\string\n\string"}
17636 \write\glswrite{item_x2
17637 \string"\string\relax \string\glresetentrylist\string\n
17638 \string"}
17639 \write\glswrite{delim_0 \string"\string\{\string
17640 \glossaryentrynumbers\string\{\string\relax \string"}
17641 \write\glswrite{delim_1 \string"\string\{\string
17642 \glossaryentrynumbers\string\{\string\relax \string"}
17643 \write\glswrite{delim_2 \string"\string\{\string
17644 \glossaryentrynumbers\string\{\string\relax \string"}
17645 \write\glswrite{delim_t \string"\string\}\string\}\string"}
17646 \write\glswrite{delim_n \string"\string\delimN \string"}
17647 \write\glswrite{delim_r \string"\string\delimR \string"}
17648 \write\glswrite{headings_flag 1}
17649 \write\glswrite{heading_prefix
17650 \string"\string\glsgroupheading\string\{\string"}
17651 \write\glswrite{heading_suffix
17652 \string"\string\}\string\relax
17653 \string\glresetentrylist \string"}
17654 \write\glswrite{symhead_positive \string"glssymbols\string"}
17655 \write\glswrite{numhead_positive \string"glnumbers\string"}
17656 \write\glswrite{page_compositor \string"glscpositor\string"}
17657 \@gls@escbsdq\gls@suffixF
17658 \@gls@escbsdq\gls@suffixFF
17659 \ifx\gls@suffixF\@empty
17660 \else
17661 \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
17662 \fi
17663 \ifx\gls@suffixFF\@empty
17664 \else
17665 \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
17666 \fi
17667 \@gls@writeisthook
17668 \closeout\glswrite
17669 \let\writeist\relax
17670 }
17671 \fi
17672 \newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
17673 \@onlypremakeg\GlsSetWriteIstHook
17674 \newcommand*\@gls@writeisthook}{
17675 \ifglxindy
17676 \newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
17677 \newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
17678 \else
17679 \newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
17680 \@ifpackageloaded{tracklang}%
17681 {%
17682 \IfTrackedLanguage{german}%
17683 {%

```

```

17684     \def\@gls@extramakeindexopts{-g}%
17685 }%
17686 {}%
17687 }%
17688 {}%
17689 \edef\@gls@docheckquotedef{%
17690   \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
17691     \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
17692     \noexpand\toks@={####1}%
17693     \noexpand\ifx\noexpand\null####2\noexpand\null
17694     \noexpand\ifx\noexpand\null####3\noexpand\null
17695     \noexpand\edef\noexpand\@gls@checkedmkidx{%
17696       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
17697     \noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
17698   \noexpand\else
17699     \noexpand\edef\noexpand\@gls@checkedmkidx{%
17700       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17701       \noexpand\@gls@quotechar\noexpand\@gls@quotechar
17702       \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
17703     \noexpand\def\noexpand\@gls@checkquote{%
17704       \noexpand\@gls@checkquote####3\noexpand\null}%
17705   \noexpand\fi
17706 \noexpand\else
17707   \noexpand\edef\noexpand\@gls@checkedmkidx{%
17708     \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17709     \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
17710   \noexpand\ifx\noexpand\null####3\noexpand\null
17711     \noexpand\def\noexpand\@gls@checkquote{%
17712       \noexpand\@gls@checkquote####2#1#1\noexpand\null}%
17713   \noexpand\else
17714     \noexpand\def\noexpand\@gls@checkquote{%
17715       \noexpand\@gls@checkquote####2#1####3\noexpand\null}%
17716   \noexpand\fi
17717 \noexpand\fi
17718 \noexpand\@gls@checkquote
17719 }%
17720 }%
17721 \@gls@docheckquotedef
17722 \edef\@gls@docheckquotedef{%
17723   \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
17724     \noexpand\def\noexpand\@gls@checkedmkidx{%
17725       \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
17726       #1#1\noexpand\null
17727       \noexpand\expandafter\noexpand\@gls@updatechecked
17728       \noexpand\@gls@checkedmkidx{####1}%
17729     \noexpand\def\noexpand\@gls@checkedmkidx{%
17730       \noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
17731       \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
17732       \noexpand\null

```

```

17733 \noexpand\expandafter\noexpand\@gls@updatechecked
17734 \noexpand\@gls@checkedmkidx{####1}%
17735 \noexpand\def\noexpand\@gls@checkedmkidx{%
17736 \noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
17737 \noexpand\?\noexpand\?\noexpand\null
17738 \noexpand\expandafter\noexpand\@gls@updatechecked
17739 \noexpand\@gls@checkedmkidx{####1}%
17740 \noexpand\def\noexpand\@gls@checkedmkidx{%
17741 \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
17742 \noexpand?\noexpand?\noexpand\null
17743 \noexpand\expandafter\noexpand\@gls@updatechecked
17744 \noexpand\@gls@checkedmkidx{####1}%
17745 \noexpand\def\noexpand\@gls@checkedmkidx{%
17746 \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
17747 \noexpand|\noexpand|\noexpand\null
17748 \noexpand\expandafter\noexpand\@gls@updatechecked
17749 \noexpand\@gls@checkedmkidx{####1}%
17750 \noexpand\def\noexpand\@gls@checkedmkidx{%
17751 \noexpand\expandafter\noexpand\@gls@checkesbar####1\noexpand\@nil
17752 \noexpand||\noexpand||\noexpand\null
17753 \noexpand\expandafter\noexpand\@gls@updatechecked
17754 \noexpand\@gls@checkedmkidx{####1}%
17755 \noexpand\def\noexpand\@gls@checkedmkidx{%
17756 \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
17757 \noexpand!\noexpand!\noexpand\null
17758 \noexpand\expandafter\noexpand\@gls@updatechecked
17759 \noexpand\@gls@checkedmkidx{####1}%
17760 }%
17761 }%
17762 \@gls@docheckquotedef
17763 \edef\@gls@docheckquotedef{%
17764 \noexpand\def\noexpand\@gls@checkescquote####1%
17765 \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
17766 ####3\noexpand\null{%
17767 \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
17768 \noexpand\toks@={####1}%
17769 \noexpand\ifx\noexpand\null####2\noexpand\null
17770 \noexpand\ifx\noexpand\null####3\noexpand\null
17771 \noexpand\edef\noexpand\@gls@checkedmkidx{%
17772 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
17773 \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
17774 \noexpand\else
17775 \noexpand\edef\noexpand\@gls@checkedmkidx{%
17776 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17777 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
17778 \csname#1\endcsname}\noexpand\@gls@quotechar
17779 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
17780 \csname#1\endcsname}\noexpand\@gls@quotechar}%
17781 \noexpand\def\noexpand\@gls@checkescquote{%

```

```

17782         \noexpand\@gls@checkescquote####3\noexpand\null}%
17783     \noexpand\fi
17784     \noexpand\else
17785     \noexpand\edef\noexpand\@gls@checkedmkidx{%
17786         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17787         \noexpand\@gls@quotearch\noexpand\string
17788         \expandonce{\csname#1\endcsname}\noexpand\@gls@quotearch}%
17789     \noexpand\ifx\noexpand\null####3\noexpand\null
17790     \noexpand\def\noexpand\@gls@checkescquote{%
17791         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
17792         \expandonce{\csname#1\endcsname}\noexpand\null}%
17793     \noexpand\else
17794     \noexpand\def\noexpand\@gls@checkescquote{%
17795         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
17796         ####3\noexpand\null}%
17797     \noexpand\fi
17798     \noexpand\fi
17799     \noexpand\@gls@checkescquote
17800 }%
17801 }%
17802 \@gls@docheckquotedef
17803 }
17804 \newcommand*{\@gls@nosetquote}[1]{\PackageError{glossaries}%
17805     {\string\GlsSetQuote\space not permitted here}%
17806     {\Move \string\GlsSetQuote\space earlier in the preamble, as
17807     soon as possible after glossaries.sty has been loaded}}
17808 \fi
17809 \newcommand*{\@gls@extramakeindexopts}[1]{%
17810 \newcommand{\noist}{%
17811     \@gls@addpredefinedattributes
17812     \let\writeist\relax
17813 }
17814 \newcommand*{\@makeglossary}[1]{%
17815     \ifglossaryexists{#1}%
17816     {%
17817         \ifglssavewrites
17818         \expandafter\newtoks\csname glo@#1@filetok\endcsname
17819         \else
17820         \expandafter\newwrite\csname glo@#1@file\endcsname
17821         \expandafter\@gls@openfile\csname glo@#1@file\endcsname{#1}%
17822         \fi
17823         \@gls@renewglossary
17824         \writeist
17825     }%
17826     {%
17827         \PackageError{glossaries}%
17828         {Glossary type ‘#1’ not defined}%
17829         {New glossaries must be defined before using \string\makeglossaries}%
17830     }%

```

```

17831 }
17832 \newcommand*{\@glsopenfile}[2]{%
17833   \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
17834   \PackageInfo{glossaries}{Writing glossary file
17835     \jobname.\csname @glotype@#2@out\endcsname}%
17836 }
17837 \newcommand*{\@closegls}[1]{%
17838   \closeout\csname glo@#1@file\endcsname
17839 }
17840 \ifglxindy
17841   \newcommand*{\@gls@automake}[1]{%
17842     \ifglossaryexists{#1}
17843     {%
17844       \@closegls{#1}%
17845       \ifdefstring{\glsorder}{letter}%
17846       {\def\@gls@order{-M ord/letorder }}%
17847       {\let\@gls@order\@empty}%
17848       \ifcsundef{\xdy@#1@language}%
17849       {\let\@gls@langmod\@xdy@main@language}%
17850       {\letcs\@gls@langmod{\xdy@#1@language}}%
17851       \edef\@gls@dothiswrite{\noexpand\write18{xindy
17852         -I xindy
17853         \@gls@order
17854         -L \@gls@langmod\space
17855         -M \@gls@istfilebase\space
17856         -C \@gls@codepage\space
17857         -t \jobname.\csuse{@glotype@#1@log}
17858         -o \jobname.\csuse{@glotype@#1@in}
17859         \jobname.\csuse{@glotype@#1@out}}}%
17860     }%
17861     \@gls@dothiswrite
17862   }%
17863   {%
17864     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17865   }%
17866 }
17867 \else
17868   \newcommand*{\@gls@automake}[1]{%
17869     \ifglossaryexists{#1}
17870     {%
17871       \@closegls{#1}%
17872       \ifdefstring{\glsorder}{letter}%
17873       {\def\@gls@order{-l }}%
17874       {\let\@gls@order\@empty}%
17875       \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
17876         -s \istfilename\space
17877         -t \jobname.\csuse{@glotype@#1@log}
17878         -o \jobname.\csuse{@glotype@#1@in}
17879         \jobname.\csuse{@glotype@#1@out}}}%

```

```

17880     }%
17881     \@gls@dothiswrite
17882 }%
17883 {%
17884     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17885 }%
17886 }
17887 \fi
17888 \ifglsxindy
17889 \newcommand*{\@gls@automake@immediate}[1]{%
17890     \ifglossaryexists{#1}
17891     {%
17892         \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
17893             {%
17894                 \ifdefstring{\glsorder}{letter}%
17895                 {\def\@gls@order{-M ord/letorder }}%
17896                 {\let\@gls@order\empty}%
17897                 \ifcsundef{xdy@#1@language}%
17898                 {\let\@gls@langmod\@xdy@main@language}%
17899                 {\letcs\@gls@langmod{\@xdy@#1@language}}%
17900                 \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
17901                     -I xindy
17902                     \@gls@order
17903                     -L \@gls@langmod\space
17904                     -M \gls@istfilebase\space
17905                     -C \gls@codepage\space
17906                     -t \jobname.\csuse{@glotype@#1@log}
17907                     -o \jobname.\csuse{@glotype@#1@in}
17908                     \jobname.\csuse{@glotype@#1@out}}}%
17909             }%
17910             \@gls@dothiswrite
17911         }%
17912         {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
17913             doesn't exist. Rerun may be required}}%
17914     }%
17915     {%
17916         \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17917     }%
17918 }
17919 \else
17920 \newcommand*{\@gls@automake@immediate}[1]{%
17921     \ifglossaryexists{#1}
17922     {%
17923         \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
17924             {%
17925                 \ifdefstring{\glsorder}{letter}%
17926                 {\def\@gls@order{-l }}%
17927                 {\let\@gls@order\empty}%
17928                 \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order

```



```

17929         -s \istfilename\space
17930         -t \jobname.\csuse{@glotype@#1@log}
17931         -o \jobname.\csuse{@glotype@#1@in}
17932         \jobname.\csuse{@glotype@#1@out}}}%
17933     }%
17934     \@gls@dothiswrite
17935 }%
17936 {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
17937     doesn't exist. Rerun may be required}}}%
17938 }%
17939 {%
17940     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17941 }%
17942 }
17943 \fi
17944 \newcommand*{\@warn@nomakeglossaries}{}
17945 \newcommand*{\@warn@nomakeglossaries}{\@warn@nomakeglossaries}
17946 \newcommand{\@gls@@automake@immediate}{%
17947     \ifnum\gls@automake@nr=2\relax
17948         \@for\@gls@type:=\@glo@types\do{%
17949             \ifdefempty{\@gls@type}{}%
17950             {\@gls@automake@immediate{\@gls@type}}}%
17951         }%
17952         \glsautomakefalse
17953         \renewcommand*{\@gls@doautomake}{}%
17954     \fi
17955 }
17956 \newcommand*{\makeglossaries}{%
17957     \@domakeglossaries
17958     {%
17959         \@gls@@automake@immediate
17960         \ifundef{\glswrite}{\newwrite\glswrite}{}%
17961         \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
17962         \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
17963         \ifundef\@gls@extramakeindexopts
17964         {}%
17965         {%
17966             \protected@write\@auxout{}{\string\providecommand
17967                 \string\@gls@extramakeindexopts[1]{} }
17968             \protected@write\@auxout{}{\string\@gls@extramakeindexopts
17969                 {\@gls@extramakeindexopts}}}%
17970         }%
17971         \protected@write\@auxout{}{\string\@istfilename{\istfilename}}}%
17972         \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
17973         \@for\@glo@type:=\@glo@types\do{%
17974             \ifthenelse{\equal{\@glo@type}{} }{}{%
17975                 \@makeglossary{\@glo@type}}}%
17976         }%
17977         \renewcommand*{\newglossary[4] []}{%

```

```

17978 \PackageError{glossaries}{New glossaries
17979 must be created before \string\makeglossaries}{You need
17980 to move \string\makeglossaries\space after all your
17981 \string\newglossary\space commands}}}%
17982 \let\@makeglossary\@gobble
17983 \let\makeglossaries\relax
17984 \@disable@onlypremakeg
17985 \let\gls@checkseeallowed\relax
17986 \let\warn@nomakeglossaries\relax
17987 \def\warn@noprintglossary{%
17988   \ifdefstring{\@glo@types}{,}%
17989   {%
17990     \GlossariesWarningNoLine{No glossaries have been defined}%
17991   }%
17992   {%
17993     \GlossariesWarningNoLine{No \string\printglossary\space
17994       or \string\printglossaries\space
17995       found. ^^J(Remove \string\makeglossaries\space if you
17996       don't want any glossaries.) ^^JThis document will not
17997       have a glossary}%
17998   }%
17999 }%
18000 \ifglssavenumberlist
18001   \edef\@gls@dodolistparser{\noexpand\DeclareListParser
18002     {\noexpand\glsnumlistparser}{\delimN}}}%
18003   \@gls@dodolistparser
18004 \fi
18005 \let\makenoidxglossaries\@no@makeglossaries
18006 \renewcommand*{\@printgloss@setsort}{%
18007   \let\@glo@assign@sortkey\@glo@no@assign@sortkey
18008 }%
18009 \ifglssautomake
18010   \renewcommand*{\@gls@doautomake}{%
18011     \@for\@gls@type:=\@glo@types\do{%
18012       \ifdefempty{\@gls@type}{}%
18013       {\@gls@automake{\@gls@type}}%
18014     }%
18015   }%
18016 \fi
18017 \@glo@check@sortallowed\makeglossaries
18018 }%
18019 }
18020 \@onlypreamble{\makeglossaries}
18021 \AtEndDocument{%
18022   \warn@nomakeglossaries
18023   \warn@noprintglossary
18024 }
18025 \newcommand*{\makenoidxglossaries}{%
18026   \@domakeglossaries

```

```

18027 {%
18028 \renewcommand{\@gls@noref@warn}[1]{%
18029 \GlossariesWarning{Empty glossary for
18030 \string\printnoidxglossary[type={##1}].
18031 Rerun may be required (or you may have forgotten to use
18032 commands like \string\gls)}}%
18033 }%
18034 \let\@gls@checkmkidxchars\@gobble
18035 \glsclocationsfalse
18036 \let\@do@@wrglossary\gls@noidxglossary
18037 \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
18038 \let\gls@checkseeallowed\relax
18039 \renewcommand{\@do@seeglossary}[2]{%
18040 \edef\@gls@label{\glsdetoklabel{##1}}%
18041 \protected@write\@auxout{}{%
18042 \string\@gls@reference
18043 {\csname glo@\@gls@label @type\endcsname}%
18044 {\@gls@label}%
18045 {%
18046 \string\glsseeformat##2}%
18047 }%
18048 }%
18049 }%
18050 \AtBeginDocument
18051 {%
18052 \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
18053 }%
18054 \def\warn@noprintglossary{%
18055 \GlossariesWarningNoLine{No \string\printnoidxglossary\space
18056 or \string\printnoidxglossaries ^^J
18057 found. (Remove \string\makenoidxglossaries\space if you
18058 don't want any glossaries.)^^JThis document will not have a glossary}%
18059 }%
18060 \let\warn@nomakeglossaries\relax
18061 \let\makeglossaries\@no@makeglossaries
18062 \renewcommand*\@printgloss@setsort{%
18063 \let\@glo@assign@sortkey\@glo@assign@sortkey
18064 \def\@glo@sorttype{\@glo@default@sorttype}%
18065 }%
18066 \renewcommand*\new@glossaryentry[2]{%
18067 \PackageError{glossaries}{Glossary entries must be
18068 defined in the preamble^^Jwhen you use
18069 \string\makenoidxglossaries}%
18070 {Either move your definitions to the preamble or use
18071 \string\makeglossaries}%
18072 }%
18073 \renewcommand*\@glsentrynumberlist}[1]{%
18074 \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18075 \ifdef\@gls@loclist

```

```

18076 {%
18077   \glsnoidxloclist{\@gls@loclist}%
18078 }%
18079 {%
18080   ??\glsdoifexists{##1}%
18081   {%
18082     \GlossariesWarning{Missing location list for ‘##1’. Either
18083       a rerun is required or you haven’t referenced the entry}%
18084   }%
18085 }%
18086 }%
18087 \renewcommand*{\glsdisplaynumberlist}[1]{%
18088   \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18089   \ifdef\@gls@loclist
18090   {%
18091     \def\@gls@noidxloclist@sep{%
18092       \def\@gls@noidxloclist@sep{%
18093         \def\@gls@noidxloclist@sep{%
18094           \glsnumlistsep
18095         }%
18096         \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
18097       }%
18098     }%
18099     \def\@gls@noidxloclist@finalsep{}%
18100     \def\@gls@noidxloclist@prev{}%
18101     \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
18102     \@gls@noidxloclist@finalsep
18103     \@gls@noidxloclist@prev
18104   }%
18105   {%
18106     ??\glsdoifexists{##1}%
18107     {%
18108       \GlossariesWarning{Missing location list for ‘##1’. Either
18109         a rerun is required or you haven’t referenced the entry}%
18110     }%
18111   }%
18112 }%
18113 \renewcommand*{\glsnumberlistloop}[3]{%
18114   \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18115   \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
18116   \let\@gls@org@glsseeformat\glsseeformat
18117   \let\glsnoidxdisplayloc##2\relax
18118   \let\glsseeformat##3\relax
18119   \ifdef\@gls@loclist
18120   {%
18121     \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
18122   }%
18123   {%
18124     ??\glsdoifexists{##1}%

```

```

18125      {%
18126      \GlossariesWarning{Missing location list for ‘##1’. Either
18127      a rerun is required or you haven’t referenced the entry}%
18128      }%
18129      }%
18130      \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
18131      \let\glsseeformat\@gls@org@glsseeformat
18132      }%
18133      \let\@gls@sanitizesort\@gls@noidx@sanitizesort
18134      \let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
18135      \@gls@noidx@setsanitizesort
18136      \@glo@check@sortallowed\makenoidxglossaries
18137      }%
18138      }
18139      \onlypreamble{\makenoidxglossaries}
18140      \newcommand*{\glsnumberlistloop}[2]{%
18141      \PackageError{glossaries}{\string\glsnumberlistloop\space
18142      only works with \string\makenoidxglossaries}{}%
18143      }
18144      \newcommand*{\glsnoidxnumberlistloop}[1]{%
18145      #1%
18146      }
18147      \newcommand*{\@no@makeglossaries}{%
18148      \PackageError{glossaries}{You can’t use both
18149      \string\makeglossaries\space and \string\makenoidxglossaries}%
18150      {Either use one or other (or none) of those commands but not both
18151      together.}%
18152      }
18153      \newcommand{\@gls@noref@warn}[1]{%
18154      \GlossariesWarning{\string\makenoidxglossaries\space
18155      is required to make \string\printnoidxglossary[type={#1}] work}%
18156      }
18157      \newcommand*{\gls@noidxglossary}{%
18158      \protected@write\@auxout{}{%
18159      \string\@gls@reference
18160      {\csname glo@\@gls@label @type\endcsname}%
18161      {\@gls@label}%
18162      {\string\glsnoidxdisplayloc
18163      {\@glo@counterprefix}%
18164      {\@gls@counter}%
18165      {\@glsnumberformat}%
18166      {\@glslocref}%
18167      }%
18168      }%
18169      }
18170      \providecommand\istfile{\glswrite}
18171      \AtEndDocument{%
18172      \glswritefiles
18173      }

```

```

18174 \newcommand*{\@glswritefiles}{%
18175   \forallglossaries{\@glo@type}{%
18176     \ifcsundef{glo@\@glo@type @filetok}%
18177     {%
18178       \def\gls@tmp{}%
18179     }%
18180     {%
18181       \edef\gls@tmp{\expandafter\the
18182         \csname glo@\@glo@type @filetok\endcsname}%
18183     }%
18184     \ifx\gls@tmp\@empty
18185       \ifx\@glo@type\glsdefaulttype
18186         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
18187           entries.^^JRemember to use package option 'nomain' if
18188 you
18189           don't want to^^Juse the main glossary}%
18190       \else
18191         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
18192           entries}%
18193       \fi
18194     \else
18195       \@glsopenfile{\glswrite}{\@glo@type}%
18196       \immediate\write\glswrite{%
18197         \expandafter\the
18198         \csname glo@\@glo@type @filetok\endcsname}%
18199       \immediate\closeout\glswrite
18200     \fi
18201   }%
18202 }
18203 \newcommand*{\gls@glossary}[1]{%
18204   \@gls@glossary{#1}%
18205 }
18206 \newcommand*{\@gls@glossary}[2]{%
18207   \if@gls@debug
18208     \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
18209   \fi
18210 }
18211 \newcommand{\@gls@renewglossary}{%
18212   \gdef\@gls@glossary##1{\@bsphack\beginingroup\gls@wrglossary{##1}}%
18213   \let\@gls@renewglossary\@empty
18214 }
18215 \newcommand*{\gls@wrglossary}[2]{%
18216   \ifglssavewrites
18217     \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
18218     \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
18219     \expandafter{\@gls@tmp^^J}%
18220   \else
18221     \ifcsdef{glo@#1@file}%
18222     {%

```

```

18223 \expandafter\protected@write\csname glo@#1@file\endcsname{%
18224 \gls@disablepagerefexpansion}{#2}%
18225 }%
18226 {%
18227 \ifignoredglossary{#1}{}%
18228 {%
18229 \GlossariesWarning{No file defined for glossary '#1'}%
18230 }%
18231 }%
18232 \fi
18233 \endgroup\@esphack
18234 }
18235 \newcommand*{\@do@wrglossary}[1]{%
18236 \glswriteentry{#1}{\@do@wrglossary{#1}}%
18237 }
18238 \newcommand*{\glswriteentry}[2]{%
18239 \ifglsindexonlyfirst
18240 \ifglsused{#1}{#2}%
18241 \else
18242 #2%
18243 \fi
18244 }
18245 \newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
18246 \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
18247 \newcommand*{\gls@disablepagerefexpansion}{%
18248 \@for\@gls@this:=\gls@protected@pagefmts\do
18249 {%
18250 \expandafter\let\@gls@this\relax
18251 }%
18252 }
18253 \newcommand*{\gls@alphpage}{\@alph\c@page}
18254 \newcommand*{\gls@Alphpage}{\@Alph\c@page}
18255 \newcommand*{\gls@numberpage}{\number\c@page}
18256 \newcommand*{\gls@arabicpage}{\@arabic\c@page}
18257 \newcommand*{\gls@romanpage}{\romannumeral\c@page}
18258 \newcommand*{\gls@Romanpage}{\@Roman\c@page}
18259 \newcommand*{\glsaddprotectedpagefmt}[1]{%
18260 \eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
18261 \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
18262 \eappto\@wrglossarynumberhook{%
18263 \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
18264 \expandonce{\csname#1\endcsname}%
18265 \noexpand\def\expandonce{\csname#1\endcsname}{%
18266 \noexpand\@wrglossary@pageformat
18267 \expandonce{\csname gls#1page\endcsname}%
18268 \expandonce{\csname org@gls#1\endcsname}%
18269 }%
18270 }%
18271 }

```

```

18272 \newcommand*\@wrglossarynumberhook{}
18273 \newcommand{\@wrglossary@pageformat}[3]{%
18274   \ifx#3\c@page #1\else #2#3\fi
18275 }
18276 \newcommand*{\@do@wrglossary}[1]{%
18277   \ifglswrallowlocations
18278     \@do@esc@wrglossary{#1}%
18279   \else
18280     \@do@noesc@wrglossary{#1}%
18281   \fi
18282 }
18283 \newcommand*{\@do@noesc@wrglossary}[1]{%
18284   \expandafter\def\expandafter\@glslocref\expandafter{\theglslsentrycounter}%
18285   \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglslsentrycounter}%
18286   \ifx\@glsHlocref\@glslocref
18287     \def\@glo@counterprefix{}%
18288   \else
18289     \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
18290       {\@glslocref}{\@glsHlocref}}%
18291     }%
18292     \@do@gls@getcounterprefix
18293   \fi
18294   \edef\@gls@label{\glsdetoklabel{#1}}%
18295   \@do@@@wrglossary
18296 }
18297 \newif\ifglswrallowprimitivemods
18298 \glswrallowprimitivemodstrue
18299 \newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
18300   \begingroup
18301     \let\gls@orgthe\the
18302     \let\gls@orgnumber\number
18303     \let\gls@orgarabic\@arabic
18304     \let\gls@orgromannumeral\romannumeral
18305     \let\gls@orgalph\@alph
18306     \let\gls@orgAlph\@Alph
18307     \let\gls@orgRoman\@Roman
18308     \ifglswrallowprimitivemods
18309       \def\gls@the##1{%
18310         \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
18311       \def\the{\expandafter\gls@the}%
18312       \def\gls@number##1{%
18313         \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
18314       \def\number{\expandafter\gls@number}%
18315     \fi
18316     \def\@arabic##1{%
18317       \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
18318     \def\romannumeral##1{%
18319       \ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
18320     \def\@Roman##1{%

```



```

18321 \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
18322 \def\@alph##1{%
18323 \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
18324 \def\@Alph##1{%
18325 \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
18326 \@wrglossarynumberhook
18327 \gls@disablepagerefexpansion
18328 \protected@xdef\@glslocref{\theglsentrycounter}%
18329 \endgroup
18330 \@gls@checkmkidxchars\@glslocref
18331 \expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
18332 \def\@glo@counterprefix{}%
18333 \else
18334 \protected@edef\@glsHlocref{\theglsentrycounter}%
18335 \@gls@checkmkidxchars\@glsHlocref
18336 \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
18337 {\@glslocref}{\@glsHlocref}}%
18338 }%
18339 \@do@gls@getcounterprefix
18340 \fi
18341 \edef\@gls@label{\glsdetoklabel{#1}}%
18342 \@do@wrglossary
18343 }
18344 \newcommand*{\@do@wrglossary}{%
18345 \ifglxindy
18346 \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
18347 \def\@glo@range{}%
18348 \expandafter\if\@glo@prefix\relax
18349 \def\@glo@range{:open-range}%
18350 \else
18351 \expandafter\if\@glo@prefix\relax
18352 \def\@glo@range{:close-range}%
18353 \fi
18354 \fi
18355 \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
18356 (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
18357 :locref \string{\@glo@counterprefix}{\@glslocref}\string" %
18358 :attr \string"\@gls@counter\@glo@suffix\string"
18359 \@glo@range
18360 )
18361 }%
18362 \else
18363 \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
18364 {\@glo@counterprefix}%
18365 \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
18366 \string\glossaryentry{\csname glo@\@gls@label @index\endcsname
18367 \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
18368 \fi
18369 }

```

```

18370 \newcommand*\@gls@getcounterprefix[2]{%
18371   \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
18372   \ifx\@gls@thisloc\@gls@thisHloc
18373     \def\@glo@counterprefix{}%
18374   \else
18375     \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
18376       \def\@glo@tmp{##2}%
18377       \ifx\@glo@tmp\@empty
18378         \def\@glo@counterprefix{}%
18379       \else
18380         \def\@glo@counterprefix{##1}%
18381     \fi
18382   }%
18383   \@gls@get@counterprefix#2.#1\end@getprefix
18384   \ifx\@glo@counterprefix\@empty
18385     \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
18386       prefixing^^Jlocation ‘#1’. You need to modify the
18387       definition of \string\theH\@gls@counter^^Jotherwise you
18388       will get the warning: “‘name{\@gls@counter.#1}’ has been^^J
18389       referenced but does not exist”}%
18390   \fi
18391 \fi
18392 }
18393 \newcommand{\@do@seeglossary}[2]{%
18394 \def\@gls@xref{#2}%
18395 \@onelevel@sanitize\@gls@xref
18396 \@gls@checkmkidxchars\@gls@xref
18397 \ifglxindy
18398   \gls@glossary{\csname glo@#1@type\endcsname}{%
18399     (indexentry
18400       :tkey (\csname glo@#1@index\endcsname)
18401       :xref (\string"\@gls@xref\string")
18402       :attr \string"see\string"
18403     )
18404   }%
18405 \else
18406   \gls@glossary{\csname glo@#1@type\endcsname}{%
18407     \string@glossaryentry{\csname glo@#1@index\endcsname
18408       \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
18409 \fi
18410 }
18411 \def\@gls@fixbraces#1#2#3\@nil{%
18412   \ifx#2[\relax
18413     \@gls@fixbraces#1#2#3\@end@fixbraces
18414   \else
18415     \def#1{{#2#3}}%
18416   \fi
18417 }
18418 \def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%

```

```

18419 \def#1{[#2]{#3}}%
18420 }
18421 \newrobustcmd*{\glssee}[3][\seename]{%
18422   \@do@seeglossary{#2}{[#1]{#3}}}
18423 \newcommand*{\@glssee}[3][\seename]{%
18424   \glssee[#1]{#3}{#2}}
18425 \newrobustcmd*{\glsseeformat}[3][\seename]{%
18426   \emph{#1} \glsseelist{#2}}
18427 \newrobustcmd*{\glsseelist}[1]{%
18428   \let\@gls@dolast\relax
18429   \let\@gls@donext\relax
18430   \@for\@gls@thislabel:=#1\do{%
18431     \ifx\@xfor@nextelement\@nnil
18432       \@gls@dolast
18433     \else
18434       \@gls@donext
18435     \fi
18436     \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
18437     \let\@gls@dolast\glsseelastsep
18438     \let\@gls@donext\glsseesep
18439   }%
18440 }
18441 \newcommand*{\glsseelastsep}{\space\andname\space}
18442 \newcommand*{\glsseesep}{, }
18443 \newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
18444 \newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
18445 \newcommand*{\gls@save@numberlist}[1]{%
18446   \ifglssavenumberlist
18447     \toks@{#1}%
18448     \edef\@do@writeaux@info{%
18449       \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
18450     }%
18451     \@onelevel@sanitize\@do@writeaux@info
18452     \protected@write\@auxout{}\@do@writeaux@info%
18453   \fi
18454 }
18455 \newcommand*{\warn@noprintglossary}{}%
18456 \ifcsundef{printglossary}{}%
18457 {%
18458   \@gls@warnonglossdefined
18459   \undef\printglossary
18460 }
18461 \newcommand*{\@printgloss@checkexists@noignored}[2]{%
18462   \@ifglossaryexists{#1}%
18463   {#2}%
18464   {%
18465     \ifignoredglossary{#1}%
18466     {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
18467     {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%

```

```

18468 }%
18469 }
18470 \newcommand*{\@printgloss@checkexists@allowignored}[2]{%
18471   \s@ifglossaryexists{#1}%
18472   {#2}%
18473   {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
18474 }
18475 \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18476 \newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
18477   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18478   \@printglossary{#1}{\@print@glossary}%
18479 }
18480 \newcommand*{\printglossaries}{%
18481   \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
18482 }
18483 \newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
18484   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18485   \@printglossary{#1}{\@print@noidx@glossary}%
18486 }
18487 \newcommand*{\printnoidxglossaries}{%
18488   \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
18489 }
18490 \newcommand*{\@printgloss@setsort}{}
18491 \newcommand*{\@gls@preglossaryhook}{}
18492 \newcommand{\@printglossary}[2]{%
18493   \def\@glo@type{\glsdefaulttype}%
18494   \def\glossarytitle{\csname @glotype@\@glo@type @title\endcsname}%
18495   \def\glossarytoctitle{\glossarytitle}%
18496   \let\org@glossarytitle\glossarytitle
18497   \def\@glossarystyle{%
18498     \ifx\@glossary@default@style\relax
18499       \GlossariesWarning{No default glossary style provided \MessageBreak
18500         for the glossary ‘\@glo@type’. \MessageBreak
18501         Using deprecated fallback. \MessageBreak
18502         To fix this set the style with \MessageBreak
18503         \string\setglossarystyle\space or use the \MessageBreak
18504         style key=value option}%
18505     \fi
18506   }%
18507   \def\gls@dotocitle{\glssettocitle{\@glo@type}}%
18508   \let\@org@glossaryentrynumbers\glossaryentrynumbers
18509   \bgroup
18510   \@printgloss@setsort
18511   \setkeys{printgloss}{#1}%
18512   \@printgloss@checkexists{\@glo@type}%
18513   {%
18514     \ifx\glossarytitle\org@glossarytitle
18515     \else
18516       \expandafter\let\csname @glotype@\@glo@type @title\endcsname

```

```

18517             \glossarytitle
18518     \fi
18519     \let\currentglossary\@glo@type
18520     \let\org@glossaryentrynumbers\glossaryentrynumbers
18521     \let\glsnonextpages\@glsnonextpages
18522     \let\glsnextpages\@glsnextpages
18523     \let\nopostdesc\@nopostdesc
18524     \gls@dotoc@title
18525     \@glossarystyle
18526     \let\gls@org@glossaryentryfield\glossentry
18527     \let\gls@org@glossarysubentryfield\subglossentry
18528     \renewcommand{\glossentry}[1]{%
18529       \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
18530       \gls@org@glossaryentryfield{##1}%
18531     }%
18532     \renewcommand{\subglossentry}[2]{%
18533       \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
18534       \gls@org@glossarysubentryfield{##1}{##2}%
18535     }%
18536     \@gls@preglossaryhook
18537     #2%
18538   }%
18539 \egroup
18540 \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
18541 \global\let\warn@noprintglossary\relax
18542 }
18543 \newcommand{\@print@glossary}{%
18544   \makeatletter
18545   \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
18546   \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
18547   {}%
18548   {\null}%
18549   \ifglxindy
18550     \ifcsundef{@xdy@\@glo@type @language}%
18551     {%
18552       \edef\@do@auxoutstuff{%
18553         \noexpand\AtEndDocument{%
18554           \noexpand\immediate\noexpand\write\@auxout{%
18555             \string\providecommand\string\@xdylanguage[2]{}}%
18556           \noexpand\immediate\noexpand\write\@auxout{%
18557             \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
18558         }%
18559       }%
18560     }%
18561   {%
18562     \edef\@do@auxoutstuff{%
18563       \noexpand\AtEndDocument{%
18564         \noexpand\immediate\noexpand\write\@auxout{%
18565           \string\providecommand\string\@xdylanguage[2]{}}%

```

```

18566         \noexpand\immediate\noexpand\write\@auxout{%
18567             \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
18568                 @language\endcsname}}}%
18569     }%
18570 }%
18571 }%
18572 \do@auxoutstuff
18573 \edef\do@auxoutstuff{%
18574     \noexpand\AtEndDocument{%
18575         \noexpand\immediate\noexpand\write\@auxout{%
18576             \string\providecommand\string\@gls@codepage[2]{}}%
18577         \noexpand\immediate\noexpand\write\@auxout{%
18578             \string\@gls@codepage{\@glo@type}{\@gls@codepage}}}%
18579     }%
18580 }%
18581 \do@auxoutstuff
18582 \fi
18583 \renewcommand*{\@warn@nomakeglossaries}{%
18584     \GlossariesWarningNoLine{\string\makeglossaries\space
18585         hasn't been used,^^Jthe glossaries will not be updated}%
18586 }%
18587 }
18588 \newcommand*{\@glo@sortentries}[2]{%
18589     \glosortentrieswarning
18590     \def\@glo@sortinglist{}%
18591     \def\@glo@sortinghandler{#1}%
18592     \edef\@glo@type{#2}%
18593     \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
18594     \csdef{\@glsref@#2}{}%
18595     \@for\@this@label:=\@glo@sortinglist\do{%
18596         \xifinlistcs{\@this@label}{\@glsref@#2}%
18597         {}%
18598         {%
18599             \listcsxadd{\@glsref@#2}{\@this@label}%
18600         }%
18601         \ifcsdef{\@glo@sortingchildren@\@this@label}%
18602         {%
18603             \@glo@addchildren{#2}{\@this@label}%
18604         }%
18605         {}%
18606     }%
18607 }
18608 \newcommand*{\@glo@addchildren}[2]{%
18609     \bgroup
18610     \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
18611     \@for\@this@childlabel:=\@glo@childlist\do
18612     {%
18613         \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
18614         {}%

```

```

18615     {%
18616         \listcsxadd{@glsref@#1}{\@this@childlabel}%
18617     }%
18618     \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
18619     {%
18620         \@glo@addchildren{#1}{\@this@childlabel}%
18621     }%
18622     {%
18623     }%
18624 }%
18625 \egroup
18626 }
18627 \newcommand*{\@glo@do@sortentries}[1]{%
18628     \ifglshasparent{#1}%
18629     {%
18630         \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
18631         \ifcsundef{@glo@sortingchildren@\@glo@parent}%
18632         {%
18633             \csdef{@glo@sortingchildren@\@glo@parent}{}%
18634         }%
18635     }%
18636     \expandafter\@glo@sortedinsert
18637         \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
18638     \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
18639     {%
18640     }%
18641     {%
18642         \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
18643     }%
18644 }%
18645 {%
18646     \@glo@sortedinsert{\@glo@sortinglist}{#1}%
18647 }%
18648 }
18649 \newcommand*{\@glo@sortedinsert}[2]{%
18650     \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
18651 }%
18652 \newcommand*{\@glo@sorthandler@word}[2]{%
18653     \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
18654     \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
18655     \edef\@glo@do@compare{%
18656         \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
18657         {\expandonce\@gls@sort@B}%
18658         {\expandonce\@gls@sort@A}%
18659     }%
18660     \@glo@do@compare
18661 }
18662 \newcommand*{\@glo@sorthandler@letter}[2]{%
18663     \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%

```

```

18664 \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18665 \edef\glo@do@compare{%
18666   \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
18667   {\expandonce\@gls@sort@B}%
18668   {\expandonce\@gls@sort@A}%
18669 }%
18670 \glo@do@compare
18671 }
18672 \newcommand*{\@glo@sorthandler@case}[2]{%
18673   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
18674   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18675   \edef\glo@do@compare{%
18676     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
18677     {\expandonce\@gls@sort@B}%
18678     {\expandonce\@gls@sort@A}%
18679   }%
18680   \glo@do@compare
18681 }
18682 \newcommand*{\@glo@sorthandler@nocase}[2]{%
18683   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
18684   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18685   \edef\glo@do@compare{%
18686     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
18687     {\expandonce\@gls@sort@B}%
18688     {\expandonce\@gls@sort@A}%
18689   }%
18690   \glo@do@compare
18691 }
18692 \newcommand*{\@glo@sortmacro@word}[1]{%
18693   \ifdefstring{\@glo@default@sorttype}{standard}%
18694   {%
18695     \@glo@sortentries{\@glo@sorthandler@word}{#1}%
18696   }%
18697   {%
18698     \PackageError{glossaries}{Conflicting sort options:^^J
18699       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18700       \string\printnoidxglossary[sort=word]}{}}%
18701   }%
18702 }
18703 \newcommand*{\@glo@sortmacro@letter}[1]{%
18704   \ifdefstring{\@glo@default@sorttype}{standard}%
18705   {%
18706     \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
18707   }%
18708   {%
18709     \PackageError{glossaries}{Conflicting sort options:^^J
18710       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18711       \string\printnoidxglossary[sort=letter]}{}}%
18712   }%

```



```

18713 }
18714 \newcommand*{\@glo@sortmacro@standard}[1]{%
18715   \ifdefstring{\@glo@default@sorttype}{standard}%
18716   {%
18717     \ifcsdef{\@glo@sorthandler@\glsorder}%
18718     {%
18719       \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
18720     }%
18721     {%
18722       \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}%
18723     }%
18724   }%
18725   {%
18726     \PackageError{glossaries}{Conflicting sort options:^^J
18727       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18728       \string\printnoidxglossary[sort=standard]}{}%
18729   }%
18730 }
18731 \newcommand*{\@glo@sortmacro@case}[1]{%
18732   \ifdefstring{\@glo@default@sorttype}{standard}%
18733   {%
18734     \@glo@sortentries{\@glo@sorthandler@case}{#1}%
18735   }%
18736   {%
18737     \PackageError{glossaries}{Conflicting sort options:^^J
18738       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18739       \string\printnoidxglossary[sort=case]}{}%
18740   }%
18741 }
18742 \newcommand*{\@glo@sortmacro@nocase}[1]{%
18743   \ifdefstring{\@glo@default@sorttype}{standard}%
18744   {%
18745     \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
18746   }%
18747   {%
18748     \PackageError{glossaries}{Conflicting sort options:^^J
18749       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18750       \string\printnoidxglossary[sort=nocase]}{}%
18751   }%
18752 }
18753 \newcommand*{\@glo@sortmacro@def}[1]{%
18754   \def\@glo@sortinglist{}%
18755   \for\glsentries[#1]{\@gls@thislabel}%
18756   {%
18757     \xifinlistcs{\@gls@thislabel}{\@gls@ref@#1}%
18758     {%
18759       \listead{\@glo@sortinglist}{\@gls@thislabel}%
18760     }%
18761     {%

```

```

18762 }%
18763 }%
18764 \cslet{@glsref@#1}{\@glo@sortinglist}%
18765 }
18766 \newcommand*{\@glo@sortmacro@def@do}[1]{%
18767 \ifinlistcs{#1}{@glsref@\@glo@type}%
18768 {}%
18769 {%
18770 \listcsadd{@glsref@\@glo@type}{#1}%
18771 }%
18772 \ifcsdef{@glo@sortingchildren@#1}%
18773 {%
18774 \@glo@addchildren{\@glo@type}{#1}%
18775 }%
18776 {}%
18777 }
18778 \newcommand*{\@glo@sortmacro@use}[1]{%
18779 \newcommand*{\@print@noidx@glossary}{%
18780 \ifcsdef{@glsref@\@glo@type}%
18781 {%
18782 \ifcsdef{@glo@sortmacro@\@glo@sorttype}%
18783 {%
18784 \csuse{@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
18785 }%
18786 {%
18787 \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
18788 }%
18789 \glossarysection[\glossarytoctitle]{\glossarytitle}%
18790 \glossarypreamble
18791 \def\@gls@currentlettergroup{}%
18792 \begin{theglossary}%
18793 \glossaryheader
18794 \glsresetentrylist
18795 \forlistcsloop{\@gls@noidx@do}{@glsref@\@glo@type}%
18796 \end{theglossary}%
18797 \glossarypostamble
18798 }%
18799 {%
18800 \@gls@noref@warn{\@glo@type}%
18801 }%
18802 }
18803 \def\glo@grabfirst#1#2\@nil{%
18804 \def\@gls@firsttok{#1}%
18805 \ifdefempty\@gls@firsttok
18806 {%
18807 \def\@glo@thislettergrp{0}%
18808 }%
18809 {%
18810 \@onelevel@sanitize\@gls@firsttok

```

```

18811 \expandafter\@glo@grabfirst\@gls@firsttok{}\}\@nil
18812 }%
18813 }
18814 \def\@glo@grabfirst#1#2\@nil{%
18815 \ifdefempty\@glo@thislettergrp
18816 {%
18817 \def\@glo@thislettergrp{glssymbols}%
18818 }%
18819 {%
18820 \count@=\uccode'#1\relax
18821 \ifnum\count@=0\relax
18822 \def\@glo@thislettergrp{glssymbols}%
18823 \else
18824 \ifdefstring\@glo@sorttype{case}%
18825 {%
18826 \count@='#1\relax
18827 }%
18828 {%
18829 }%
18830 \edef\@glo@thislettergrp{\the\count@}%
18831 \fi
18832 }%
18833 }
18834 \newcommand{\@gls@noidx@do}[1]{%
18835 \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
18836 \ifglshasparent{#1}%
18837 {%
18838 \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
18839 \ifdefvoid{\@gls@loclist}
18840 {%
18841 \subglossentry{\gls@level}{#1}{}%
18842 }%
18843 {%
18844 \subglossentry{\gls@level}{#1}%
18845 }%
18846 \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
18847 }%
18848 }%
18849 }%
18850 {%
18851 \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
18852 \expandafter\glo@grabfirst\@gls@sort{}\}\@nil
18853 \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
18854 {}%
18855 {%
18856 \ifdefempty{\@gls@currentlettergroup}{}%
18857 {%
18858 \global\let\@glo@thislettergrp\@glo@thislettergrp
18859 \glsgroupskip

```

```

18860     }%
18861     \glsgroupheading{\@glo@thislettergrp}%
18862     }%
18863     \global\let\@gls@currentlettergroup\@glo@thislettergrp
18864     \ifdefvoid{\@gls@loclist}
18865     {%
18866         \glossentry{#1}{}%
18867     }%
18868     {%
18869         \glossentry{#1}%
18870     }%
18871     \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
18872     }%
18873     }%
18874 }%
18875 }
18876 \newcommand*\@glsnoidxloclist[1]{%
18877     \def\@gls@noidxloclist@sep{}%
18878     \def\@gls@noidxloclist@prev{}%
18879     \forlistloop{\glsnoidxloclisthandler}{#1}%
18880 }
18881 \newcommand*\@glsnoidxloclisthandler[1]{%
18882     \ifdefstring{\@gls@noidxloclist@prev}{#1}%
18883     {%
18884     }%
18885     {%
18886         \@gls@noidxloclist@sep
18887         #1%
18888         \def\@gls@noidxloclist@sep{\delimN}%
18889         \def\@gls@noidxloclist@prev{#1}%
18890     }%
18891 }
18892 \newcommand*\@glsnoidxdisplayloclisthandler[1]{%
18893     \ifdefstring{\@gls@noidxloclist@prev}{#1}%
18894     {%
18895     }%
18896     {%
18897         \@gls@noidxloclist@sep
18898         \@gls@noidxloclist@prev
18899         \def\@gls@noidxloclist@prev{#1}%
18900     }%
18901 }
18902 \newcommand*\@glsnoidxdisplayloc[4]{%
18903     \setentrycounter{#1}{#2}%
18904     \csuse{#3}{#4}%
18905 }
18906 \newcommand*\@gls@reference}[3]{%
18907     \glsdoifexistsorwarn{#2}%
18908     {%

```

```

18909 \ifcsundef{@glsref@#1}{\csgdef{@glsref@#1}{}}{}%
18910 \ifinlistcs{#2}{@glsref@#1}%
18911 {}%
18912 {\listcsgadd{@glsref@#1}{#2}}%
18913 \ifcsundef{glo@glstdetoklabel{#2}@loclist}%
18914 {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%
18915 {}%
18916 \listcsgadd{glo@glstdetoklabel{#2}@loclist}{#3}%
18917 }%
18918 }
18919 \define@key{printgloss}{type}{\def@glo@type{#1}}
18920 \define@key{printgloss}{title}{%
18921 \def@glossarytitle{#1}%
18922 \let@gls@dotoc@title\relax
18923 }
18924 \define@key{printgloss}{toctitle}{%
18925 \def@glossarytoctitle{#1}%
18926 \let@gls@dotoc@title\relax
18927 }
18928 \define@key{printgloss}{style}{%
18929 \ifcsundef{@glsstyle@#1}%
18930 {%
18931 \PackageError{glossaries}%
18932 {Glossary style ‘#1’ undefined}{}%
18933 }%
18934 {%
18935 \def@glossarystyle{\setglossentrycompatibility
18936 \csname @glsstyle@#1\endcsname}%
18937 }%
18938 }
18939 \define@choicekey{printgloss}{numberedsection}%
18940 [\gls@numberedsection@val\gls@numberedsection@nr]%
18941 {false,nolabel,autolabel,nameref}[nolabel]%
18942 {%
18943 \ifcase\gls@numberedsection@nr\relax
18944 \renewcommand*{\@@glossarysecstar}{*}%
18945 \renewcommand*{\@@glossaryseclabel}{}%
18946 \or
18947 \renewcommand*{\@@glossarysecstar}{}%
18948 \renewcommand*{\@@glossaryseclabel}{}%
18949 \or
18950 \renewcommand*{\@@glossarysecstar}{*}%
18951 \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix@glo@type}}%
18952 \or
18953 \renewcommand*{\@@glossarysecstar}{*}%
18954 \renewcommand*{\@@glossaryseclabel}{%
18955 \protected@edef\@currentlabelname{\glossarytoctitle}%
18956 \label{\glsautoprefix@glo@type}}%
18957 \fi

```

```

18958 }
18959 \define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
18960   \csuse{glsnogroupskip#1}%
18961 }
18962 \define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
18963   \csuse{glsnopostdot#1}%
18964 }
18965 \newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
18966 \define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
18967   \csuse{glsentrycounter#1}%
18968   \@gls@define@glossaryentrycounter
18969 }
18970 \define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
18971   \csuse{glssubentrycounter#1}%
18972   \@gls@define@glossarysubentrycounter
18973 }
18974 \define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
18975   \ifglsnonumberlist
18976     \def\glossaryentrynumbers##1{%
18977 \else
18978     \def\glossaryentrynumbers##1{##1}%
18979 \fi}
18980 \define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
18981 \newcommand*{\@glo@no@assign@sortkey}[1]{%
18982   \PackageError{glossaries}{‘sort’ key not permitted with
18983     \string\printglossary}%
18984   {The ‘sort’ key may only be used with \string\printnoidxglossary}%
18985 }
18986 \newcommand*{\@glo@assign@sortkey}[1]{%
18987   \def\@glo@sorttype{#1}%
18988 }
18989 \newcommand*{\@glsnextpages}{%
18990   \gdef\glossaryentrynumbers##1{%
18991     \glsresetentrylist
18992   }%
18993 }
18994 \newcommand*{\@glsnextpages}{%
18995   \gdef\glossaryentrynumbers##1{%
18996     ##1\glsresetentrylist}}
18997 \newcommand*{\glsresetentrylist}{%
18998   \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
18999 \newcommand*{\glsnonextpages}{}
19000 \newcommand*{\glsnextpages}{}
19001 \@gls@define@glossaryentrycounter
19002 \@gls@define@glossarysubentrycounter
19003 \newcommand*{\glsresetsubentrycounter}{%
19004   \ifglssubentrycounter
19005     \setcounter{glossarysubentry}{0}%
19006   \fi

```

```

19007 }
19008 \newcommand*{\glsresetentrycounter}{%
19009   \ifglssentrycounter
19010     \setcounter{glossaryentry}{0}%
19011   \fi
19012 }
19013 \newcommand*{\glsstepentry}[1]{%
19014   \ifglssentrycounter
19015     \refstepcounter{glossaryentry}%
19016     \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19017   \fi
19018 }
19019 \newcommand*{\glsstepsubentry}[1]{%
19020   \ifglsssubentrycounter
19021     \edef\currentglsssubentry{\glsdetoklabel{#1}}%
19022     \refstepcounter{glossarysubentry}%
19023     \label{\GlsEntryCounterLabelPrefix\currentglsssubentry}%
19024   \fi
19025 }
19026 \newcommand*{\glsrefentry}[1]{%
19027   \ifglssentrycounter
19028     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19029   \else
19030     \ifglsssubentrycounter
19031       \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19032     \else
19033       \gls{#1}%
19034     \fi
19035   \fi
19036 }
19037 \newcommand*{\glsentrycounterlabel}{%
19038   \ifglssentrycounter
19039     \theglossaryentry.\space
19040   \fi
19041 }
19042 \newcommand*{\glssubentrycounterlabel}{%
19043   \ifglsssubentrycounter
19044     \theglossarysubentry)\space
19045   \fi
19046 }
19047 \newcommand*{\glsentryitem}[1]{%
19048   \ifglssentrycounter
19049     \glsstepentry{#1}\glsentrycounterlabel
19050   \else
19051     \glsresetsubentrycounter
19052   \fi
19053 }
19054 \newcommand*{\glssubentryitem}[1]{%
19055   \ifglsssubentrycounter

```

```

19056 \glsstepsubentry{#1}\glssubentrycounterlabel
19057 \fi
19058 }
19059 \ifcsundef{theglossary}%
19060 {%
19061 \newenvironment{theglossary}{}{}%
19062 }%
19063 {%
19064 \@gls@warnontheglossdefined
19065 \renewenvironment{theglossary}{}{}%
19066 }
19067 \newcommand*{\glossaryheader}{}
19068 \newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
19069 \providecommand*{\compatibleglossentry}[2]{%
19070 \toks@{#2}%
19071 \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
19072 \noexpand\glsnamefont
19073 {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
19074 {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
19075 {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
19076 {\the\toks@}%
19077 }%
19078 \@do@glossentry
19079 }
19080 \newcommand*{\glossentryname}[1]{%
19081 \glsdoifexistsorwarn{#1}%
19082 {%
19083 \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
19084 \expandafter\glsnamefont\expandafter{\glo@name}%
19085 }%
19086 }
19087 \newcommand*{\Glossentryname}[1]{%
19088 \glsdoifexistsorwarn{#1}%
19089 {%
19090 \glsnamefont{\Glsentryname{#1}}%
19091 }%
19092 }
19093 \newcommand*{\glossentrydesc}[1]{%
19094 \glsdoifexistsorwarn{#1}%
19095 {%
19096 \glsentrydesc{#1}%
19097 }%
19098 }
19099 \newcommand*{\Glossentrydesc}[1]{%
19100 \glsdoifexistsorwarn{#1}%
19101 {%
19102 \Glsentrydesc{#1}%
19103 }%
19104 }

```



```

19105 \newcommand*{\glossentrysymbol}[1]{%
19106   \glsdoifexistsorwarn{#1}%
19107   {%
19108     \glsentrysymbol{#1}%
19109   }%
19110 }
19111 \newcommand*{\Glossentrysymbol}[1]{%
19112   \glsdoifexistsorwarn{#1}%
19113   {%
19114     \Glsentrysymbol{#1}%
19115   }%
19116 }
19117 \providecommand*{\compatiblesubglossentry}[3]{%
19118   \toks@{#3}%
19119   \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
19120     {#2}%
19121     {\noexpand\glsnamefont
19122       {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
19123     {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
19124     {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
19125     {\the\toks@}%
19126   }%
19127   \@do@subglossentry
19128 }
19129 \newcommand*{\setglossentrycompatibility}{%
19130   \let\glossentry\compatibleglossentry
19131   \let\subglossentry\compatiblesubglossentry
19132 }
19133 \setglossentrycompatibility
19134 \newcommand{\glossaryentryfield}[5]{%
19135   \GlossariesWarning
19136   {Deprecated use of \string\glossaryentryfield.^^J
19137     I recommend you change to \string\glossentry.^^J
19138     If you've just upgraded, try removing your gls auxiliary
19139     files^^J and recompile}%
19140   \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
19141 \newcommand*{\glossarysubentryfield}[6]{%
19142   \GlossariesWarning
19143   {Deprecated use of \string\glossarysubentryfield.^^J
19144     I recommend you change to \string\subglossentry.^^J
19145     If you've just upgraded, try removing your gls auxiliary
19146     files^^J and recompile}%
19147   \glstarget{#2}{\strut}#4. #6\par}
19148 \newcommand*{\glsgroupskip}{}
19149 \newcommand*{\glsgroupheading}[1]{}
19150 \newcommand*{\glsgetgrouptitle}[1]{%
19151   \@gls@getgrouptitle{#1}{\@gls@grptitle}%
19152   \@gls@grptitle
19153 }

```

```

19154 \newcommand*{\@gls@getgrouptitle}[2]{%
19155   \dtl@ifsingle{#1}%
19156   {%
19157     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19158   }%
19159   {%
19160     \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
19161                 or test{\ifstrequal{#1}{glsnumbers}}}%
19162     {%
19163       \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19164     }%
19165     {%
19166       \def#2{#1}%
19167     }%
19168   }%
19169 }
19170 \newcommand*{\@gls@noidx@getgrouptitle}[2]{%
19171   \DTLifint{#1}%
19172   {\edef#2{\char#1\relax}}%
19173   {%
19174     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19175   }%
19176 }
19177 \newcommand*{\glsgetgrouplabel}[1]{%
19178   \ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
19179   \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
19180 \newcommand*{\setentrycounter}[2][ ]{%
19181   \def\@glo@counterprefix{#1}%
19182   \ifx\@glo@counterprefix\@empty
19183     \def\@glo@counterprefix{.}%
19184   \else
19185     \def\@glo@counterprefix{.#1.}%
19186   \fi
19187   \def\glsentrycounter{#2}%
19188 }
19189 \newcommand*{\setglossarystyle}[1]{%
19190   \ifcsundef{@glsstyle@#1}%
19191   {%
19192     \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
19193   }%
19194   {%
19195     \csname @glsstyle@#1\endcsname
19196   }%
19197   \ifx\@glossary@default@style\relax
19198     \protected@edef\@glossary@default@style{#1}%
19199   \fi
19200 }
19201 \newcommand*{\glossarystyle}[1]{%
19202   \ifcsundef{@glsstyle@#1}%

```

```

19203 {%
19204   \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
19205 }%
19206 {%
19207   \GlossariesWarning
19208   {Deprecated command \string\glossarystyle.^~J
19209    I recommend you switch to \string\setglossarystyle\space unless
19210    you want to maintain backward compatibility}%
19211   \setglossentrycompatibility
19212   \csname @glsstyle@#1\endcsname
19213   \ifcsdef{@glscompstyle@#1}%
19214     {\setglossentrycompatibility\csuse{@glscompstyle@#1}}%
19215   }%
19216 }%
19217 \ifx\@glossary@default@style\relax
19218   \protected@edef\@glossary@default@style{#1}%
19219 \fi
19220 }
19221 \newcommand{\newglossarystyle}[2]{%
19222   \ifcsundef{@glsstyle@#1}%
19223   {%
19224     \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
19225   }%
19226   {%
19227     \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
19228   }%
19229 }
19230 \newcommand{\renewglossarystyle}[2]{%
19231   \ifcsundef{@glsstyle@#1}%
19232   {%
19233     \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
19234   }%
19235   {%
19236     \csdef{@glsstyle@#1}{#2}%
19237   }%
19238 }
19239 \newcommand*{\glsnamefont}[1]{#1}
19240 \ifcsundef{hyperlink}%
19241 {%
19242   \def\glshypernumber#1{#1}%
19243 }%
19244 {%
19245   \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
19246 }
19247 \def\@glshypernumber#1\nohyperpage#2#3\@nil{%
19248   \ifx\#1\%
19249     \else
19250       \@delimR#1\delimR\delimR\%
19251   \fi

```

```

19252 \ifx\#2\%
19253 \else
19254 #2%
19255 \fi
19256 \ifx\#3\%
19257 \else
19258 \@glshypernumber#3\@nil
19259 \fi
19260 }
19261 \def\@delimR#1\delimR #2\delimR #3\%
19262 \ifx\#2\%
19263 \@delimN{#1}%
19264 \else
19265 \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
19266 \fi}
19267 \def\@delimN#1{\@delimN#1\delimN \delimN\}
19268 \def\@delimN#1\delimN #2\delimN#3\%
19269 \ifx\#3\%
19270 \@gls@numberlink{#1}%
19271 \else
19272 \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
19273 \fi
19274 }
19275 \def\@gls@numberlink#1{%
19276 \begingroup
19277 \toks@={}%
19278 \@gls@removespaces#1 \@nil
19279 \endgroup}
19280 \def\@gls@removespaces#1 #2\@nil{%
19281 \toks@=\expandafter{\the\toks@#1}%
19282 \ifx\#2\%
19283 \edef\x{\the\toks@}%
19284 \ifx\x\empty
19285 \else
19286 \hyperlink{\glstrycounter\@glo@counterprefix\the\toks@}%
19287 {\the\toks@}%
19288 \fi
19289 \else
19290 \@gls@ReturnAfterFi{%
19291 \@gls@removespaces#2\@nil
19292 }%
19293 \fi
19294 }
19295 \long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
19296 \newcommand*{\hyperrm}[1]{\textrm{\glshypernumber{#1}}}
19297 \newcommand*{\hypersf}[1]{\textsf{\glshypernumber{#1}}}
19298 \newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}
19299 \newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
19300 \newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}

```

```

19301 \newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}
19302 \newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
19303 \newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
19304 \newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
19305 \newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}
19306 \newcommand{\oldacronym}[4][\gls@label]{%
19307   \def\gls@label{#2}%
19308   \newacronym[#4]{#1}{#2}{#3}%
19309   \ifcsundef{xspace}%
19310   {%
19311     \expandafter\edef\csname#1\endcsname{%
19312       \noexpand@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
19313   }%
19314 }%
19315 {%
19316   \expandafter\edef\csname#1\endcsname{%
19317     \noexpand@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
19318       \noexpand\gls{#1}\noexpand\xspace}%
19319   }%
19320 }%
19321 }
19322 \newcommand{\newacronym}[4][{}]{
19323 \newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
19324 \newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
19325 \newcommand*{\glsshortkey}{short}
19326 \newcommand*{\glsshortpluralkey}{shortplural}
19327 \newcommand*{\glslongkey}{long}
19328 \newcommand*{\glslongpluralkey}{longplural}
19329 \newrobustcmd*{\acrfull}{\@gls@hyp@opt\@ns@acrfull}
19330 \newcommand*\ns@acrfull[2][{}]{%
19331   \new@ifnextchar[{\@acrfull{#1}{#2}}{%
19332     {\@acrfull{#1}{#2}}[{}]}%
19333 }
19334 \def\@acrfull#1#2[#3]{%
19335   \acrfullfmt{#1}{#2}{#3}%
19336 }
19337 \newcommand*{\acrfullfmt}[3]{%
19338   \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
19339 }
19340 \newcommand{\acrlinkfullformat}[5]{%
19341   \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[{}]}%
19342 }
19343 \newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
19344 \newrobustcmd{\glsspace}{\space}
19345 \newrobustcmd*{\Acrfull}{\@gls@hyp@opt\@ns@Acrfull}
19346 \newcommand*\ns@Acrfull[2][{}]{%
19347   \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
19348     {\@Acrfull{#1}{#2}}[{}]}%
19349 }

```

```

19350 \def\@Acrfull#1#2[#3]{%
19351   \Acrfullfmt{#1}{#2}{#3}%
19352 }
19353 \newcommand*\@Acrfullfmt}[3]{%
19354   \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
19355 }
19356 \newrobustcmd*\@Acrfull{\@gls@hyp@opt\ns@Acrfull}
19357 \newcommand*\ns@Acrfull[2][{}]{%
19358   \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
19359     {\@Acrfull{#1}{#2}[]}%
19360 }
19361 \def\@ACRfull#1#2[#3]{%
19362   \ACRfullfmt{#1}{#2}{#3}%
19363 }
19364 \newcommand*\@ACRfullfmt}[3]{%
19365   \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
19366 }
19367 \newrobustcmd*\@acrfullpl{\@gls@hyp@opt\ns@acrfullpl}
19368 \newcommand*\ns@acrfullpl[2][{}]{%
19369   \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
19370     {\@acrfullpl{#1}{#2}[]}%
19371 }
19372 \def\@acrfullpl#1#2[#3]{%
19373   \acrfullplfmt{#1}{#2}{#3}%
19374 }
19375 \newcommand*\@acrfullplfmt}[3]{%
19376   \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
19377 }
19378 \newrobustcmd*\@Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
19379 \newcommand*\ns@Acrfullpl[2][{}]{%
19380   \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
19381     {\@Acrfullpl{#1}{#2}[]}%
19382 }
19383 \def\@Acrfullpl#1#2[#3]{%
19384   \Acrfullplfmt{#1}{#2}{#3}%
19385 }
19386 \newcommand*\@Acrfullplfmt}[3]{%
19387   \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
19388 }
19389 \newrobustcmd*\@ACRfullpl{\@gls@hyp@opt\ns@ACRfullpl}
19390 \newcommand*\ns@ACRfullpl[2][{}]{%
19391   \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
19392     {\@ACRfullpl{#1}{#2}[]}%
19393 }
19394 \def\@ACRfullpl#1#2[#3]{%
19395   \ACRfullplfmt{#1}{#2}{#3}%
19396 }
19397 \newcommand*\@ACRfullplfmt}[3]{%
19398   \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%

```

```

19399 }
19400 \newcommand{\acronymfont}[1]{#1}
19401 \newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
19402 \newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
19403 \newtoks\glskeylisttok
19404 \newtoks\glslabeltok
19405 \newtoks\glsshorttok
19406 \newtoks\glslongtok
19407 \newcommand*{\newacronymhook}{}
19408 \newcommand*{\SetGenericNewAcronym}{%
19409   \let\@Gls@entryname\@Gls@acrenryname
19410   \renewcommand{\newacronym}[4][]{%
19411     \ifdefempty{\@Gls@acronymlists}%
19412       {%
19413         \def\@glo@type{\acronymtype}%
19414         \setkeys{glossentry}{##1}%
19415         \DeclareAcronymList{\@glo@type}%
19416       }%
19417     }%
19418     \glskeylisttok{##1}%
19419     \glslabeltok{##2}%
19420     \glsshorttok{##3}%
19421     \glslongtok{##4}%
19422     \newacronymhook
19423     \protected@edef\@do@newglossaryentry{%
19424       \noexpand\newglossaryentry{\the\glslabeltok}%
19425       {%
19426         type=\acronymtype,%
19427         name={\expandonce{\acronymentry{##2}}},%
19428         sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
19429         text={\the\glsshorttok},%
19430         short={\the\glsshorttok},%
19431         shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
19432         long={\the\glslongtok},%
19433         longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
19434         \GenericAcronymFields,%
19435         \the\glskeylisttok
19436       }%
19437     }%
19438     \@do@newglossaryentry
19439   }%
19440   \renewcommand*{\acrfullfmt}[3]{%
19441     \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
19442   \renewcommand*{\Acrfullfmt}[3]{%
19443     \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
19444   \renewcommand*{\ACRfullfmt}[3]{%
19445     \glslink[##1]{##2}{%
19446       \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
19447   \renewcommand*{\acrfullplfmt}[3]{%

```

```

19448 \glslink{##1}{##2}{\genplacrfullformat{##2}{##3}}}%
19449 \renewcommand*{\Acrfullplfmt}[3]{%
19450 \glslink{##1}{##2}{\Genplacrfullformat{##2}{##3}}}%
19451 \renewcommand*{\ACRfullplfmt}[3]{%
19452 \glslink{##1}{##2}{%
19453 \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}}}%
19454 \renewcommand*{\glstryfull}[1]{\genacrfullformat{##1}{}}}%
19455 \renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
19456 \renewcommand*{\glstryfullpl}[1]{\genplacrfullformat{##1}{}}}%
19457 \renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
19458 }
19459 \newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
19460 \newcommand*{\acronymentry}[1]{\acronymfont{\glstryshort{#1}}}
19461 \newcommand*{\acronymsort}[2]{#1}
19462 \newcommand*{\setacronymstyle}[1]{%
19463 \ifcsundef{@glsacr@dispstyle@#1}
19464 {%
19465 \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
19466 }%
19467 {%
19468 \ifdefempty{\@glsacronymlists}%
19469 {%
19470 \DeclareAcronymList{\acronymtype}%
19471 }%
19472 }%
19473 \SetGenericNewAcronym
19474 \GlsUseAcrStyleDefs{#1}%
19475 \@for\@gls@type:=\@glsacronymlists\do{%
19476 \defglstryfmt[\@gls@type]{\GlsUseAcrEntryDisplayStyle{#1}}}%
19477 }%
19478 }%
19479 }
19480 \newcommand*{\newacronymstyle}[3]{%
19481 \ifcsdef{@glsacr@dispstyle@#1}%
19482 {%
19483 \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
19484 }%
19485 {%
19486 \csdef{@glsacr@dispstyle@#1}{#2}%
19487 \csdef{@glsacr@styledefs@#1}{#3}%
19488 }%
19489 }
19490 \newcommand*{\renewacronymstyle}[3]{%
19491 \ifcsdef{@glsacr@dispstyle@#1}%
19492 {%
19493 \csdef{@glsacr@dispstyle@#1}{#2}%
19494 \csdef{@glsacr@styledefs@#1}{#3}%
19495 }%
19496 {%

```



```

19497 \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
19498 }%
19499 }
19500 \newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
19501 \newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
19502 \newacronymstyle{long-short}%
19503 {%
19504 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19505 }%
19506 {%
19507 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19508 \renewcommand*{\genacrfullformat}[2]{%
19509 \glsentrylong{##1}##2\space
19510 (\protect\firstacronymfont{\glsentryshort{##1}})%
19511 }%
19512 \renewcommand*{\Genacrfullformat}[2]{%
19513 \Glsentrylong{##1}##2\space
19514 (\protect\firstacronymfont{\glsentryshort{##1}})%
19515 }%
19516 \renewcommand*{\genplacrfullformat}[2]{%
19517 \glsentrylongpl{##1}##2\space
19518 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19519 }%
19520 \renewcommand*{\Genplacrfullformat}[2]{%
19521 \Glsentrylongpl{##1}##2\space
19522 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19523 }%
19524 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19525 \renewcommand*{\acronymsort}[2]{##1}%
19526 \renewcommand*{\acronymfont}[1]{##1}%
19527 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19528 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19529 }
19530 \newacronymstyle{long-sp-short}%
19531 {%
19532 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19533 }%
19534 {%
19535 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19536 \renewcommand*{\genacrfullformat}[2]{%
19537 \glsentrylong{##1}##2\glsacspace{##1}%
19538 (\protect\firstacronymfont{\glsentryshort{##1}})%
19539 }%
19540 \renewcommand*{\Genacrfullformat}[2]{%
19541 \Glsentrylong{##1}##2\glsacspace{##1}%
19542 (\protect\firstacronymfont{\glsentryshort{##1}})%
19543 }%
19544 \renewcommand*{\genplacrfullformat}[2]{%
19545 \glsentrylongpl{##1}##2\glsacspace{##1}%

```

```

19546 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19547 }%
19548 \renewcommand*{\Genplacrfullformat}[2]{%
19549 \Glsentrylongpl{##1}##2\glsacspace{##1}%
19550 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19551 }%
19552 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19553 \renewcommand*{\acronymsort}[2]{##1}%
19554 \renewcommand*{\acronymfont}[1]{##1}%
19555 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19556 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19557 }
19558 \newcommand*{\glsacspace}[1]{%
19559 \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{#1}})}%
19560 \ifdim\dimen@<3em~\else\spacefi
19561 }
19562 \newacronymstyle{short-long}%
19563 {%
19564 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19565 }%
19566 {%
19567 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19568 \renewcommand*{\genacrfullformat}[2]{%
19569 \protect\firstacronymfont{\glsentryshort{##1}}##2\space
19570 (\glsentrylong{##1})%
19571 }%
19572 \renewcommand*{\Genacrfullformat}[2]{%
19573 \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
19574 (\glsentrylong{##1})%
19575 }%
19576 \renewcommand*{\genplacrfullformat}[2]{%
19577 \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
19578 (\glsentrylongpl{##1})%
19579 }%
19580 \renewcommand*{\Genplacrfullformat}[2]{%
19581 \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
19582 (\glsentrylongpl{##1})%
19583 }%
19584 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19585 \renewcommand*{\acronymsort}[2]{##1}%
19586 \renewcommand*{\acronymfont}[1]{##1}%
19587 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19588 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19589 }
19590 \newacronymstyle{long-sc-short}%
19591 {%
19592 \GlsUseAcrEntryDispStyle{long-short}%
19593 }%
19594 {%

```

```

19595 \GlsUseAcrStyleDefs{long-short}%
19596 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19597 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19598 }
19599 \newacronymstyle{long-sm-short}%
19600 {%
19601 \GlsUseAcrEntryDispStyle{long-short}%
19602 }%
19603 {%
19604 \GlsUseAcrStyleDefs{long-short}%
19605 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19606 \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
19607 }
19608 \newacronymstyle{sc-short-long}%
19609 {%
19610 \GlsUseAcrEntryDispStyle{short-long}%
19611 }%
19612 {%
19613 \GlsUseAcrStyleDefs{short-long}%
19614 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19615 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19616 }
19617 \newacronymstyle{sm-short-long}%
19618 {%
19619 \GlsUseAcrEntryDispStyle{short-long}%
19620 }%
19621 {%
19622 \GlsUseAcrStyleDefs{short-long}%
19623 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19624 \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
19625 }
19626 \newacronymstyle{long-short-desc}%
19627 {%
19628 \GlsUseAcrEntryDispStyle{long-short}%
19629 }%
19630 {%
19631 \GlsUseAcrStyleDefs{long-short}%
19632 \renewcommand*{\GenericAcronymFields}{}%
19633 \renewcommand*{\acronymsort}[2]{##2}%
19634 \renewcommand*{\acronymentry}[1]{%
19635 \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19636 }
19637 \newacronymstyle{long-sp-short-desc}%
19638 {%
19639 \GlsUseAcrEntryDispStyle{long-sp-short}%
19640 }%
19641 {%
19642 \GlsUseAcrStyleDefs{long-sp-short}%
19643 \renewcommand*{\GenericAcronymFields}{}%

```

```

19644 \renewcommand*{\acronymsort}[2]{##2}%
19645 \renewcommand*{\acronymentry}[1]{%
19646     \glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
19647 }
19648 \newacronymstyle{long-sc-short-desc}%
19649 {%
19650     \GlsUseAcrEntryDisplayStyle{long-sc-short}%
19651 }%
19652 {%
19653     \GlsUseAcrStyleDefs{long-sc-short}%
19654     \renewcommand*{\GenericAcronymFields}{}%
19655     \renewcommand*{\acronymsort}[2]{##2}%
19656     \renewcommand*{\acronymentry}[1]{%
19657         \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
19658 }
19659 \newacronymstyle{long-sm-short-desc}%
19660 {%
19661     \GlsUseAcrEntryDisplayStyle{long-sm-short}%
19662 }%
19663 {%
19664     \GlsUseAcrStyleDefs{long-sm-short}%
19665     \renewcommand*{\GenericAcronymFields}{}%
19666     \renewcommand*{\acronymsort}[2]{##2}%
19667     \renewcommand*{\acronymentry}[1]{%
19668         \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
19669 }
19670 \newacronymstyle{short-long-desc}%
19671 {%
19672     \GlsUseAcrEntryDisplayStyle{short-long}%
19673 }%
19674 {%
19675     \GlsUseAcrStyleDefs{short-long}%
19676     \renewcommand*{\GenericAcronymFields}{}%
19677     \renewcommand*{\acronymsort}[2]{##2}%
19678     \renewcommand*{\acronymentry}[1]{%
19679         \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
19680 }
19681 \newacronymstyle{sc-short-long-desc}%
19682 {%
19683     \GlsUseAcrEntryDisplayStyle{sc-short-long}%
19684 }%
19685 {%
19686     \GlsUseAcrStyleDefs{sc-short-long}%
19687     \renewcommand*{\GenericAcronymFields}{}%
19688     \renewcommand*{\acronymsort}[2]{##2}%
19689     \renewcommand*{\acronymentry}[1]{%
19690         \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
19691 }
19692 \newacronymstyle{sm-short-long-desc}%

```

```

19693 {%
19694   \GlsUseAcrEntryDispStyle{sm-short-long}%
19695 }%
19696 {%
19697   \GlsUseAcrStyleDefs{sm-short-long}%
19698   \renewcommand*{\GenericAcronymFields}{}%
19699   \renewcommand*{\acronymsort}[2]{##2}%
19700   \renewcommand*{\acronymentry}[1]{%
19701     \glstrylong{##1}\space (\acronymfont{\glstryshort{##1}})}%
19702 }
19703 \newacronymstyle{dua}%
19704 {%
19705   \ifdefempty\glscustomtext
19706   {%
19707     \ifglshaslong{\glslabel}%
19708     {%
19709       \glssifplural
19710       {%
19711         \glscapscase
19712         {%
19713           \glstrylongpl{\glslabel}\glssinsert
19714         }%
19715         {%
19716           \Glsentrylongpl{\glslabel}\glssinsert
19717         }%
19718         {%
19719           \mfirstucMakeUppercase
19720             {\glstrylongpl{\glslabel}\glssinsert}%
19721         }%
19722       }%
19723     }%
19724     \glscapscase
19725     {%
19726       \glstrylong{\glslabel}\glssinsert
19727     }%
19728     {%
19729       \Glsentrylong{\glslabel}\glssinsert
19730     }%
19731     {%
19732       \mfirstucMakeUppercase
19733         {\glstrylong{\glslabel}\glssinsert}%
19734     }%
19735   }%
19736 }%
19737 {%
19738   \glsgenentryfmt
19739 }%
19740 }%
19741 {\glscustomtext\glssinsert}%

```

```

19742 }%
19743 {%
19744   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19745   \renewcommand*{\acrfullfmt}[3]{%
19746     \glslink{##1}{##2}{\glsentrylong{##2}##3\space
19747       (\acronymfont{\glsentryshort{##2}})}}%
19748   \renewcommand*{\Acrfullfmt}[3]{%
19749     \glslink{##1}{##2}{\Glsentrylong{##2}##3\space
19750       (\acronymfont{\glsentryshort{##2}})}}%
19751   \renewcommand*{\ACRfullfmt}[3]{%
19752     \glslink{##1}{##2}{%
19753       \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
19754         (\acronymfont{\glsentryshort{##2}})}}}%
19755   \renewcommand*{\acrfullplfmt}[3]{%
19756     \glslink{##1}{##2}{\glsentrylongpl{##2}##3\space
19757       (\acronymfont{\glsentryshortpl{##2}})}}%
19758   \renewcommand*{\Acrfullplfmt}[3]{%
19759     \glslink{##1}{##2}{\Glsentrylongpl{##2}##3\space
19760       (\acronymfont{\glsentryshortpl{##2}})}}%
19761   \renewcommand*{\ACRfullplfmt}[3]{%
19762     \glslink{##1}{##2}{%
19763       \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
19764         (\acronymfont{\glsentryshortpl{##2}})}}}%
19765   \renewcommand*{\glsentryfull}[1]{%
19766     \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
19767   }%
19768   \renewcommand*{\Glsentryfull}[1]{%
19769     \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
19770   }%
19771   \renewcommand*{\glsentryfullpl}[1]{%
19772     \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
19773   }%
19774   \renewcommand*{\Glsentryfullpl}[1]{%
19775     \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
19776   }%
19777   \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19778   \renewcommand*{\acronymsort}[2]{##1}%
19779   \renewcommand*{\acronymfont}[1]{##1}%
19780   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19781 }
19782 \newacronymstyle{dua-desc}%
19783 {%
19784   \GlsUseAcrEntryDisplayStyle{dua}%
19785 }%
19786 {%
19787   \GlsUseAcrStyleDefs{dua}%
19788   \renewcommand*{\GenericAcronymFields}{}%
19789   \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
19790   \renewcommand*{\acronymsort}[2]{##2}%

```

```

19791 }%
19792 \newacronymstyle{footnote}%
19793 {%
19794   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
19795 }%
19796 {%
19797   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19798   \glshyperfirstfalse
19799   \renewcommand*{\genacrfullformat}[2]{%
19800     \protect\firstacronymfont{\glsentryshort{##1}}##2%
19801     \protect\footnote{\glsentrylong{##1}}%
19802   }%
19803   \renewcommand*{\Genacrfullformat}[2]{%
19804     \firstacronymfont{\Glsentryshort{##1}}##2%
19805     \protect\footnote{\glsentrylong{##1}}%
19806   }%
19807   \renewcommand*{\genplacrfullformat}[2]{%
19808     \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
19809     \protect\footnote{\glsentrylongpl{##1}}%
19810   }%
19811   \renewcommand*{\Genplacrfullformat}[2]{%
19812     \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
19813     \protect\footnote{\glsentrylongpl{##1}}%
19814   }%
19815   \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19816   \renewcommand*{\acronymsort}[2]{##1}%
19817   \renewcommand*{\acronymfont}[1]{##1}%
19818   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19819   \renewcommand*{\acrfullfmt}[3]{%
19820     \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
19821       (\glsentrylong{##2})}%
19822   \renewcommand*{\Acrfullfmt}[3]{%
19823     \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
19824       (\glsentrylong{##2})}%
19825   \renewcommand*{\ACRfullfmt}[3]{%
19826     \glslink[##1]{##2}{%
19827       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
19828         (\glsentrylong{##2})}}%
19829   \renewcommand*{\acrfullplfmt}[3]{%
19830     \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
19831       (\glsentrylongpl{##2})}%
19832   \renewcommand*{\Acrfullplfmt}[3]{%
19833     \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
19834       (\glsentrylongpl{##2})}%
19835   \renewcommand*{\ACRfullplfmt}[3]{%
19836     \glslink[##1]{##2}{%
19837       \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
19838         (\glsentrylongpl{##2})}}%
19839   \renewcommand*{\glsentryfull}[1]{%

```

```

19840      \acronymfont{\glentryshort{##1}}\space(\glentrylong{##1}})%
19841 \renewcommand*{\Glsentryfull}[1]{%
19842      \acronymfont{\Glsentryshort{##1}}\space(\glentrylong{##1}})%
19843 \renewcommand*{\glentryfullpl}[1]{%
19844      \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1}})%
19845 \renewcommand*{\Glsentryfullpl}[1]{%
19846      \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1}})%
19847 }
19848 \newacronymstyle{footnote-sc}%
19849 {%
19850   \GlsUseAcrEntryDisplayStyle{footnote}%
19851 }%
19852 {%
19853   \GlsUseAcrStyleDefs{footnote}%
19854   \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
19855   \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19856   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19857 }%
19858 \newacronymstyle{footnote-sm}%
19859 {%
19860   \GlsUseAcrEntryDisplayStyle{footnote}%
19861 }%
19862 {%
19863   \GlsUseAcrStyleDefs{footnote}%
19864   \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
19865   \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19866   \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
19867 }%
19868 \newacronymstyle{footnote-desc}%
19869 {%
19870   \GlsUseAcrEntryDisplayStyle{footnote}%
19871 }%
19872 {%
19873   \GlsUseAcrStyleDefs{footnote}%
19874   \renewcommand*{\GenericAcronymFields}{}%
19875   \renewcommand*{\acronymsort}[2]{##2}%
19876   \renewcommand*{\acronymentry}[1]{%
19877     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})%
19878 }
19879 \newacronymstyle{footnote-sc-desc}%
19880 {%
19881   \GlsUseAcrEntryDisplayStyle{footnote-sc}%
19882 }%
19883 {%
19884   \GlsUseAcrStyleDefs{footnote-sc}%
19885   \renewcommand*{\GenericAcronymFields}{}%
19886   \renewcommand*{\acronymsort}[2]{##2}%
19887   \renewcommand*{\acronymentry}[1]{%
19888     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})%

```



```

19889 }
19890 \newacronymstyle{footnote-sm-desc}%
19891 {%
19892   \GlsUseAcrEntryDispStyle{footnote-sm}%
19893 }%
19894 {%
19895   \GlsUseAcrStyleDefs{footnote-sm}%
19896   \renewcommand*{\GenericAcronymFields}{}%
19897   \renewcommand*{\acronymsort}[2]{##2}%
19898   \renewcommand*{\acronymentry}[1]{%
19899     \glstrylong{##1}\space (\acronymfont{\glstryshort{##1}})}%
19900 }
19901 \newcommand*{\DefineAcronymSynonyms}{%
19902   \let\acs\acrshort
19903   \let\Acs\Acrshort
19904   \let\acsp\acrshortpl
19905   \let\Acsp\Acrshortpl
19906   \let\acl\acrlong
19907   \let\aclp\acrlongpl
19908   \let\Acl\Acrlong
19909   \let\Aclp\Acrlongpl
19910   \let\acf\acrfull
19911   \let\acfp\acrfullpl
19912   \let\Acf\Acrfull
19913   \let\Acfp\Acrfullpl
19914   \let\ac\gls
19915   \let\Ac\Gls
19916   \let\acp\glspl
19917   \let\Acp\Glspl
19918 }
19919 \ifglssacrshortcuts
19920   \DefineAcronymSynonyms
19921 \fi
19922 \newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
19923   \defglstryfmt[#1]{\glsgenentryfmt}%
19924 }
19925 \newcommand*{\DefaultNewAcronymDef}{%
19926   \edef\@do@newglossaryentry{%
19927     \noexpand\newglossaryentry{\the\glslabeltok}%
19928     {%
19929       type=\acronymtype,%
19930       name={\the\glsshorttok},%
19931       sort={\the\glsshorttok},%
19932       text={\the\glsshorttok},%
19933       first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
19934       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
19935       firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
19936                     {\noexpand\expandonce\noexpand\@glo@shortpl}},%
19937       short={\the\glsshorttok},%

```

```

19938     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
19939     long={\the\glslongtok},%
19940     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
19941     description={\the\glslongtok},%
19942     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
19943     \the\glskeylisttok
19944   }%
19945 }%
19946 \let\@org@gls@assign@firstpl\gls@assign@firstpl
19947 \let\@org@gls@assign@plural\gls@assign@plural
19948 \let\@org@gls@assign@descplural\gls@assign@descplural
19949 \def\gls@assign@firstpl##1##2{%
19950   \@@gls@expand@field{##1}{firstpl}{##2}%
19951 }%
19952 \def\gls@assign@plural##1##2{%
19953   \@@gls@expand@field{##1}{plural}{##2}%
19954 }%
19955 \def\gls@assign@descplural##1##2{%
19956   \@@gls@expand@field{##1}{descplural}{##2}%
19957 }%
19958 \do@newglossaryentry
19959 \let\gls@assign@firstpl\@org@gls@assign@firstpl
19960 \let\gls@assign@plural\@org@gls@assign@plural
19961 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
19962 }
19963 \newcommand*{\SetDefaultAcronymStyle}{%
19964   \@for\@gls@type:=\@glsacronymlists\do{%
19965     \SetDefaultAcronymDisplayStyle{\@gls@type}%
19966   }%
19967   \renewcommand{\newacronym}[4][ ]{%
19968     \ifx\@glsacronymlists\@empty
19969       \def\@glo@type{\acronymtype}%
19970       \setkeys{glossentry}{##1}%
19971       \DeclareAcronymList{\@glo@type}%
19972       \SetDefaultAcronymDisplayStyle{\@glo@type}%
19973     \fi
19974     \glskeylisttok{##1}%
19975     \glslabeltok{##2}%
19976     \glsshorttok{##3}%
19977     \glslongtok{##4}%
19978     \newacronymhook
19979     \DefaultNewAcronymDef
19980   }%
19981   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19982 }
19983 \newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
19984 \newcommand*{\acrlinkfootnote}[3]{%
19985   \footnote{\glslink[#1]{#2}{#3}}%
19986 }

```

```

19987 \newcommand*{\acrnolinkfootnote}[3]{%
19988   \footnote{#3}%
19989 }
19990 \newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
19991   \defglsentryfmt[#1]{%
19992     \ifdefempty\glscustomtext
19993     {%
19994       \ifglsused{\glslabel}%
19995       {%
19996         \acronymfont{\glsgenentryfmt}%
19997       }%
19998     {%
19999       \firstacronymfont{\glsgenentryfmt}%
20000       \ifgls hassymbol{\glslabel}%
20001       {%
20002         \expandafter\protect\expandafter\acrfootnote\expandafter
20003         {\@gls@link@opts}{\@gls@link@label}%
20004       {%
20005         \glsifplural
20006         {\glsentrysymbolplural{\glslabel}}%
20007         {\glsentrysymbol{\glslabel}}%
20008       }%
20009     }%
20010   }%
20011 }%
20012 {\glscustomtext\glsinsert}%
20013 }%
20014 }
20015 \newcommand*{\DescriptionFootnoteNewAcronymDef}{%
20016   \edef\@do@newglossaryentry{%
20017     \noexpand\newglossaryentry{\the\glslabeltok}%
20018     {%
20019       type=\acronymtype,%
20020       name={\noexpand\acronymfont{\the\glsshorttok}},%
20021       sort={\the\glsshorttok},%
20022       first={\the\glsshorttok},%
20023       firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20024       text={\the\glsshorttok},%
20025       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20026       short={\the\glsshorttok},%
20027       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20028       long={\the\glslongtok},%
20029       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20030       symbol={\the\glslongtok},%
20031       symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20032       \the\glskeylisttok
20033     }%
20034   }%
20035   \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

20036 \let\org@gl@s@assign@plural\gl@s@assign@plural
20037 \let\org@gl@s@assign@symbolplural\gl@s@assign@symbolplural
20038 \def\gl@s@assign@firstpl##1##2{%
20039   \@@gl@s@expand@field{##1}{firstpl}{##2}%
20040 }%
20041 \def\gl@s@assign@plural##1##2{%
20042   \@@gl@s@expand@field{##1}{plural}{##2}%
20043 }%
20044 \def\gl@s@assign@symbolplural##1##2{%
20045   \@@gl@s@expand@field{##1}{symbolplural}{##2}%
20046 }%
20047 \do@newglossaryentry
20048 \let\gl@s@assign@plural\org@gl@s@assign@plural
20049 \let\gl@s@assign@firstpl\org@gl@s@assign@firstpl
20050 \let\gl@s@assign@symbolplural\org@gl@s@assign@symbolplural
20051 }
20052 \newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
20053   \renewcommand{\newacronym}[4][]{%
20054     \ifx\@gl@s@acronymlists\empty
20055       \def\@glo@type{\acronymtype}%
20056       \setkeys{glossentry}{##1}%
20057       \DeclareAcronymList{\@glo@type}%
20058       \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
20059     \fi
20060     \gl@keylisttok{##1}%
20061     \gl@labeltok{##2}%
20062     \gl@shorttok{##3}%
20063     \gl@longtok{##4}%
20064     \newacronymhook
20065     \DescriptionFootnoteNewAcronymDef
20066   }%
20067   \@for\@gl@s@type:=\@gl@s@acronymlists\do{%
20068     \SetDescriptionFootnoteAcronymDisplayStyle{\@gl@s@type}%
20069   }%
20070   \ifgl@sacrsmallcaps
20071     \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
20072     \renewcommand*{\acrpluralsuffix}{\gl@supacrpluralsuffix}%
20073   \else
20074     \ifgl@sacrsmaller
20075       \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20076     \fi
20077   \fi
20078   \ifgl@sacrdua
20079     \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
20080       can’t both be set}{}%
20081   \fi
20082 }%
20083 \newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
20084   \defgl@sentryfmt[##1]{\gl@genentryfmt}%

```

```

20085 }
20086 \newcommand*{\DescriptionDUANewAcronymDef}{%
20087   \edef\@do@newglossaryentry{%
20088     \noexpand\newglossaryentry{\the\glslabeltok}%
20089     {%
20090       type=\acronymtype,%
20091       name={\the\glslongtok},%
20092       sort={\the\glslongtok},
20093       text={\the\glslongtok},%
20094       first={\the\glslongtok},%
20095       plural={\noexpand\expandonce\noexpand\@glo@longpl},%
20096       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20097       short={\the\glsshorttok},%
20098       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20099       long={\the\glslongtok},%
20100       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20101       symbol={\the\glsshorttok},%
20102       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20103       \the\glskeylisttok
20104     }%
20105   }%
20106   \let\@org@gls@assign@firstpl\gls@assign@firstpl
20107   \let\@org@gls@assign@plural\gls@assign@plural
20108   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20109   \def\gls@assign@firstpl##1##2{%
20110     \@@gls@expand@field{##1}{firstpl}{##2}%
20111   }%
20112   \def\gls@assign@plural##1##2{%
20113     \@@gls@expand@field{##1}{plural}{##2}%
20114   }%
20115   \def\gls@assign@symbolplural##1##2{%
20116     \@@gls@expand@field{##1}{symbolplural}{##2}%
20117   }%
20118   \@do@newglossaryentry
20119   \let\gls@assign@firstpl\@org@gls@assign@firstpl
20120   \let\gls@assign@plural\@org@gls@assign@plural
20121   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20122 }
20123 \newcommand*{\SetDescriptionDUAAcronymStyle}{%
20124   \ifglsacrsmallcaps
20125     \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
20126       can’t both be set}{}%
20127   \else
20128     \ifglsacrsmaller
20129       \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
20130         can’t both be set}{}%
20131     \fi
20132   \fi
20133   \renewcommand{\newacronym}[4][\]{%

```

```

20134 \ifx\@glsacronymlists\@empty
20135 \def\@glo@type{\acronymtype}%
20136 \setkeys{glossentry}{##1}%
20137 \DeclareAcronymList{\@glo@type}%
20138 \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
20139 \fi
20140 \glskeylisttok{##1}%
20141 \glslabeltok{##2}%
20142 \glsshorttok{##3}%
20143 \gslongtok{##4}%
20144 \newacronymhook
20145 \DescriptionDUANewAcronymDef
20146 }%
20147 \@for\@gls@type:=\@glsacronymlists\do{%
20148 \SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
20149 }%
20150 }%
20151 \newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
20152 \defglsentryfmt[#1]{%
20153 \ifdefempty\glscustomtext
20154 {%
20155 \ifglsused{\glslabel}%
20156 {%
20157 \let\gls@org@insert\glsinsert
20158 \let\glsinsert\@empty
20159 \acronymfont{\gls@genentryfmt}\gls@org@insert
20160 }%
20161 {%
20162 \gls@genentryfmt
20163 \ifgls@hassymbol{\glslabel}%
20164 {%
20165 \gls@ifplural
20166 {%
20167 \def\@glo@symbol{\gls@entrysymbolplural{\glslabel}}%
20168 }%
20169 {%
20170 \def\@glo@symbol{\gls@entrysymbol{\glslabel}}%
20171 }%
20172 \space(\protect\firstacronymfont
20173 {\gls@scapscase
20174 {\@glo@symbol}
20175 {\@glo@symbol}
20176 {\mfirstucMakeUppercase{\@glo@symbol}}})%
20177 }%
20178 {}%
20179 }%
20180 }%
20181 {\glscustomtext\glsinsert}%
20182 }%

```

```

20183 }
20184 \newcommand*{\DescriptionNewAcronymDef}{%
20185   \edef\@do@newglossaryentry{%
20186     \noexpand\newglossaryentry{\the\glslabeltok}%
20187     {%
20188       type=\acronymtype,%
20189       name={\noexpand
20190         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
20191       sort={\the\glsshorttok},%
20192       first={\the\glslongtok},%
20193       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20194       text={\the\glsshorttok},%
20195       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20196       short={\the\glsshorttok},%
20197       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20198       long={\the\glslongtok},%
20199       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20200       symbol={\noexpand\@glo@text},%
20201       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20202       \the\glskeylisttok}%
20203   }%
20204   \let\@org@gls@assign@firstpl\gls@assign@firstpl
20205   \let\@org@gls@assign@plural\gls@assign@plural
20206   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20207   \def\gls@assign@firstpl##1##2{%
20208     \@@gls@expand@field{##1}{firstpl}{##2}%
20209   }%
20210   \def\gls@assign@plural##1##2{%
20211     \@@gls@expand@field{##1}{plural}{##2}%
20212   }%
20213   \def\gls@assign@symbolplural##1##2{%
20214     \@@gls@expand@field{##1}{symbolplural}{##2}%
20215   }%
20216   \@do@newglossaryentry
20217   \let\gls@assign@firstpl\@org@gls@assign@firstpl
20218   \let\gls@assign@plural\@org@gls@assign@plural
20219   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20220 }
20221 \newcommand*{\SetDescriptionAcronymStyle}{%
20222   \renewcommand{\newacronym}[4][ ]{%
20223     \ifx\@glsacronymlists\@empty
20224       \def\@glo@type{\acronymtype}%
20225       \setkeys{glossentry}{##1}%
20226       \DeclareAcronymList{\@glo@type}%
20227       \SetDescriptionAcronymDisplayStyle{\@glo@type}%
20228     \fi
20229     \glskeylisttok{##1}%
20230     \glslabeltok{##2}%
20231     \glsshorttok{##3}%

```

```

20232 \glslongtok{##4}%
20233 \newacronymhook
20234 \DescriptionNewAcronymDef
20235 }%
20236 \@for\@gls@type:=\@glsacronymlists\do{%
20237 \SetDescriptionAcronymDisplayStyle{\@gls@type}%
20238 }%
20239 \ifglsacrsmallcaps
20240 \renewcommand{\acronymfont}[1]{\textsc{##1}}
20241 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20242 \else
20243 \ifglsacrsmaller
20244 \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20245 \fi
20246 \fi
20247 }%
20248 \newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
20249 \defglsentryfmt[#1]{%
20250 \ifdefempty\glscustomtext
20251 {%
20252 \let\gls@org@insert\glsinsert
20253 \let\glsinsert\@empty
20254 \ifglsused{\glslabel}%
20255 {%
20256 \acronymfont{\glsgenentryfmt}\gls@org@insert
20257 }%
20258 {%
20259 \firstacronymfont{\glsgenentryfmt}\gls@org@insert
20260 \ifglsashaslong{\glslabel}%
20261 {%
20262 \expandafter\protect\expandafter\acrfootnote\expandafter
20263 {\@gls@link@opts}{\@gls@link@label}%
20264 {%
20265 \glsifplural
20266 {\glsentrylongpl{\glslabel}}%
20267 {\glsentrylong{\glslabel}}%
20268 }%
20269 }%
20270 {}}%
20271 }%
20272 }%
20273 {\glscustomtext\glsinsert}%
20274 }%
20275 }
20276 \newcommand*{\FootnoteNewAcronymDef}{%
20277 \edef\@do@newglossaryentry{%
20278 \noexpand\newglossaryentry{\the\glslabeltok}%
20279 {%
20280 type=\acronymtype,%

```



```

20281     name={\noexpand\acronymfont{\the\glsshorttok}},%
20282     sort={\the\glsshorttok},%
20283     text={\the\glsshorttok},%
20284     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20285     first={\the\glsshorttok},%
20286     firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20287     short={\the\glsshorttok},%
20288     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20289     long={\the\glslongtok},%
20290     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20291     description={\the\glslongtok},%
20292     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20293     \the\glskeylisttok
20294   }%
20295 }%
20296 \let\@org@gls@assign@plural\gls@assign@plural
20297 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20298 \let\@org@gls@assign@descplural\gls@assign@descplural
20299 \def\gls@assign@firstpl##1##2{%
20300   \@gls@expand@field{##1}{firstpl}{##2}%
20301 }%
20302 \def\gls@assign@plural##1##2{%
20303   \@gls@expand@field{##1}{plural}{##2}%
20304 }%
20305 \def\gls@assign@descplural##1##2{%
20306   \@gls@expand@field{##1}{descplural}{##2}%
20307 }%
20308 \do@newglossaryentry
20309 \let\gls@assign@plural\@org@gls@assign@plural
20310 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20311 \let\gls@assign@descplural\@org@gls@assign@descplural
20312 }
20313 \newcommand*\SetFootnoteAcronymStyle{%
20314   \renewcommand{\newacronym}[4][ ]{%
20315     \ifx\@glsacronymlists\@empty
20316       \def\@glo@type{\acronymtype}%
20317       \setkeys{glossentry}{##1}%
20318       \DeclareAcronymList{\@glo@type}%
20319       \SetFootnoteAcronymDisplayStyle{\@glo@type}%
20320     \fi
20321     \glskeylisttok{##1}%
20322     \glslabeltok{##2}%
20323     \glsshorttok{##3}%
20324     \glslongtok{##4}%
20325     \newacronymhook
20326     \FootnoteNewAcronymDef
20327   }%
20328   \@for\@gls@type:=\@glsacronymlists\do{%
20329     \SetFootnoteAcronymDisplayStyle{\@gls@type}%

```

```

20330 }%
20331 \ifglsmallcaps
20332   \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
20333   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20334 \else
20335   \ifglsmaller
20336     \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20337   \fi
20338 \fi
20339 \ifglacrdua
20340   \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
20341     can’t both be set}{}%
20342 \fi
20343 }%
20344 \DeclareRobustCommand*{\glsdoparenifnotempty}[2]{%
20345   \protected@edef\gls@tmp{#1}%
20346   \ifdefempty\gls@tmp
20347     {}%
20348   {%
20349     \ifx\gls@tmp\@gls@default@value
20350     \else
20351       \space (#2{#1})%
20352     \fi
20353   }%
20354 }
20355 \newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
20356   \defglentryfmt[#1]{%
20357     \ifdefempty\glscustomtext
20358     {%
20359       \let\gls@org@insert\glsinsert
20360       \let\glsinsert\@empty
20361       \ifglused{\glslabel}%
20362       {%
20363         \acronymfont{\glsgenentryfmt}\gls@org@insert
20364       }%
20365     {%
20366       \glsgenentryfmt
20367       \ifglshassymbol{\glslabel}%
20368       {%
20369         \glsifplural
20370         {%
20371           \def\@glo@symbol{\glstrysymbolplural{\glslabel}}%
20372         }%
20373         {%
20374           \def\@glo@symbol{\glstrysymbol{\glslabel}}%
20375         }%
20376         \space
20377         (\glscapscase
20378           {\firstacronymfont{\@glo@symbol}}%

```

```

20379         {\firstacronymfont{\@glo@symbol}}}%
20380         {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
20381     }%
20382     {}%
20383 }%
20384 }%
20385 {\glscustomtext\glsinsert}%
20386 }%
20387 }
20388 \newcommand*{\SmallNewAcronymDef}{%
20389 \edef\@do@newglossaryentry{%
20390 \noexpand\newglossaryentry{\the\glslabeltok}%
20391 {%
20392     type=\acronymtype,%
20393     name={\noexpand\acronymfont{\the\glsshorttok}},%
20394     sort={\the\glsshorttok},%
20395     text={\the\glsshorttok},%
20396     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20397     first={\the\glslongtok},%
20398     firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20399     short={\the\glsshorttok},%
20400     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20401     long={\the\glslongtok},%
20402     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20403     description={\noexpand\@glo@first},%
20404     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20405     symbol={\the\glsshorttok},%
20406     symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20407     \the\glskeylisttok
20408 }%
20409 }%
20410 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20411 \let\@org@gls@assign@plural\gls@assign@plural
20412 \let\@org@gls@assign@descplural\gls@assign@descplural
20413 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20414 \def\gls@assign@firstpl##1##2{%
20415 \@@gls@expand@field{##1}{firstpl}{##2}%
20416 }%
20417 \def\gls@assign@plural##1##2{%
20418 \@@gls@expand@field{##1}{plural}{##2}%
20419 }%
20420 \def\gls@assign@descplural##1##2{%
20421 \@@gls@expand@field{##1}{descplural}{##2}%
20422 }%
20423 \def\gls@assign@symbolplural##1##2{%
20424 \@@gls@expand@field{##1}{symbolplural}{##2}%
20425 }%
20426 \@do@newglossaryentry
20427 \let\gls@assign@firstpl\@org@gls@assign@firstpl

```

```

20428 \let\gls@assign@plural\@org@gls@assign@plural
20429 \let\gls@assign@descplural\@org@gls@assign@descplural
20430 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20431 }
20432 \newcommand*{\SetSmallAcronymStyle}{%
20433   \renewcommand{\newacronym}[4][]{%
20434     \ifx\@glsacronymlists\@empty
20435       \def\@glo@type{\acronymtype}%
20436       \setkeys{glossentry}{##1}%
20437       \DeclareAcronymList{\@glo@type}%
20438       \SetSmallAcronymDisplayStyle{\@glo@type}%
20439     \fi
20440     \glskeylisttok{##1}%
20441     \glslabeltok{##2}%
20442     \glsshorttok{##3}%
20443     \glslongtok{##4}%
20444     \newacronymhook
20445     \SmallNewAcronymDef
20446   }%
20447   \@for\@gls@type:=\@glsacronymlists\do{%
20448     \SetSmallAcronymDisplayStyle{\@gls@type}%
20449   }%
20450   \ifglsacrsmallcaps
20451     \renewcommand*{\acronymfont}[1]{\textsc{##1}}
20452     \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20453   \else
20454     \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}
20455   \fi
20456   \ifglsacrdua
20457     \ifglsacrsmallcaps
20458       \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
20459         can’t both be set}{}%
20460     \else
20461       \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
20462         can’t both be set}{}%
20463     \fi
20464   \fi
20465 }%
20466 \newcommand*{\SetDUADisplayStyle}[1]{%
20467   \defglsentryfmt[#1]{\glsgenentryfmt}%
20468 }
20469 \newcommand*{\DUANewAcronymDef}{%
20470   \edef\@do@newglossaryentry{%
20471     \noexpand\newglossaryentry{\the\glslabeltok}%
20472     {%
20473       type=\acronymtype,%
20474       name={\the\glsshorttok},%
20475       text={\the\glslongtok},%
20476       first={\the\glslongtok},%

```

```

20477 plural={\noexpand\expandonce\noexpand\@glo@longpl},%
20478 firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20479 short={\the\glsshorttok},%
20480 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20481 long={\the\glslongtok},%
20482 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20483 description={\the\glslongtok},%
20484 descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20485 symbol={\the\glsshorttok},%
20486 symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20487 \the\glskeylisttok
20488 }%
20489 }%
20490 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20491 \let\@org@gls@assign@plural\gls@assign@plural
20492 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20493 \let\@org@gls@assign@descplural\gls@assign@descplural
20494 \def\gls@assign@firstpl##1##2{%
20495   \@gls@expand@field{##1}{firstpl}{##2}%
20496 }%
20497 \def\gls@assign@plural##1##2{%
20498   \@gls@expand@field{##1}{plural}{##2}%
20499 }%
20500 \def\gls@assign@symbolplural##1##2{%
20501   \@gls@expand@field{##1}{symbolplural}{##2}%
20502 }%
20503 \def\gls@assign@descplural##1##2{%
20504   \@gls@expand@field{##1}{descplural}{##2}%
20505 }%
20506 \do@newglossaryentry
20507 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20508 \let\gls@assign@plural\@org@gls@assign@plural
20509 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20510 \let\gls@assign@descplural\@org@gls@assign@descplural
20511 }
20512 \newcommand*\SetDUASstyle{%
20513   \renewcommand{\newacronym}[4][\]{%
20514     \ifx\@glsacronymlists\@empty
20515       \def\@glo@type{\acronymtype}%
20516       \setkeys{glossentry}{##1}%
20517       \DeclareAcronymList{\@glo@type}%
20518       \SetDUADisplayStyle{\@glo@type}%
20519     \fi
20520     \glskeylisttok{##1}%
20521     \glslabeltok{##2}%
20522     \glsshorttok{##3}%
20523     \glslongtok{##4}%
20524     \newacronymhook
20525     \DUANewAcronymDef

```

```

20526 }%
20527 \@for\@gls@type:=\@glsacronymlists\do{%
20528   \SetDUADisplayStyle{\@gls@type}%
20529 }%
20530 }
20531 \newcommand*{\SetAcronymStyle}{%
20532   \SetDefaultAcronymStyle
20533   \ifglsacrdescription
20534     \ifglsacrfootnote
20535       \SetDescriptionFootnoteAcronymStyle
20536     \else
20537       \ifglsacrdua
20538         \SetDescriptionDUAAcronymStyle
20539       \else
20540         \SetDescriptionAcronymStyle
20541       \fi
20542     \fi
20543   \else
20544     \ifglsacrfootnote
20545       \SetFootnoteAcronymStyle
20546     \else
20547       \ifthenelse{\boolean{glsacrsmalldcaps}\OR
20548         \boolean{glsacrsmaller}}{%
20549         {%
20550           \SetSmallAcronymStyle
20551         }%
20552         {%
20553           \ifglsacrdua
20554             \SetDUASStyle
20555           \fi
20556         }%
20557       \fi
20558     \fi
20559 }
20560 \SetAcronymStyle
20561 \newcommand*{\SetCustomDisplayStyle}[1]{%
20562   \defglsentryfmt[#1]{\glsentryfmt}%
20563 }
20564 \newcommand*{\CustomAcronymFields}{%
20565   name={\the\glsshorttok},%
20566   description={\the\glslongtok},%
20567   first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
20568   firstplural={\acrfullformat
20569     {\noexpand\glsentrylongpl{\the\glslabeltok}}}%
20570     {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
20571   text={\the\glsshorttok},%
20572   plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
20573 }
20574 \newcommand*{\CustomNewAcronymDef}{%

```

```

20575 \protected@edef\@do@newglossaryentry{%
20576   \noexpand\newglossaryentry{\the\glslabeltok}%
20577   {%
20578     type=\acronymtype,%
20579     short={\the\glsshorttok},%
20580     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20581     long={\the\glslongtok},%
20582     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20583     user1={\the\glsshorttok},%
20584     user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
20585     user3={\the\glslongtok},%
20586     user4={\the\glslongtok\noexpand\acrpluralsuffix},%
20587     \CustomAcronymFields,%
20588     \the\glskeylisttok
20589   }%
20590 }%
20591 \@do@newglossaryentry
20592 }
20593 \newcommand*\@SetCustomStyle{%
20594   \renewcommand{\newacronym}[4][\@
20595     \ifx\@glsacronymlists\@empty
20596       \def\@glo@type{\acronymtype}%
20597       \setkeys{glossentry}{##1}%
20598       \DeclareAcronymList{\@glo@type}%
20599       \SetCustomDisplayStyle{\@glo@type}%
20600     \fi
20601     \glskeylisttok{##1}%
20602     \glslabeltok{##2}%
20603     \glsshorttok{##3}%
20604     \glslongtok{##4}%
20605     \newacronymhook
20606     \CustomNewAcronymDef
20607   }%
20608   \@for\@gls@type:=\@glsacronymlists\do{%
20609     \SetCustomDisplayStyle{\@gls@type}%
20610   }%
20611 }
20612 \RequirePackage{glossary-hypernav}
20613 \@gls@loadlist
20614 \@gls@loadlong
20615 \@gls@loadsuper
20616 \@gls@loadtree
20617 \ifx\@glossary@default@style\relax
20618 \else
20619   \setglossarystyle{\@glossary@default@style}
20620 \fi
20621 \newcommand*\@showgloparent}[1]{%
20622   \expandafter\show\csname glo@glsdetoklabel{#1}@parent\endcsname
20623 }

```

```

20624 \newcommand*{\showglolevel}[1]{%
20625   \expandafter\show\csname glo@glstetoklabel{#1}@level\endcsname
20626 }
20627 \newcommand*{\showglotext}[1]{%
20628   \expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
20629 }
20630 \newcommand*{\showgloplural}[1]{%
20631   \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
20632 }
20633 \newcommand*{\showglofirst}[1]{%
20634   \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
20635 }
20636 \newcommand*{\showglofirstpl}[1]{%
20637   \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
20638 }
20639 \newcommand*{\showglotype}[1]{%
20640   \expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
20641 }
20642 \newcommand*{\showglocounter}[1]{%
20643   \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
20644 }
20645 \newcommand*{\showglouserii}[1]{%
20646   \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
20647 }
20648 \newcommand*{\showglouseriii}[1]{%
20649   \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
20650 }
20651 \newcommand*{\showglouseriv}[1]{%
20652   \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
20653 }
20654 \newcommand*{\showglouseriv}[1]{%
20655   \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
20656 }
20657 \newcommand*{\showglouserv}[1]{%
20658   \expandafter\show\csname glo@glstetoklabel{#1}@userv\endcsname
20659 }
20660 \newcommand*{\showglouservi}[1]{%
20661   \expandafter\show\csname glo@glstetoklabel{#1}@uservi\endcsname
20662 }
20663 \newcommand*{\showglongame}[1]{%
20664   \expandafter\show\csname glo@glstetoklabel{#1}@name\endcsname
20665 }
20666 \newcommand*{\showglodesc}[1]{%
20667   \expandafter\show\csname glo@glstetoklabel{#1}@desc\endcsname
20668 }
20669 \newcommand*{\showglodescplural}[1]{%
20670   \expandafter\show\csname glo@glstetoklabel{#1}@descplural\endcsname
20671 }
20672 \newcommand*{\showglosort}[1]{%

```



```

20673 \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
20674 }
20675 \newcommand*{\showglosymbol}[1]{%
20676 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
20677 }
20678 \newcommand*{\showglosymbolplural}[1]{%
20679 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
20680 }
20681 \newcommand*{\showgloshort}[1]{%
20682 \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
20683 }
20684 \newcommand*{\showglolong}[1]{%
20685 \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
20686 }
20687 \newcommand*{\showgloindex}[1]{%
20688 \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
20689 }
20690 \newcommand*{\showgloflag}[1]{%
20691 \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
20692 }
20693 \newcommand*{\showgloloclist}[1]{%
20694 \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
20695 }
20696 \newcommand*{\showglofield}[2]{%
20697 \csshow{glo@\glsdetoklabel{#1}@#2}%
20698 }
20699 \newcommand*{\showacronymlists}{%
20700 \show@glsacronymlists
20701 }
20702 \newcommand*{\showglossaries}{%
20703 \show@glo@types
20704 }
20705 \newcommand*{\showglossaryin}[1]{%
20706 \expandafter\show\csname @glotype@#1@in\endcsname
20707 }
20708 \newcommand*{\showglossaryout}[1]{%
20709 \expandafter\show\csname @glotype@#1@out\endcsname
20710 }
20711 \newcommand*{\showglossarytitle}[1]{%
20712 \expandafter\show\csname @glotype@#1@title\endcsname
20713 }
20714 \newcommand*{\showglossarycounter}[1]{%
20715 \expandafter\show\csname @glotype@#1@counter\endcsname
20716 }
20717 \newcommand*{\showglossaryentries}[1]{%
20718 \expandafter\show\csname glolist@#1\endcsname
20719 }
20720 \csname ifglscompatible-2.07\endcsname
20721 \RequirePackage{glossaries-compatible-207}

```

20722\fi

Glossary

`makeindex` An indexing application [10](#), [14](#), [30](#), [33](#), [189](#), [200](#)

`xindy` An flexible indexing application with multilingual support written in Perl [10](#), [14](#), [30](#), [33](#), [189](#), [200](#)

Change History

1.01 (2007-05-17)	numberline: numberline option added . . . 8
General: Added range facility in format key 121	
\writeist: Added spaces after \delimN and \delimR in ist file 169	
1.04 (2007-08-03)	1.12 (2008-03-08)
General: Added \glstextformat 105	\@GLSpl: now uses
1.05 (2007-08-10)	\glentrydescplural and
\glossarysection: added \@mkboth to \glossarysection 47	\glentrysymbolplural instead of
\gls@defglossaryentry: Changed the default value of the sort key to just the value of the name key 89	\glentrydesc and
1.07 (2007-09-13)	\glentrysymbol 136
\@gls@link: fixed bug caused by \theglentrycounter setting the page number too soon 119	\@Glspl@: now uses
\glsadd: fixed bug caused by \theglentrycounter setting the page number too soon 166	\glentrydescplural and
1.08 (2007-10-13)	\glentrysymbolplural instead of
General: Added babel support 41	\glentrydesc and
listgroup: changed listgroup style to use \glsgetgrouptitle 286	\glentrysymbol 134
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle 287	General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget) 130
1.1 (2008-02-22)	descriptionplural: new 70
\@glossarysection: numbered sections and auto label added 48	\gls@defglossaryentry: Changed default first plural to be first key with s appended (was text key with s appended) 89
\@gls@tmpb: changed \toksdef to \newtoks 123	descriptionplural support added 89
\@gls@toc: numberline added 49	symbolplural support added 89
\@p@glossarysection: numbered sections and auto label added 48	\Glsentrydescplural: New 159
General: amsgen now loaded (\new@ifnextchar needed) 4	\glentrydescplural: New 159
translate: translate option added 28	\Glsentrysymbolplural: New 160
\setglossarysection: new 48	\glentrysymbolplural: New 160
numberedsection: numberedsection package option added 9	\SetDescriptionFootnoteAcronymStyle: Added \protect before \footnote and \glslink 252
	\SetFootnoteAcronymStyle: Added \protect before \footnote and \glslink 258
	symbolplural: new 71

1.13 (2008-05-10)	
General: fixed bug that ignored 3rd parameter	137-144
\ACRfullpl: new	232
\Acrfullpl: new	231
\acrfullpl: new	231
\acrpluralsuffix: New	229
\gls@defglossaryentry: Changed default first value	89
Changed default firstplural value	89
Removed restriction on only using \newglossaryentry in the preamble	94
\newacronym: Removed restriction on only using \newacronym in the preamble	229
1.14 (2008-06-17)	
\@gls@hypergroup: new	281
General: added nonumberlist key to \printglossary	216
added numberedsection key to \printglossary	214
\firstacronymfont: new	232
\glsautoprefix: new	9
\glsnavhyperlink: changed \edef to \protected@edef	280
\glsnavhypertarget: added write to aux file	280
\glsnavigation: changed to only use labels for groups that are present ..	282
1.15 (2008-08-15)	
\@gls@link: added \glslabel	119
\gls@defglossaryentry: check for \@glo@first in description	93
check for \@glo@text in symbol	93
\gls@hypergroup: new	281
\glsnavhypertarget: added check if rerun required	280
\glssettoctitle: new	40
\printglossary: changed the way the TOC title is set	200
1.16 (2008-08-27)	
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	133
\@GLSpl: Test glossary type is \acronymtype in addition to checking if footnote option has been used	136
\@Gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	132
\@Glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	135
\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	132
\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used	136
\@glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	134
\@glstarget: raised the hypertarget so the target text doesn't scroll off the top of the page	130
\gls@defglossaryentry: Changed def to let	89
1.17 (2008-12-26)	
\@do@esc@wrglossary: new	193
\@do@seeglossary: new	197
\@glo@storeentry: new	95
\@gls@glossary: changed definition to use \index instead of \@index	190
\@glsdefaultplural: new	75
\@glsdefaultsort: new	75
\@glshypernumber: new	226
\@glsnoname: new	74
\@glsnonextpages: new	216
General: added xindy support	30
parent: new	72
see: new	72
\gls@defglossaryentry: added nonumberlist key	89
added parent key	89
added see key	89
Stored main part of entry format when entry is defined	94
\gls@suffixF: new	45
\gls@suffixFF: new	45
\gls@wrglossary: modified to allow for xindy support	190

\glshyperlink: new	166	\SetDescriptionFootnoteAcronymStyle:	
\glshypernumber: modified to allow		changed \acronymfont to use	
material to be attached to location .	226	\textsmaller instead of \smaller	252
\glshnavhyperlink: replaced		\SetFootnoteAcronymStyle: changed	
\hyperlink to \@glslink	280	\acronymfont to use \textsmaller	
\glshnavhypertarget: replaced		instead of \smaller	258
\hypertarget to \@glstarget ...	280	\SetSmallAcronymStyle: changed	
\glsssee: new	198	\acronymfont to use \textsmaller	
\glssseeformat: new	198	instead of \smaller	261
\glssSetSuffixF: new	45	1.46 (2021-09-20)	
\glssSetSuffixFF: new	45	\@printglossary: changed \xdef to	
\ifglsxindy: new	30	\protected\xdef	203
\istfilename: added xindy support ...	44	General: changed \edef to	
\newglossarystyle: made		\protected@edef	394
\newglossarystyle long	225	\glss@defglossaryentry: changed	
\nopostdesc: new	43	\xdef to \protected\xdef ..	90, 91, 93
nonumberlist: new	72	\glssaddprotectedpagefmt: changed	
\printglossary: added check to		\eappto to \protected@eappto ..	192
determine if \printglossary is		2.01 (2009 May 30)	
already defined	200	\@glss@link: moved \@do@wrglossary	
added print language to aux file	200	before term is displayed to prevent	
order: order package option added	30	unwanted whatsit	120
\writeist: added xindy support	169	\forallglossaries: replaced	
1.18 (2009-01-14)		\ifthenelse with \ifx	59
\@glss@loadlist: new	11	\forallglsentries: replaced \ifthenelse	
\@glss@loadlong: new	10	with \ifx	59
\@glss@loadsuper: new	11	\glssdefmain: new	17
\@glss@loadtree: new	11	\glssdescwidth: changed \linewidth to	
\glss@defglossaryentry: Changed		\hsize	288, 310
default value of sort to		\glsslistdottedwidth: changed	
\@glssdefaultsort	89	\linewidth to \hsize	288
moved sort sanitization to		\glsspagelistwidth: changed	
\newglossaryentry	93	\linewidth to \hsize	288, 310
\glstarget: new	219	nomain: added nomain package option .	17
\oldacronym: new	228	\writeist: removed item_02 - no such	
nolist: new	11	makeindex key	174
nolong: new	11	2.02 (2007-07-13)	
sort: moved sanitization to		\@printglossary: suppressed warning	
\newglossaryentry	70	globally rather than locally	203
nostyles: new	11	2.02 (2009-07-13)	
nosuper: new	11	\glossarysection: changed \@mkboth	
notree: new	11	to \glossarymark	47
1.19 (2009-03-02)		\glsglossarymark: New	47
\glsclearpage: new	49	2.03 (2009-09-23)	
\glssdisp: new	136	\@GLS@: Added check for hyperfirst	133
\SetDescriptionAcronymStyle:		\@GLSpl: Added check for hyperfirst ...	136
changed \acronymfont to use		\@Gls@: Added check for hyperfirst	132
\textsmaller instead of \smaller	256	\@Glspl@: Added check for hyperfirst ..	135
		\@Gls@: Added check for hyperfirst	132

\@gls@link: new	118	\glsentryuseriv: new	162
\@gls@link: added \leavevmode	119	\Glsentryuseriv: new	163
Moved entry existence check to avoid		\glsentryuseriv: new	163
duplicate code	119	\Glsentryuseriv: new	163
\@glsdisp: Added check for hyperfirst .	136	\glsentryuseriv: new	163
\@glspl0: Added check for hyperfirst ..	134	\ns@newglossary: added check to	
\gls glossarymark: Added check to see		determine if \@gls@<type>@display	
if it's already defined	47	and \@gls@<type>@displayfirst	
hyperfirst: new	29	have been defined.	67
2.04 (2009-11-10)		\SetAcronymLists: new	20
\@GLS@: Changed test to check if glossary		\SetDefaultAcronymDisplayStyle:	
type has been identified as a list of		new	247
acronyms	133	\SetDefaultAcronymStyle: new	249
\@GLSp1@: Changed test to check if		\SetDescriptionAcronymDisplayStyle:	
glossary type has been identified as a		new	254
list of acronyms	136	\SetDescriptionDUAAcronymDisplayStyle:	
\@GLS@: Changed test to check if glossary		new	252
type has been identified as a list of		\SetDescriptionFootnoteAcronymDisplayStyle:	
acronyms	132	new	250
\@GLSp10@: Changed test to check if		\SetDUADisplayStyle: new	262
glossary type has been identified as a		\SetFootnoteAcronymDisplayStyle:	
list of acronyms	135	new	256
\@glossaryentryfield: new	95	\SetSmallAcronymDisplayStyle: new	259
\@glossarysubentryfield: new	95	2.05 (2010-02-06)	
\@gls@: Changed test to check if glossary		\@glsdisp: Added closing brace. Patch	
type has been identified as a list of		provided by Sergiu Dotenco	136
acronyms	132	Removed spurious brace. Patch	
\@glsacronymlists: new	19	provided by Sergiu Dotenco	137
\@glsdisp: Changed test to check if		\writeist: Added \string before	
glossary type has been identified as a		opening and closing braces. Patch	
list of acronyms	136	provided by Segiu Dotenco	174
\@glspl0@: Changed test to check if		2.06 (2010-06-14)	
glossary type has been identified as a		\altnewglossary: new	68
list of acronyms	134	\CustomAcronymFields: new	264
\@newglossaryentryposthook: new ..	94	\CustomNewAcronymDef: new	265
\@newglossaryentryprehook: new ...	94	\SetCustomDisplayStyle: new	264
acronymlists: new	20	\SetCustomStyle: new	265
\DeclareAcronymList: new	19	2.07 (2010-07-10)	
\DefineAcronymSynonyms: new	245	General: glsadd format key stored in	
\@gls@defglossaryentry: added user1-6		\@glsnumberformat (was mistakenly	
keys	89	stored in \@glo@format)	166
\glsadd: fixed bug that ignored counter	166	3.0 (2010-07-12)	
\Glsentryuseri: new	162	\@makeglossary: Added check for	
\glsentryuseri: new	162	savewrites	179
\Glsentryuserii: new	162	\gls@wrglossary: modified to take into	
\glsentryuserii: new	162	account savewrites	190
\Glsentryuseriii: new	162	3.0 (2010/03/31)	
\glsentryuseriii: new	162	\@set@glo@numformat: added 4th	
\Glsentryuseriv: new	163	argument	121

3.0 (2011-04-02)

\@do@esc@wrglossary: added check for hyper location prefix	195	\gls@codepage: replaced \ifundefined with \ifcsundef ..	31
modified to use new format	193	\gls@defglossaryentry: added \@gls@defsort	93
\@glossarysec: replaced \ifundefined with \ifcsundef ...	8	added short and long keys	90
\@do@seeglossary: Sanitize and escape cross-referencing information	197	replaced \@ifundefined with \ifcsundef	90
\@gls@counterwithin: new	13	\gls@doclearpage: replaced \ifundefined with \ifcsundef ..	49
\@gls@ifinlist: new	50	\glsadd: added \@gls@saveentrycounter	166
\@gls@link: added \@gls@saveentrycounter	120	\GlsAddXdyCounters: new	50
added \@gls@setsort	120	\glseentrycounterlabel: new	218
\@gls@saveentrycounter: new	120	\glseentryitem: new	218
\@gls@setupsort@def: new	15	\Glsentrylong: new	164
\@gls@setupsort@standard: new	14	\glseentrylong: new	163
\@gls@setupsort@use: new	15	\Glsentrylongpl: new	164
\@gls@xdy@locationlist: new	53	\glseentrylongpl: new	164
\@glslink: replaced \@ifundefined with \ifcsundef	130	\Glsentryshort: new	163
\@glsnextpages: new	216	\glseentryshort: new	163
\@print@glossary: replaced \ifundefined with \ifcsundef .	203	\Glsentryshorttpl: new	163
\@printglossary: added \currentglossary	202	\glseentryshorttpl: new	163
added \glsnextpages	202	\glsgetgrouptitle: replaced \ifundefined with \ifcsundef .	222
make toctitle default to title	202	\glsglossarymark: replaced \ifundefined with \ifcsundef ..	47
\@xdy@attributelist: new	50	\glshyperlink: changed default from \glseentryname to \glseentrytext	166
General: added prefix to hyperlink	227	\glshypernumber: replaced \ifundefined with \ifcsundef .	226
etoolbox now loaded	4	\glsnumberformat: replaced \ifundefined with \ifcsundef ..	45
replaced \@ifundefined with \ifcsundef	39, 41, 116, 214	\glseentry: new	217
\acrfootnote: new	249	\glseentrysubentrycounter: new ...	217
\ACRfull: added starred version	231	\glseentryitem: hyperlink uses \glseentryformat instead of \glseentryname	199
\Acrfull: added starred version	230	\glseentryformat: new	199
\acrfull: added starred version	229	\glssortnumberfmt: new	15
\ACRfullpl: added starred version ...	232	\glsestepentry: new	217
\Acrfullpl: added starred version ...	231	\glsestepsubentry: new	217
\acrfullpl: added starred version ...	231	\glssubentrycounterlabel: new ...	218
\acrlinkfootnote: new	249	\glssubentryitem: new	218
\acrlinkfootnote: new	249	theglossary: replaced \@ifundefined with \ifcsundef	218
savewrites: new	34	short: new	74
see: added \@glo@seeautonumberlist	72	shortplural: new	74
seeautonumberlist: new	10	\ifglossaryexists: replaced \ifundefined with \ifcsundef ..	60
\glossarysection: replaced \ifundefined with \ifcsundef ..	47		
\glossarystyle: replaced \ifundefined with \ifcsundef .	224		

<code>\ifglentryexists</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	60
<code>\istfile</code> : deprecated	189
<code>glossaryentry</code> : new	12
<code>glossarysubentry</code> : new	13
<code>\newglossaryentry</code> : replaced	
<code>\DeclareRobustCommand</code> with	
<code>\newrobustcmd</code>	77
<code>\newglossarystyle</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	225
<code>\ns@newglossary</code> : added	
<code>\@gls@defsortcount</code>	68
replaced <code>\@ifundefined</code> with	
<code>\ifcsundef</code>	67
<code>entrycounter</code> : new	13
<code>\oldacronym</code> : replaced <code>\@ifundefined</code>	
with <code>\ifcsundef</code>	228
compatible-2.07: compatible-2.07	
option added	34
<code>long</code> : new	74
<code>longplural</code> : new	74
<code>nonumberlist</code> : now boolean	72
<code>sort</code> : new	14
<code>counter</code> : replaced <code>\@ifundefined</code> with	
<code>\ifcsundef</code>	71
<code>counterwithin</code> : new	13
<code>\printglossary</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	200
<code>\SetDescriptionFootnoteAcronymDisplayStyle</code>	
expanded options link options	250
<code>\setentrycounter</code> : added optional	
argument	223
<code>\showacronymlists</code> : new	270
<code>\showglocounter</code> : new	267
<code>\showglodesc</code> : new	269
<code>\showglodescplural</code> : new	269
<code>\showglofirst</code> : new	267
<code>\showglofirstpl</code> : new	267
<code>\showgloflag</code> : new	270
<code>\showgloindex</code> : new	270
<code>\showglolevel</code> : new	266
<code>\showglolevel</code> : new	268
<code>\showgloparent</code> : new	266
<code>\showgloplural</code> : new	267
<code>\showglosort</code> : new	269
<code>\showglossaries</code> : new	270
<code>\showglossarycounter</code> : new	271
<code>\showglossaryentries</code> : new	271
<code>\showglossaryin</code> : new	271
<code>\showglossaryout</code> : new	271
<code>\showglossarytitle</code> : new	271
<code>\showglosymbol</code> : new	269
<code>\showglosymbolplural</code> : new	269
<code>\showglotext</code> : new	266
<code>\showglotype</code> : new	267
<code>\showglouserii</code> : new	267
<code>\showglouseriii</code> : new	268
<code>\showglouseriv</code> : new	268
<code>\showglouserv</code> : new	268
<code>\showglouservi</code> : new	268
<code>subentrycounter</code> : new	13
<code>\writeist</code> : added xindy-only macro	
definitions to glossary open tag	171
modified to support new format	169
3.01 (2011-04-12)	
<code>\@glswritefiles</code> : added check for	
empty glossaries	189
General: made robust	133
<code>\ACRfull</code> : made robust	231
<code>\Acrfull</code> : made robust	230
<code>\acrfull</code> : made robust	229
<code>\acrfullformat</code> : removed	
<code>\acronymfont</code> as it should already be	
set in the second argument.	230
<code>\ACRfullpl</code> : made robust	232
<code>\Acrfullpl</code> : made robust	231
<code>\acrfullpl</code> : made robust	231
<code>\ACRlong</code> : made robust	154
<code>\Acrlong</code> : made robust	153
<code>\acrlong</code> : made robust	153
<code>\ACRlongpl</code> : made robust	156
<code>\Acrlongpl</code> : made robust	155
<code>\acrlongpl</code> : made robust	155
<code>\ACRshort</code> : made robust	150
<code>\Acrshort</code> : made robust	150
<code>\acrshort</code> : made robust	149
<code>\ACRshortpl</code> : made robust	152
<code>\Acrshortpl</code> : made robust	152
<code>\acrshortpl</code> : made robust	151
<code>\Gls</code> : made robust	132
<code>\glsadd</code> : made robust	166
<code>\glsaddall</code> : made robust	167
<code>\GLSdesc</code> : made robust	142
<code>\Glsdesc</code> : made robust	141
<code>\glsdesc</code> : made robust	141
<code>\GLSdescplural</code> : made robust	143
<code>\Glsdescplural</code> : made robust	142

\glsdescplural: made robust	142	\do@wr@glossary code into	
\glsfirst: made robust	138	\do@wrglossary	191
\GLSfirstplural: made robust	140	\@gls@missingnumberlist: new	75
\Glsfirstplural: made robust	140	\@gls@writefiles: added check for	
\glsfirstplural: made robust	140	existence of token in case	
\glslink: made robust	118	\makeglossaries has been	
\GLSname: made robust	141	omitted	189
\Glsname: made robust	141	\@printglossary: add a way to fetch	
\glsname: made robust	140	current entry label	203
\GLSpl: made robust	135	savenumberlist: new	10
\Glspl: made robust	134	ucmark: new	12
\glspl: made robust	134	\gls@defglossaryentry: added	
\GLSplural: made robust	139	numberlist element	93
\GLSsymbol: made robust	143	\gls@save@numberlist: new	199
\Glsymbol: made robust	143	\gls@wrglossary: added check for	
\glssymbol: made robust	143	glossary file defined	191
\GLSsymbolplural: made robust	144	\glsdisplaynumberlist: new	164
\Glsymbolplural: made robust	144	\glentrycounter: set default value ..	120
\glssymbolplural: made robust	144	\Glentryfull: fixed bug (replaced	
\Glstext: made robust	138	\glentryshortpl with	
\glstext: made robust	137	\glentryshort)	164
\GLSuseri: made robust	145	\glentryfullpl: fixed bug (replaced	
\Glsuseri: made robust	145	\glentryshort with	
\glsuseri: made robust	145	\glentryshortpl)	164
\GLSuserii: made robust	146	\glentrynumberlist: new	164
\Glsuserii: made robust	146	\glsmoveentry: new	95
\glsuserii: made robust	145	\glsresetsubentrycounter: new ...	217
\GLSuseriii: made robust	147	\ifglshaschildren: new	62
\Glsuseriii: made robust	146	\ifglshasparent: new	62
\glsuseriii: made robust	146	\makeglossaries: added list parser ..	183
\GLSuseriv: made robust	147	indexonlyfirst: new	29
\Glsuseriv: made robust	147	\renewglossarystyle: new	225
\glsuseriv: made robust	147	\showglossaryentries: fixed misspelt	
\GLSuserv: made robust	148	command	271
\Glsuserv: made robust	148	\SmallNewAcronymDef: fixed broken	
\glsuserv: made robust	148	short and long plural	260
\GLSuservi: made robust	149	3.03 (2012/09/21)	
\Glsuservi: made robust	149	\@gls@sanitizesort: new	23
\glsuservi: made robust	149	\@gls@setupsort@standard: used	
3.02 (2012-05-19)		\@gls@sanitizesort	14
\glsnumlistlastsep: new	165	\@printglossary: allow title to override	
\glsnumlistsep: new	165	default toctitle	201
3.02 (2012-05-21)		General: allow title to set toctitle	214
\@do@wrglossary: changed		\glsinlinedescformat: new	284
\@glslocref to		\glsinlineemptydescformat: new ..	284
\theglentrycounter	196	\glsinlinenameformat: new	284
\do@wrglossary: changed		\glsinlinepostchild: new	284
\do@wr@glossary to test for		\glsinlinesubdescformat: new	284
indexonlyfirst option; put old		\glsinlinesubnameformat: new	284

\glspostinline: replaced “.” with	unwanted whatsit occurring here. . .	203
\glspostdescription	General: Added check for doc package . . .	4
list: added check for glsnogroupskip .	added datatool-base as a required	
altlongragged4col: added check for	package	4
glsnogroupskip	added local key	116
altsuperragged4col: added check for	\gls@Alphpage: new	192
glsnogroupskip	\gls@alphpage: new	192
alttree: added check for	\gls@disablepagerefexpansion: new	191
glsnogroupskip	\gls@numberpage: new	192
index: added check for glsnogroupskip	\gls@protected@pagefmts: new . . .	191
325	\gls@romanpage: new	192
nogroupskip: new	\glsdefmain: added check for doc	
12	package	17
long: added check for glsnogroupskip .	\glsorg@endtheglossary: new	5
289	\glsorg@theglossary: new	5
long3col: added check for	\PrintChanges: new	5
glsnogroupskip	3.05 (2013-04-21)	
291	\@@do@esc@wrglossary: add Roman	
long4col: added check for	case. Fixed bugs in the else	
glsnogroupskip	statements	195
292	\@gls@link: added check for	
longragged: added check for	“nohypertypes”	119
glsnogroupskip	mcolalttree: replaced ‘2’ with	
300	\glsmcols	309
longragged3col: added check for	mcolindex: replaced ‘2’ with \glsmcols	305
glsnogroupskip	mcolindexspannav: replaced ‘2’ with	
301	\glsmcols	306
no postdot: new	mcoltree: replaced ‘2’ with \glsmcols	306
12	mcoltreename: replaced ‘2’ with	
tree: added check for glsnogroupskip .	\glsmcols	308
326	mcoltreesspannav: replaced ‘2’ with	
treename: added check for	\glsmcols	307
glsnogroupskip	\gls@protected@pagefmts: added	
328	Roman to list	191
super: added check for glsnogroupskip	\gls@Romanpage: new	192
311	\glsgetgrouplabel: fixed bug (typo in	
super3col: added check for	\equal)	223
glsnogroupskip	\nopostdesc: made robust	43
313	3.05 (2013/04/21)	
super4col: added check for	\@gls@nohyperlist: new	20
glsnogroupskip	\GlsDeclareNoHyperList: new	20
315	nohypertypes: new	20
superragged: added check for	3.06 (2013/06/17)	
glsnogroupskip	\@xdy@main@language: Changed back to	
318	using \language	30
superragged3col: added check for	\findrootlanguage: Obsoleted	57
glsnogroupskip	3.07 (2013-07-05)	
320	\@gls@link: fixed bug that failed to find	
3.04 (2012-11-11)	entry in list	119
altlist: replaced \newline with		
paragraph break		
286		
3.04 (2012-11-18)		
\@@do@@wrglossary: changed		
\theglsentrycounter back to		
\@glslocref		196
\@@do@esc@wrglossary: modified to		
compensate for possible incorrect		
page number		195
\@gls@escbsdq: unsanitize		
\gls@numberpage, \gls@alphpage,		
\gls@Alphpage and		
\gls@romanpage		122
\@print@glossary: Moved aux write to		
end of document to prevent		

\glossarypreamble: modified to work with \setglossarypreamble	46	\glsaddallunused: new	167
\gls@docclearpage: added check for openright	49	\glsexpandfields: new	77
\glspostdescription: Added spacefactor code	12	\glsnoexpandfields: new	77
\GlsSetXdyCodePage: Added check for fontspec	58	\glssee: made robust	198
\SetDescriptionAcronymDisplayStyle: now using \glsdoparenifnotempty	254	\glsseeformat: made robust	198
\setglossarypreamble: new	46	\glsseeitem: made robust	199
3.08a (2013-08-30)		\glsseelist: made robust	198
list: updated list style to use \glossentry and \subglossentry	285	\ifglsdescsuppressed: new	63
listdotted: updated listdotted style to use \glossentry and \subglossentry	287	\ifglshasdesc: new	63
altlist: updated altlist style to use \glossentry and \subglossentry	286	\ifglshassymbol: new	63
inline: updated inline style to use \glossentry and \subglossentry	283	altlongragged4col: updated to use \glossentry and \subglossentry	303
3.08a (2013-09-28)		alttree: updated to use \glossentry and \subglossentry	330
\@glo@storeentry: no longer need to check for special characters in any of the fields other than sort	96	index: added paragraph break at end of environment	324
updated for \glossentry	96	updated to use \glossentry and \subglossentry	324
\@glossaryentryfield: switched to \glossentry	95	long: updated to use \glossentry and \subglossentry	289
\@glossarysubentryfield: switched to \subglossentry	95	longragged: updated to use \glossentry and \subglossentry	300
General: added nogroupskip key to \printglossary	215	longragged3col: updated to use \glossentry and \subglossentry	301
removed definition of \@glossaryentryfield	384	tree: updated to use \glossentry and \subglossentry	326
removed definition of \@glossarysubentryfield	384	\setglossarystyle: new	224
\compatibleglossentry: new	219	\setglossentrycompatibility: new	221
\compatiblesubglossentry: new ...	220	superragged: updated to use \glossentry and \subglossentry	318
\glossaryentryfield: deprecated ...	221	3.09a (2013-10-09)	
\Glossentrydesc: new	220	\@gls@assign@symbolplural@field: new	23
\glossentrydesc: new	220	\@gls@default@value: new	71
\Glossentryname: new	220	\Glsentrydesc: made robust	159
\glossentryname: new	219	\Glsentrydescplural: made robust ..	159
\Glossentrysymbol: new	220	\Glsentryfirst: made robust	161
\glossentrysymbol: new	220	\Glsentryfirstplural: made robust .	161
\gls@assign@desc@field: new	22	\Glsentryfull: made robust	164
\gls@assign@descplural@field: new	22	\Glsentryfullpl: made robust	164
\gls@assign@field: new	77	\Glsentrylong: made robust	164
\gls@ifnotmeasuring: new	97	\Glsentrylongpl: made robust	164
		\Glsentryname: made robust	158
		\Glsentryplural: made robust	160
		\Glsentryshort: made robust	163
		\Glsentryshortpl: made robust	163
		\Glsentrysymbol: made robust	160
		\Glsentrysymbolplural: made robust	160
		\Glsentrytext: made robust	160

\Glsentryuseri: made robust	162	\@GLSpl: add \glslabel,	
\Glsentryuserii: made robust	162	\glsifplural, \glscapscase,	
\Glsentryuseriii: made robust	162	\glscustomtext and \glsinsert	135
\Glsentryuseriv: made robust	163	change to using \glsentryfmt style	
\Glsentryuserv: made robust	163	commands	136
\Glsentryuservi: made robust	163	removed \MakeUppercase as now	
\glstextup: new	229	dealt with in \glsentryfmt	136
\ifglshassymbol: changed test to check		\@Gls@: add \glsifplural,	
for \@gls@default@symbol	63	\glscapscase, \glscustomtext and	
3.10a (2013-09-28)		\glsinsert	132
\gls@assign@type@field: new	22	change to using \glsentryfmt style	
3.10a (2013-10-13)		commands	132
\@gls@keymap: new	80	removed \makefirstuc (now dealt	
\@gls@provide@newglossary: new . . .	66	with in \glsentryfmt)	132
\@gls@writedef: new	80	\@Glspl@: add \glsifplural,	
\@glsdefaultplural: Obsolete	75	\glscapscase, \glscustomtext and	
\@glsnodelsc: new	75	\glsinsert	135
\@print@glossary: Added		change to using \glsentryfmt style	
providecommand code to aux file . .	204	commands	135
\gls@defglossaryentry: Changed to		removed \makefirstuc (now dealt	
using \@gls@default@value	89	with in \glsentryfmt)	135
new	88	\@acrlong: added \glslabel,	
\glswritedefhook: new	88	\glsifplural, \glscapscase,	
\makeglossaries: Added		\glsinsert and \glscustomtext	377
providecommand code to aux file . .	182	\@acrshort: added \glslabel,	
\new@glossaryentry: new	78	\glsifplural, \glscapscase,	
\ns@newglossary: added		\glsinsert and \glscustomtext	376
\@gls@provide@newglossary	67	\@gls@: add \glslabel, \glsifplural,	
3.11a (2013-10-15)		\glscapscase, \glscustomtext and	
\@ACRlong: added \glslabel,		\glsinsert	131
\glsifplural, \glscapscase,		change to using \glsentryfmt style	
\glsinsert and \glscustomtext	378	commands	132
\@ACRshort: added \glslabel,		\@gls@noexpand@fields: Fixed bug	
\glsifplural, \glscapscase,		expand replaced with noexpand	76
\glsinsert and \glscustomtext	377	\@glsdisp: add \glslabel,	
\@Acrlong: added \glslabel,		\glsifplural, \glscapscase,	
\glsifplural, \glscapscase,		\glscustomtext and \glsinsert	136
\glsinsert and \glscustomtext	378	change to using \glsentryfmt style	
\@Acrshort: added \glslabel,		commands	136
\glsifplural, \glscapscase,		\@glspl@: add \glslabel,	
\glsinsert and \glscustomtext	376	\glsifplural, \glscapscase,	
\@GLS@: add \glslabel, \glsifplural,		\glscustomtext and \glsinsert	134
\glscapscase, \glscustomtext and		change to using \glsentryfmt style	
\glsinsert	133	commands	134
change to using \glsentryfmt style		General: added \glslabel,	
commands	133	\glsifplural, \glscapscase,	
removed \MakeUppercase (now		\glsinsert and	
moved to \glsentryfmt)	133	\glscustomtext	150–156

changed to just use		<code>\defglentryfmt:new</code>	66
<code>\Glsentrydescplural</code>	142	<code>\forglentries:replaced \ifx with</code>	
changed to just use		<code>\ifdefempty</code>	59
<code>\glentrydescplural</code>	142, 143	<code>\gls@assign@desc:new</code>	88
changed to just use <code>\Glsentrydesc</code> .	142	<code>\gls@defglossaryentry: Fixed default</code>	
changed to just use		counter if none supplied	92
<code>\glentrydesc</code>	141, 142	<code>\gls@doentryfmt:new</code>	66
changed to just use		<code>\glsdisplay:obsoleted</code>	114
<code>\Glsentryfirstplural</code>	140	<code>\glsdisplayfirst:obsoleted</code>	114
changed to just use		<code>\glsgenentryfmt:new</code>	109
<code>\glentryfirstplural</code>	140	<code>\glsgetgroupitle: Added check in</code>	
changed to just use <code>\Glsentryfirst</code>	138	case non-Latin alphabet in use	222
changed to just use		<code>\glsglossarymark:replaced</code>	
<code>\glentryfirst</code>	138, 139	<code>\MakeUppercase with</code>	
changed to just use <code>\Glsentryname</code> .	141	<code>\mfirstucMakeUppercase</code>	47
changed to just use <code>\glentryname</code> .	141	<code>\glsnavigation: switched to using</code>	
changed to just use <code>\Glsentryplural</code>	139	<code>\@gls@getgroupitle</code>	282
changed to just use <code>\glentryplural</code>	139	<code>\ifglshasdesc:replaced \ifdefempty</code>	
changed to just use		with <code>\ifcempty</code>	63
<code>\Glsentrysymbolplural</code>	144	<code>\ifglshaslong:new</code>	63
changed to just use		<code>\ifglshasshort:new</code>	63
<code>\glentrysymbolplural</code>	144	<code>\ifglshassymbol:replaced</code>	
changed to just use <code>\Glsentrysymbol</code>	143	<code>\ifdefempty with \ifcempty</code>	63
changed to just use		<code>\ifglused:replaced \ifthenelse with</code>	
<code>\glentrysymbol</code>	143, 144	<code>\ifbool</code>	60
Changed to just use <code>\Glsentrytext</code> .	138	<code>\longnewglossaryentry:new</code>	88
changed to just use <code>\glentrytext</code> .	137	<code>\ns@newglossary:replaced</code>	
changed to just use		<code>\glsdisplay and</code>	
<code>\Glsentryuseriii</code>	147	<code>\glsdisplayfirst with</code>	
changed to just use		<code>\glentryfmt</code>	67
<code>\glentryuseriii</code>	146, 147	compatible-3.07: <code>cnew</code>	34
changed to just use <code>\Glsentryuserii</code>	146	<code>\SetCustomDisplayStyle: updated to</code>	
changed to just use		use <code>\defglentryfmt</code>	264
<code>\glentryuserii</code>	145, 146	<code>\SetDefaultAcronymDisplayStyle:</code>	
changed to just use <code>\Glsentryuseriv</code>	147	changed to use <code>\defglentryfmt</code> .	247
changed to just use		<code>\SetDescriptionAcronymDisplayStyle:</code>	
<code>\glentryuseriv</code>	147, 148	updated to use <code>\defglentryfmt</code> .	254
changed to just use <code>\Glsentryuseri</code>	145	<code>\SetDescriptionDUAAcronymDisplayStyle:</code>	
changed to just use <code>\glentryuseri</code>	145	updated to use <code>\defglentryfmt</code> .	252
changed to just use <code>\Glsentryuservi</code>	149	<code>\SetDescriptionFootnoteAcronymDisplayStyle:</code>	
changed to just use <code>\glentryuservi</code>	149	updated to use <code>\defglentryfmt</code> .	250
changed to just use <code>\Glsentryuserv</code>	148	<code>\SetDUADisplayStyle: updated to use</code>	
changed to just use <code>\glentryuserv</code>	148	<code>\defglentryfmt</code>	262
Now requires textcase	4	<code>\SetFootnoteAcronymDisplayStyle:</code>	
acronymlists:replaced		updated to use <code>\defglentryfmt</code> .	256
<code>\@addtoacronymlists with</code>		<code>\SetSmallAcronymDisplayStyle:</code>	
<code>\DeclareAcronymList</code>	20	updated to use <code>\defglentryfmt</code> .	259
<code>\defgldisplay:obsoleted</code>	115	<code>\setupglossaries:new</code>	38
<code>\defgldisplayfirst:obsoleted</code>	115	<code>\showglolong:new</code>	270

<code>\showgloshort:new</code>	269	<code>\glsprestandardsort:new</code>	14
<code>numbers:new</code>	36	<code>\glsssetexpandfield:new</code>	22
<code>symbols:new</code>	36	<code>\glsssetnoexpandfield:new</code>	22
3.12a (2013-10-16)		<code>altsuper4colheader:switched to</code>	
<code>\gls@defglossaryentry:added</code>		<code>\tabularnewline</code>	316
<code>\glslabel</code>	89	<code>altsuper4colheaderborder:switched</code>	
<code>\glsaddkey:new</code>	82	<code>to \tabularnewline</code>	317
3.13a (2013-11-05)		<code>long:switched to \tabularnewline</code> ..	289
<code>\@gls@assign@symbol@field:changed</code>		<code>long3col:switched to</code>	
<code>to use \glsssetnoexpandfield</code>	23	<code>\tabularnewline</code>	290
<code>\@gls@assign@symbolplural@field:</code>		<code>long3colheader:switched to</code>	
<code>changed to use</code>		<code>\tabularnewline</code>	291
<code>\glsssetnoexpandfield</code>	23	<code>long3colheaderborder:switched to</code>	
<code>\@gls@link:removed \relax</code>	120	<code>\tabularnewline</code>	291
<code>\@gls@notranslatorhook:new</code>	27	<code>long4col:switched to</code>	
<code>\@gls@setupsort@standard:moved</code>		<code>\tabularnewline</code>	292
<code>\@gls@santizesort to</code>		<code>long4colheader:switched to</code>	
<code>\glsprestandardsort</code>	14	<code>\tabularnewline</code>	292
<code>ucmark:added check for memoir</code>	12	<code>longheader:switched to</code>	
<code>see:added \gls@checkseeallowed</code> ...	72	<code>\tabularnewline</code>	290
<code>\glossarysection:changed</code>		<code>longheaderborder:switched to</code>	
<code>\glossarymark to</code>		<code>\tabularnewline</code>	290
<code>\gls glossarymark</code>	47	<code>\SetFootnoteAcronymDisplayStyle:</code>	
<code>\glossarystyle:fixed bug caused by</code>		<code>fixed missing argument bug</code>	257
<code>using \ifdef instead of \ifcsdef</code> .	224	<code>super:switched to \tabularnewline</code> .	311
<code>\gls@assign@desc@field:changed to</code>		<code>super3col:switched to</code>	
<code>use \glsssetnoexpandfield</code>	22	<code>\tabularnewline</code>	313
<code>\gls@assign@descplural@field:</code>		<code>super3colheader:switched to</code>	
<code>changed to use</code>		<code>\tabularnewline</code>	313
<code>\glsssetnoexpandfield</code>	22	<code>super4col:switched to</code>	
<code>\gls@assign@name@field:changed to</code>		<code>\tabularnewline</code>	314
<code>use \glsssetnoexpandfield</code>	23	<code>super4colheader:switched to</code>	
<code>\gls@assign@type@field:changed to</code>		<code>\tabularnewline</code>	315
<code>use \glsssetexpandfield</code>	22	<code>super4colheaderborder:switched to</code>	
<code>\gls@checkseeallowed:new</code>	72	<code>\tabularnewline</code>	315
<code>\glsaddallunused:set default to</code>		<code>superheader:switched to</code>	
<code>\@glo@types</code>	167	<code>\tabularnewline</code>	312
<code>\Glsentryfull:changed to use</code>		<code>superheaderborder:switched to</code>	
<code>\acrfullformat</code>	164	<code>\tabularnewline</code>	312
<code>\Glsentryfull:changed to use</code>		3.14a (2013-11-12)	
<code>\acrfullformat</code>	164	<code>\@glswritefiles:renamed</code>	
<code>\Glsentryfullpl:changed to use</code>		<code>\glswritefiles to</code>	
<code>\acrfullformat</code>	164	<code>\@glswritefiles and used</code>	
<code>\Glsentryfullpl:changed to use</code>		<code>“savewrites” option to set</code>	
<code>\acrfullformat</code>	164	<code>\glswritefiles</code>	189
<code>\gls glossarymark:renamed</code>		<code>General:new</code>	273
<code>\glossarymark to</code>		<code>acronyms:new</code>	18
<code>\gls glossarymark to avoid conflict</code>		<code>\gls@defglossaryentry:added check</code>	
<code>with memoir</code>	47	<code>for existence of default glossary</code>	90

set the default for firstplural to be the value of plural	92	\GlsUseAcrEntryDispStyle: new ...	235
xindygloss: new	31	\GlsUseAcrStyleDefs: new	235
\longprovideglossaryentry: new ...	88	short-long: new	237
compatible-2.07: added check for 2.07 before setting 3.07 compatibility	34	short-long-desc: new	240
notranslate: new	27	xindynoglsnumbers: new	31
\provideglossaryentry: new	78	sm-short-long: new	238
4.0 (2013-11-14)		sm-short-long-desc: new	240
\gls@defglossaryentry: added check for first key	92	index: new	37
super: fixed typo in \subglossentry (\glossentrydesc)	311	\newacronymstyle: new	235
4.01 (2013-11-16)		long-sc-short: new	238
General: fixed non-value options so that they can be passed to document class .	9	long-sc-short-desc: new	239
\CustomAcronymFields: inserted missing comma	265	long-short: new	236
4.02 (2013-12-05)		long-short-desc: new	239
\@acrfull: now using \acrfullfmt ..	230	long-sm-short: new	238
\@gls@indexdef: new	37	long-sm-short-desc: new	239
\@gls@numbersdef: new	37	long-sp-short-desc: new	239
\@gls@symbolsdef: new	36	footnote: new	243
General: Removed \acronymfont .	153–157	footnote-desc: new	245
\ACRfullfmt: new	231	footnote-sc: new	244
\Acrfullfmt: new	230	footnote-sc-desc: new	245
\acrfullfmt: new	230	footnote-sm: new	244
\ACRfullplfmt: new	232	footnote-sm-desc: new	245
\Acrfullplfmt: new	232	\setacronymstyle: new	234
\acrfullplfmt: new	231	\SetDescriptionAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	254
\acronymentry: new	234	\SetDescriptionFootnoteAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	250
sanitize: fixed bug that caused an error here	26	\SetFootnoteAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	256
sc-short-long: new	238	\SetGenericNewAcronym: new	233
sc-short-long-desc: new	240	\SetSmallAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	259
\Genacrfullformat: new	114	dua: new	241
\genacrfullformat: new	114	dua-desc: new	242
\GenericAcronymFields: new	234	numberedsection: added nameref option	9
\Genplacrfullformat: new	114	4.02 (2013-13-05)	
\Glsentryfull: bug fix: added missing \acronymfont	164	\makeglossaries: made preamble only	184
\glsentryfull: bug fix: added missing \acronymfont	164	4.03 (2014-01-17)	
\Glsentryfullpl: bug fix: added missing \acronymfont	164	General: changed default to \@empty instead of \relax	34
\glsentryfullpl: bug fix: added missing \acronymfont	164		
\glsgenacfmt: new	112		

4.03 (2014-01-20)

\@do@esc@wrglossary: added		\compatiblesubglossentry: added	
\glsdetoklabel	196	\glsdetoklabel	354
\@do@noesc@wrglossary: added		\Genacrfullformat: redefined to use	
\glsdetoklabel	193	accessibility information	375
\@ACRlong: removed \glslabel		\genacrfullformat: redefined to use	
(defined in \@gls@link)	378	accessibility information	375
\@ACRshort: removed \glslabel		\Genplacrfullformat: redefined to use	
(defined in \@gls@link)	377	accessibility information	376
\@Acrlong: removed \glslabel		\genplacrfullformat: redefined to use	
(defined in \@gls@link)	378	accessibility information	376
\@Acrshort: removed \glslabel		\glossentryname: added	
(defined in \@gls@link)	376	\glsdetoklabel	219
\@GLS@: removed \glslabel (defined in		\gls@defglossaryentry: added	
\@gls@link)	133	\glsdetoklabel	88
\@GLSpl: removed \glslabel (defined		replaced #1 with \@glo@label	90
in \@gls@link)	135	replaced \ifthenelse with	
\@Gls@: removed \glslabel (defined in		\ifdefequal	91
\@gls@link)	132	\glsadd: added \glsdetoklabel	166
\@Gls@entry@field: new	157	\glsaddkey: switched to using	
\@Glspl@: removed \glslabel (defined		\@gls@field@link	83
in \@gls@link)	135	\glsdetoklabel: new	60
\@acrlong: removed \glslabel		\glsdisplaynumberlist: added	
(defined in \@gls@link)	377	\glsdetoklabel	165
\@acrshort: removed \glslabel		\glsdoifexistsorwarn: new	61
(defined in \@gls@link)	376	\glsentryaccess: switched to using	
\@gls@: removed \glslabel (defined in		\@gls@entry@field	359
\@gls@link)	131	\glsentrydescaccess: switched to	
\@gls@access@display: new	363	using \@gls@entry@field	360
\@gls@entry@field: new	157	\glsentrydescpluralaccess: switched	
\@gls@fetchfield: new	81	to using \@gls@entry@field	360
\@gls@field@link: new	137	\glsentryfirstaccess: switched to	
\@gls@link: added \glsdetoklabel .	119	using \@gls@entry@field	360
moved \@gls@link@opts and		\glsentryfirstplural: added	
\@gls@link@label to \@gls@link	119	\glsdetoklabel	161
\@gls@writedef: added		\glsentrylongaccess: switched to	
\glsdetoklabel	80	using \@gls@entry@field	361
\@glsdisp: removed \glslabel		\glsentrylongpluralaccess: switched	
(defined in \@gls@link)	136	to using \@gls@entry@field	361
\@glspl@: removed \glslabel (defined		\glsentrypluralaccess: switched to	
in \@gls@link)	134	using \@gls@entry@field	360
\@printglossary: added		\glsentryshortaccess: switched to	
\glsdetoklabel	203	using \@gls@entry@field	360
General: removed \glslabel (defined in		\glsentryshortpluralaccess:	
\@gls@link)	150	switched to using	
sc-short-long-desc: redefined to use		\@gls@entry@field	360
accessibility information	388	\glsentrysymbolaccess: switched to	
\compatibleglossentry: added		using \@gls@entry@field	360
\glsdetoklabel	353	\glsentrysymbolpluralaccess:	
		switched to using	

\showglolong: added \glsdetoklabel	270	redefined to use accessibility	
\showglolongaccess: added		information	388
\glsdetoklabel	401	dua-desc: commented spurious EOL ..	243
\showglolongpluralaccess: added		redefined to use accessibility	
\glsdetoklabel	401	information	391
\showglongname: added \glsdetoklabel	268	4.04 (2014-03-04)	
\showglongnameaccess: added		\@gls@getcounterprefix: added	
\glsdetoklabel	399	warning if no prefix can be formed ..	197
\showgloparent: added		4.04 (2014-03-06)	
\glsdetoklabel	266	\@gls@noidx@nosanitizesort: new ..	24
\showgloplural: added		\@gls@noidx@sanitizesort: new ...	23
\glsdetoklabel	267	\@gls@nosanitizesort: new	23
\showglopluralaccess: added		\@gls@sanitizesort: new	23
\glsdetoklabel	400	\@glo@addchildren: new	205
\showgloshort: added		\@glo@do@sortentries: new	206
\glsdetoklabel	269	\@glo@grabfirst: new	211
\showgloshortaccess: added		\@glo@sortedinsert: new	206
\glsdetoklabel	400	\@glo@sortentries: new	205
\showgloshortpluralaccess: added		\@glo@sorthandler@case: new	207
\glsdetoklabel	401	\@glo@sorthandler@letter: new ...	207
\showglosort: added \glsdetoklabel	269	\@glo@sorthandler@nocase: new ...	207
\showglosymbol: added		\@glo@sorthandler@word: new	206
\glsdetoklabel	269	\@glo@sortmacro@case: new	208
\showglosymbolaccess: added		\@glo@sortmacro@def: new	209
\glsdetoklabel	400	\@glo@sortmacro@def@do: new	209
\showglosymbolplural: added		\@glo@sortmacro@letter: new	208
\glsdetoklabel	269	\@glo@sortmacro@nocase: new	209
\showglosymbolpluralaccess: added		\@glo@sortmacro@standard: new ...	208
\glsdetoklabel	400	\@glo@sortmacro@use: new	210
\showglotext: added \glsdetoklabel	266	\@glo@sortmacro@word: new	207
\showglotextaccess: added		\@gls@noidx@do: new	211
\glsdetoklabel	400	\@gls@noidx@getgrouptitle: new ..	223
\showglotype: added \glsdetoklabel	267	\@gls@noref@warn: new	188
\showglouseri: added		\@gls@reference: new	213
\glsdetoklabel	267	\@gls@warnonglossdefined: new	21
\showglouserii: added		\@gls@warnontheGLOSSdefined: new ..	22
\glsdetoklabel	268	\@no@makeglossaries: new	188
\showglouseriii: added		\@print@glossary: new	203
\glsdetoklabel	268	\@print@noidx@glossary: new	210
\showglouseriv: added		\@print@gloss@setsort: new	201
\glsdetoklabel	268	\@print@glossary: new	201
\showglouseriv: added		General: added sort key to printgloss	
\glsdetoklabel	268	group	216
\showglouseriv: added		\compatibleglossentry: changed	
\glsdetoklabel	268	\newcommand to \def as is may or	
dua: fixed bug in \acrfullfmt	242	may not be defined	353
fixed bug in \Acrfullplfmt	242	\compatiblesubglossentry: changed	
fixed bug in \acrfullplfmt	242	\newcommand to \def as is may or	
		may not be defined	354

\defglsdisplayfirst: fixed unwanted space	115	\Acrfullplfmt: fixed no case change bug	232
\glo@grabfirst: new	210	\glsletentryfield: new	157
\gls@defglossaryentry: replaced \ifx with \ifdefvoid	94	4.08 (2014-07-30)	
\glsnoidxdisplayloc: new	213	\@ACRlong: added	
\glsnoidxdisplaylocclishandler: new	213	\do@gls@link@checkfirsthyper	378
\glsnoidxloclist: new	212	\@ACRshort: added	
\glsnoidxlocclishandler: new	213	\do@gls@link@checkfirsthyper	377
\glsnoidxstripaccents: new	24	\@Acrlong: added	
alltree: moved hangindent and parindent assignments outside level test	330	\do@gls@link@checkfirsthyper	377
\makeglossaries: Moved definition of \glswrite to \makeglossaries ..	182	\@Acrshort: added	
\makenoidxglossaries: new	185	\do@gls@link@checkfirsthyper	376
\printglossary: changed to use new \@printglossary	200	\@GLS@: moved \glsifhyper	133
\printnoidxglossaries: new	201	moved check for first use to \@gls@link	133
\printnoidxglossary: new	201	\@GLSpl: moved \glsifhyper	136
\showgloclolist: new	270	moved check for first use to \@gls@link	136
\warn@noprintglossary: Activate warning in \makeglossaries	200	\@Gls@: moved \glsifhyper	132
\writeist: checked for definition of \glswrite	169, 174	moved check for first use to \@gls@link	132
4.06 (2014-03-12)		\@Glspl@: moved \glsifhyper	135
\@GLS@: added \glsifhyper	133	moved check for first use to \@gls@link	135
\@GLSpl: added \glsifhyper	136	\@acrlong: added	
\@Gls@: added \glsifhyper	132	\do@gls@link@checkfirsthyper	377
\@Glspl@: added \glsifhyper	135	\@acrshort: added	
\@gls@: added \glsifhyper	132	\do@gls@link@checkfirsthyper	376
\@gls@numbersdef: added hook to set toc title	37	\@closegls: new	179
\@gls@symbolsdef: added hook to set toc title	36	\@gls@: moved \glsifhyper	132
\@glsdisp: added \glsifhyper	136	moved check for first use to \@gls@link	132
\@glspl@: added \glsifhyper	134	\@gls@automake: new	179
General: added \glsifhyper	150–157	\@gls@doautomake: new	34
acronym: added hook to set toc title	18	\@gls@field@link: added assignment of \do@gls@link@checkfirsthyper	137
acronyms: added hook to set toc title ...	18	\@gls@forbidtexext: new	66
\glsdefmain: added hook to set toc title	17	\@gls@hyp@opt: new	117
4.07 (2014-04-04)		\@gls@link: removed redundancy	119
\@glossarysection: added optional argument when using unstarred version	48	renamed \gls@type to \glstype ...	119
\@gls@noidx@do: added \global in case it's used in a tabular-like style	211	\@gls@link@checkfirsthyper: new .	118
		\@glsdisp: moved \glsifhyper	136
		moved check for first use to \@gls@link	136
		\@glspl@: moved \glsifhyper	134
		moved check for first use to \@gls@link	134
		\@ignored@glossaries: new	69

General: added entrycounter option to		removed \@sGLStext	137
printgloss family	215	removed \@sGlstext	138
added nopostdot option to		removed \@sglstext	137
printgloss family	215	removed \@sGLSuseriii	147
added subentrycounter option to		removed \@sGlsuseriii	146
printgloss family	215	removed \@sglsuseriii	146
explicitly initialise hyper key	116	removed \@sGLSuserii	146
moved \glsifhyper	150–157	removed \@sGlsuserii	146
removed \@sACRlongpl	156	removed \@sglsuserii	145
removed \@sAcrlongpl	156	removed \@sGLSuseriv	148
removed \@sacrlongpl	155	removed \@sGlsuseriv	147
removed \@sACRlong	154	removed \@sglsuseriv	147
removed \@sAcrlong	154	removed \@sGLSuseri	145
removed \@sacrlong	153	removed \@sGlsuseri	145
removed \@sACRshortpl	152	removed \@sglsuseri	145
removed \@sAcrshortpl	152	removed \@sGLSsuservi	149
removed \@sacrshortpl	151	removed \@sGlsuservi	149
removed \@sACRshort	151	removed \@sglsuservi	149
removed \@sAcrshort	150	removed \@sGLSsuserv	148
removed \@sacrshort	149	removed \@sGlsuserv	148
removed \@sgls@link	118	removed \@sglsuserv	148
removed \@sGLSdescplural	143	removed \@sGLS	133
removed \@sGlsdescplural	142	removed \@sGls	132
removed \@sglsdescplural	142	removed \@sgls	131
removed \@sGLSdesc	142	removed \@thirdofthree (defined in	
removed \@sGlsdesc	142	kernel)	131
removed \@sglsdesc	141	removed sPGLS	278
removed \@sglsdisp	136	removed sPgls	277
removed \@sGLSfirstplural	140	removed spgls	275
removed \@sGlsfirstplural	140	removed sPGLSpl	279
removed \@sglsfirstplural	140	removed sPglspl	277
removed \@sGLSfirst	139	removed spglspl	276
removed \@sGlsfirst	138	\ACRfull: removed \@sACRfull	231
removed \@sglsfirst	138	switched to using \@gls@hyp@opt ..	231
removed \@sGLSname	141	\Acrfull: removed \@sAcrfull	230
removed \@sGlsname	141	switched to using \@gls@hyp@opt ..	230
removed \@sglsname	140	\acrfull: removed \@sacrfull	229
removed \@sGLSplural	139	switched to using \@gls@hyp@opt ..	229
removed \@sGlsplural	139	\ACRfullpl: removed \@sACRfullpl ..	232
removed \@sglsplural	139	switched to using \@gls@hyp@opt ..	232
removed \@sGLSpl	135	\Acrfullpl: removed \@sAcrfullpl ..	231
removed \@sGlspl	134	switched to using \@gls@hyp@opt ..	231
removed \@sglspl	134	\acrfullpl: removed \@sacrfullpl ..	231
removed \@sGLSsymbolplural	144	switched to using \@gls@hyp@opt ..	231
removed \@sGlsymbolplural	144	\ACRlong: switched to using	
removed \@sglsymbolplural	144	\@gls@hyp@opt	154
removed \@sGLSsymbol	143	\Acrlong: switched to using	
removed \@sGlsymbol	143	\@gls@hyp@opt	153
removed \@sglsymbol	143		

\acrlong: switched to using \@gls@hyp@opt	153	\glsdohyperlink: new	130
\ACRlongpl: switched to using \@gls@hyp@opt	156	\glsdohypertarget: new	130
\Acrlongpl: switched to using \@gls@hyp@opt	155	\glsenablehyper: added \KV@glslink@hypertrue to definition	131
\acrlongpl: switched to using \@gls@hyp@opt	155	\GLSfirst: switched to using \@gls@hyp@opt	138
\ACRshort: switched to using \@gls@hyp@opt	150	\Glsfirst: switched to using \@gls@hyp@opt	138
\Acrshort: switched to using \@gls@hyp@opt	150	\glsfirst: switched to using \@gls@hyp@opt	138
\acrshort: switched to using \@gls@hyp@opt	149	\GLSfirstplural: switched to using \@gls@hyp@opt	140
\ACRshortpl: switched to using \@gls@hyp@opt	152	\Glsfirstplural: switched to using \@gls@hyp@opt	140
\Acrshortpl: switched to using \@gls@hyp@opt	152	\glsfirstplural: switched to using \@gls@hyp@opt	140
\acrshortpl: switched to using \@gls@hyp@opt	151	\glsifhyper: deprecated	117
\forallacronyms: new	59	\glslink: switched to using \@gls@hyp@opt	118
\GLS: switched to using \@gls@hyp@opt	133	\glslinkcheckfirsthyperhook: new	119
\Gls: switched to using \@gls@hyp@opt	132	\glslinkvar: new	117
\gls: switched to using \@gls@hyp@opt	131	\GLSname: switched to using \@gls@hyp@opt	141
\glsdefglossaryentry: added check for ignored glossary	90	\Glsname: switched to using \@gls@hyp@opt	141
\gls@istfilebase: new	44	\glsname: switched to using \@gls@hyp@opt	140
\glsaddkey: removed \@sGLS@user@<key>	84	\GLSpl: switched to using \@gls@hyp@opt	135
removed \@sGls@user@<key>	84	\Glspl: switched to using \@gls@hyp@opt	134
removed \@sgls@user@<key>	83	\glspl: switched to using \@gls@hyp@opt	134
switched to using \@gls@hyp@opt	83, 84	\GLSplural: switched to using \@gls@hyp@opt	139
\GLSdesc: switched to using \@gls@hyp@opt	142	\Glsplural: switched to using \@gls@hyp@opt	139
\Glsdesc: switched to using \@gls@hyp@opt	141	\glsplural: switched to using \@gls@hyp@opt	139
\glsdesc: switched to using \@gls@hyp@opt	141	\glsspace: new	230
\GLSdescplural: switched to using \@gls@hyp@opt	143	\GLSsymbol: switched to using \@gls@hyp@opt	143
\Glsdescplural: switched to using \@gls@hyp@opt	142	\Glsymbol: switched to using \@gls@hyp@opt	143
\glsdescplural: switched to using \@gls@hyp@opt	142	\glssymbol: switched to using \@gls@hyp@opt	143
\glsdisablehyper: added \KV@glslink@hyperfalse to definition	130	\GLSsymbolplural: switched to using \@gls@hyp@opt	144
\glsdisp: switched to using \@gls@hyp@opt	136		

\Glsymbolplural: switched to using \@gls@hyp@opt	144	altlongragged4col: fixed bug that displayed description instead of symbol	303
\glssymbolplural: switched to using \@gls@hyp@opt	144	\newglossary: added starred version ..	67
\GLStext: switched to using \@gls@hyp@opt	137	\newignoredglossary: new	69
\Glstext: switched to using \@gls@hyp@opt	138	\ns@newglossary: added \@glotype@<name>@log	67
\glstext: switched to using \@gls@hyp@opt	137	new	67
\glstreenamefmt: new	323	\p@gls@hyp@opt: new	117
\GLSuseri: switched to using \@gls@hyp@opt	145	\PGLS: changed to use \@gls@hyp@opt	278
\Glsuseri: switched to using \@gls@hyp@opt	145	\Pgls: changed to use \@gls@hyp@opt	277
\glsuseri: switched to using \@gls@hyp@opt	145	\pgls: changed to use \@gls@hyp@opt	275
\GLSuserii: switched to using \@gls@hyp@opt	146	\PGLSpl: changed to use \@gls@hyp@opt	279
\Glsuserii: switched to using \@gls@hyp@opt	146	\Pglspl: changed to use \@gls@hyp@opt	277
\glsuserii: switched to using \@gls@hyp@opt	145	\pglspl: changed to use \@gls@hyp@opt	276
\GLSuseriii: switched to using \@gls@hyp@opt	147	\s@gls@hyp@opt: new	117
\Glsuseriii: switched to using \@gls@hyp@opt	146	\s@newglossary: new	67
\glsuseriii: switched to using \@gls@hyp@opt	146	automake: new	33
\GLSuseriv: switched to using \@gls@hyp@opt	147	4.09 (2014-08-12)	
\Glsuseriv: switched to using \@gls@hyp@opt	147	\glsaddkey: fixed bug in user commands	83
\glsuseriv: switched to using \@gls@hyp@opt	147	4.10 (2014-08-27)	
\GLSuserv: switched to using \@gls@hyp@opt	148	\@Gls@acentryname: new	158
\Glsuserv: switched to using \@gls@hyp@opt	148	\@Gls@entryname: new	158
\glsuserv: switched to using \@gls@hyp@opt	148	\@gls@glossary: Renamed \@glossary to \@gls@glossary	190
\GLSuservi: switched to using \@gls@hyp@opt	149	\glspercentchar: new	168
\Glsuservi: switched to using \@gls@hyp@opt	149	\glstildechar: new	168
\glsuservi: switched to using \@gls@hyp@opt	149	alttree: moved space after symbol	330, 331
\ifignoredglossary: new	69	4.11 (2014-09-01)	
		\@do@esc@wrglossary: added hook ..	195
		sanitize: none option	26
		\gls@wrglossary: renamed from \@wrglossary to \gls@wrglossary	190
		\glsaddprotectedpagefmt: new	192
		\glsbackslash: new	168
		4.12 (2014-11-22)	
		\@gls@addpredefinedattributes: Added glsignore attribute	53
		\@gls@adjustmode: new	167
		\@gls@nottranslatorhook: removed ...	27
		\@gls@toc: added \protect to \numberline	49
		\@gls@usetranslator: new	27
		\glsacrpluralsuffix: new	41
		\glsadd: added check for vertical mode	166

\glsaddallunused: replaced @gobble with glsignore	167	\glsunset: switched to \@glsunset ...	98
\glsifusedtranslatordict: new	27	4.15 (2015-03-16)	
\glsignore: new	167	General: bug fix replaced \@glo@type with \glstype	156
\glsupacrpluralsuffix: new	41	4.16 (2015-06-18)	
\ProvidesGlossariesLang: new	41	\glsaddstoragekey: new	82
\RequireGlossariesLang: new	41	4.16 (2015-07-08)	
4.13 (2015-02-03)		\@ACRlong: added \glspostlinkhook	378
\indexspace: new	285, 305, 323	\@ACRshort: added \glspostlinkhook	377
4.14 (2015-02-28)		\@Acrlong: added \glspostlinkhook	378
\@@glslocalreset: new	99	\@Acrshort: added \glspostlinkhook	377
\@@glslocalunset: new	98	\@GLS@: added \glspostlinkhook ...	133
\@@glsreset: new	99	\@GLSpl: added \glspostlinkhook ..	136
\@@glsunset: new	98	\@Gls@: added \glspostlinkhook ...	133
\@@newglossaryentry@defcounters: new	100	\@Glspl@: added \glspostlinkhook .	135
\cGls: new	103	\@acrlong: added \glspostlinkhook	377
\cGls@: new	103	\@acrshort: added \glspostlinkhook	376
\cGlspl@: new	104	\@gls@: added \glspostlinkhook ...	132
\cgls: new	103	\@gls@@link: added \glspostlinkhook	118
\cgls@: new	103	\@gls@field@link: added \glspostlinkhook	137
\cglspl: new	104	\@gls@link: moved definition of \glsifhyperon outside of this macro	120
\cglspl@: new	104	\@glsdisp: added \glspostlinkhook	137
\@gls@entry@count: new	103	\@glspl@: added \glspostlinkhook .	134
\@gls@increment@currcount: new ..	102	General: added \glspostlinkhook	150–157
\@gls@local@increment@currcount: new	102	\glsacspace: new	237
\@gls@write@entrycounts: new	102	\glsadd: changed \@do@wrglossary to \@do@wrglossary	166
\@glslocalreset: new	99	\glsfielddef: new	86
\@glslocalunset: new	98	\glsfieldedef: new	85
\@glsreset: new	99	\glsfieldfetch: new	86
\@glsunset: new	98	\glsfieldgdef: new	85
\@newglossaryentry@defcounters: new	94	\glsfieldxdef: new	85
\cGls: new	103	\glsifhyperon: moved definition of \glsifhyperon	119
\cgls: new	103	\glslinkpostsetkeys: new	119
\cGlsformat: new	104	\glspostlinkhook: new	118
\cglsformat: new	103	\glswriteentry: new	191
\cGlspl: new	104	\ifglsfieldcseq: new	87
\cglspl: new	104	\ifglsfielddefeq: new	87
\cGlsplformat: new	105	\ifglsfieldeq: new	87
\cglsplformat: new	104	long-sp-short: new	236
\gls@defdocnewglossaryentry: new .	77	\showglofield: new	270
\glsenableentrycount: new	100	4.18 (2015-09-09)	
\glslocalreset: switched to \@glslocalreset	97	General: split mfirstuc into separate bundle	4
\glslocalunset: switched to \@glslocalunset	98		
\glsreset: switched to \@glsreset ...	97		

4.19 (2015-10-31)		\gls@arabicpage: new	192
\glstreenamebox: new	329	\gls@protected@pagefmts: added	
4.19 (2015-11-22)		arabic to list	191
\@gls@link@nocheckfirsthyper: new	137	\glstentrytitlecase: new	161
\@gls@preglossaryhook: new	201	\glsfindwidesttoplevelname: new	329
\@printglossary: added		\glslistgroupheaderfmt: new	285
\@gls@preglossaryhook	203	\glslistnavigationitem: new	285
\do@glsglisablehyperinlist: new	119	\glstreegroupheaderfmt: new	323
\doifglossarynoexistsordo: new	62	\glstreenavigationfmt: new	323
\gls@gobbleopt: new	66	\ifglswrallowprimitivemods: new	193
\glsoifexistsordo: new	61	list: fixed missing space before	
4.20 (2015-11-30)		description	285
\@gls@link: added		long: fixed typo in \glossentrydesc	289
\@gls@setdefault@glslink@opts	119	super4col: fixed bug in \glossentry	314
added \glsohnohyperlink when		4.23 (2016-04-30)	
hyperlink is suppressed	120	\glscurrentfieldvalue: new	65
\@gls@setdefault@glslink@opts:		\ifglshasfield: added	
new	119	\glscurrentfieldvalue	64, 65
\gls@checkseeallowed@preambleonly:		altlongragged4col: check for	
new	72	nogroupskip changed	303
\glsohnohyperlink: new	130	altsuperragged4col: check for	
4.21 (2016-01-24)		nogroupskip changed	322
\@printglossary: warn if no style has		long: check for nogroupskip changed	289
been set	201	long-booktabs: check for nogroupskip	
General: changed checkfirsthyper		changed	295
assignment	150–156	long3col: check for nogroupskip	
\glossarystyle: set default style if not		changed	291
already set	224	long3col-booktabs: check for	
\glsltpenaltycheck: new	298	nogroupskip changed	296
\glspatchLToutput: new	298	long4col: check for nogroupskip	
\glspenaltygroupskip: new	298	changed	292
altlong4col-booktabs: new	296	long4col-booktabs: check for	
altlongragged4col-booktabs: new	297	nogroupskip changed	296
long-booktabs: new	295	longragged: check for nogroupskip	
long3col-booktabs: new	295	changed	300
long4col-booktabs: new	296	longragged3col: check for nogroupskip	
longragged-booktabs: new	297	changed	301
longragged3col-booktabs: new	297	super: check for nogroupskip changed	311
\setglossarystyle: set default style if		super3col: check for nogroupskip	
not already set	224	changed	313
4.22 (2016-04-19)		super4col: check for nogroupskip	
\@do@esc@wrglossary: added check		changed	315
for \@arabic	195	superragged: check for nogroupskip	
added test to allow temporary primitive		changed	318
modifications and added arabic case	195	superragged3col: check for	
mcolalttreespannav: new	310	nogroupskip changed	320
mcolindexspannav: new	306	4.24 (2016-05-27)	
mcoltreenonamespannav: new	308	\@gls@extramakeindexopts: new	178
mcoltreespannav: new	307		

\@gls@glossary: added check for debug mode	190	\@gls@setupsort@none: new	16
\@gls@see@noindex: new	7	\@xdycrossrefhook: new	55
debug: new	5	\@xdylocationclassorder: bug fix: changed \edef to \def	56
seenoinindex: new	8	\glosortentrieswarning: new	21
\glsnomakeindexwarning: new	50	\gls@set@xr@key: new	72
\GlsSetQuote: new	175	\gls@xr@key: new	72
\GlsSetWriteIstHook: new	175	\GlsAddXdyLocation: bug fix: changed #1 to #2	55
4.25 (2016-06-09)		\glsnoidxstripaccents: added \a ...	24
\@gls@enablesavenonumberlist: new	73	added \TH, \dh and \DH	25
\@gls@initnonumberlist: new	73	4.31 (2017-08-10)	
\@gls@savenonumberlist: new	73	nolist: added check for “list” style	11
4.25 (??)		4.31 (2017-09-10)	
General: changed		style: changed \renewcommand to \def .	9
\DeclareRobustCommand to		4.32 (2017-08-24)	
\newrobustcmd and changed		\@glsnavhypertarget: new	281
\@ifundefined to \ifcsundef ...	366	\@glsshowtarget: new	7
4.26 (2016-10-12)		\glsshowtarget: new	6
\@glossary@default@style: added		4.33 (2017-09-20)	
check for classicthesis	9	\@do@esc@wrglossary: added	
mcolindex: replaced \@idxitem with		\gls@the and \gls@number	195
\glstreeitem	305	renamed from	
mcolindexspannav: replaced \@idxitem		\@do@esc@wrglossary	193
with \glstreeitem	306	\@do@noesc@wrglossary: new	193
\glstreechildpredesc: new	324	\@do@wrglossary: changed to check	
\glstreeitem: new	323	for esclocations	193
\glstreepredesc: new	324	\@gls@missinglang@warn: new	21
\glstreesubitem: new	324	\GlsSetXdyFirstLetterAfterDigits:	
\glstreesubsubitem: new	324	added starred version	168
4.28 (2017-01-07)		\GlsSetXdyNumberGroupOrder: new .	169
\glspatchtabularx: new	97	esclocations: new	10
4.29 (2017-01-19)		4.34 (2017-11-03)	
\@gls@noidx@do: current letter group		mcolalttreespannav: removed spurious	
assignment made global	212	space	310
\@print@noidx@glossary: moved		\glsshowtarget: modified to check for	
definition of		math mode and inner	6
\@gls@currentlettergroup outside		4.35 (2017-11-14)	
of the glossary environment	210	\glsadd: added \@gls@setsort (in case	
General: added check for		of sort=use)	166
\@glsxtr@doaccsupp	353	4.36 (2018-03-07)	
\glsnavhyperlinkname: new	280	\@gls@glossary: removed \index ...	190
4.30 (2017-06-11)		4.37 (2018-04-07)	
\@glo@autosee: new	94	\gls@begindocdefs: new	79
\@glo@autoseehook: new	94	4.38 (2018-05-10)	
\@glo@check@sortallowed: new	14	\@gls@define@glossaryentrycounter:	
\@gls@noidx@do: letter group		added check for existence of	
assignment made global	212	glossaryentry counter	12
\@gls@setupsort@def: added check for		new	12
register	15		

\@gls@define@glossarysubentrycounter: new 13 prepended \currentglossary. to \theHglossarysubentry and removed spurious eol space 13	nonumberlist: changed \val and \nr to \gls@nonumberlist@val and \gls@nonumberlist@nr 72
\glsaccsupp: added braces around actual text argument 363	translate: changed \val and \nr to \gls@translate@val and \gls@translate@nr 28
\glsentrycounterlabel: bug fix: move conditional inside command 218	numberedsection: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr 9
\GlsEntryCounterLabelPrefix: new 215	4.42 (2019-01-06)
\glsentryitem: bug fix: move conditional inside command 218	\@gls@@automake@immediate: new .. 182
\glsrefentry: bug fix: move conditional inside command 217	\@gls@automake@immediate: new ... 181
\glsresetsubentrycounter: bug fix: move conditional inside command . 217	\gls@automake@nr: new 33
\glsstepentry: bug fix: move conditional inside command 217	\glsfieldedef: changed from \edef to \protected@csedef 85
\glsstepsubentry: bug fix: move conditional inside command 217	\glsfieldxdef: changed from \edef to \protected@csxdef 85
\glssubentrycounterlabel: bug fix: move conditional inside command . 218	\ifglsautomake: now defined explicitly instead of through boolean key 33
\glssubentryitem: bug fix: move conditional inside command 218	noglossaryindex: new 37
\showglongnameaccess: bug fix: corrected field (was showing text access field) 399	automake: switch from boolean to choice 33
4.40 (2018-06-01)	4.42 (??)
\istfile: changed \def to \providecommand 189	altlong4col-booktabs: removed superfluous \glspatchLToutput . 296
\makenoidxglossaries: false 185	4.43 (2019-09-28)
4.41 (2018-07-23)	\glsnoidxstripaccents: add check for LaTeX version 2019/10/01 25
\@gls@override@glossary: new 35	4.44 (2019-12-06)
General: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr 214	\@glsprefix@record@hook: new 275
debug: changed \val and \nr to \gls@debug@val and \gls@debug@nr 5	4.45 (2020-02-13)
seenoindex: changed \val and \nr to \gls@seenoindex@val and \gls@seenoindex@nr 8	\@do@write@glslabels: new 32
kernelglossredefs: new 35	\@gls@showtarget: new 7
\glossary: added warning 35	\@GLSdesc@: added accessibility support 380
\gls@original@glossary: new 34	\@GLSdescplural@: added accessibility support 381
\gls@original@makeglossary: new .. 34	\@GLSfirst@: added accessibility support 379
\makeglossaries: removed redefinition of \makeglossary 183	\@GLSfirstplural@: added accessibility support 379
\makeglossary: added warning 35	\@GLSname@: added accessibility support 380
	\@GLSplural@: added accessibility support 379
	\@GLSsymbol@: added accessibility support 381
	\@GLSsymbolplural@: added accessibility support 381
	\@GLStext@: added accessibility support 378
	\@GLSuseri@: added accessibility support 382

\@GLSuserii@: added accessibility support	382	\@glsname@: added accessibility support	380
\@GLSuseriii@: added accessibility support	382	\@glsplural@: added accessibility support	379
\@GLSuseriv@: added accessibility support	383	\@glsshowaccsupp: new	7
\@GLSuserv@: added accessibility support	383	\@glssymbol@: added accessibility support	381
\@GLSuservi@: added accessibility support	383	\@glssymbolplural@: added accessibility support	381
\@Gls@acentryname: added check for \glsshortaccessdisplay	159	\@glstext@: added accessibility support	378
\@Glsdesc@: added accessibility support	380	\@glsuseri@: added accessibility support	381
\@Glsdescplural@: added accessibility support	380	\@glsuserii@: added accessibility support	382
\@Glsfirst@: added accessibility support	379	\@glsuseriii@: added accessibility support	382
\@Glsfirstplural@: added accessibility support	379	\@glsuseriv@: added accessibility support	383
\@Glsname@: added accessibility support	380	\@glsuserv@: added accessibility support	383
\@Glsplural@: added accessibility support	379	\@glsuservi@: added accessibility support	383
\@Glsymbol@: added accessibility support	381	General: removed backward compatibility use of symbol key	357
\@Glsymbolplural@: added accessibility support	381	acronyms: changed \renewcommand to \def	18
\@Glstext@: added accessibility support	378	debug: showaccsupp	5
\@Glsuseri@: added accessibility support	382	restoremakeloss: new	32
\@Glsuserii@: added accessibility support	382	\@gls@accessibility: new	354
\@Glsuseriii@: added accessibility support	382	\@gls@accsupp@engine: new	354
\@Glsuseriv@: added accessibility support	383	\@gls@accessibility: new	354
\@Glsuserv@: added accessibility support	383	\@glsdefaultshortaccess: new	394
\@Glsuservi@: added accessibility support	383	\@glsdescriptionaccessdisplay: added check for existence	364
\@do@write@glslabels: new	32	\@glsdescriptionpluralaccessdisplay: added check for existence	364
\@domakeglossaries: new	31	\@glsentrydescpluralaccess: corrected field reference	360
\@gls@fieldaccess@display: new ..	363	\@glsentryfirstpluralaccess: switched to using \@gls@entry@field	360
\@glsdesc@: added accessibility support	380	\@glsentryparent: new	162
\@glsdescplural@: added accessibility support	380	\@glsentryprefix: added \@glsdetoklabel	274
\@glsfirst@: added accessibility support	379	\@glsentryprefix: added \@glsdetoklabel	274
\@glsfirstplural@: added accessibility support	379	\@glsentryprefixfirst: added \@glsdetoklabel	274
		\@glsentryprefixfirst: added \@glsdetoklabel	274

\Glsentryprefixfirstplural: added		\glssymbolpluralaccessdisplay:	
\glstoklabel 274		added check for existence 364	
\Glsentryprefixfirstplural: added		\glstextaccessdisplay: added check	
\glstoklabel 274		for existence 363	
\Glsentryprefixplural: added		\glsseriaccessdisplay: new 365	
\glstoklabel 274		\glsseriiaccessdisplay: new 365	
\Glsentryprefixplural: added		\glsseriiiaccessdisplay: new ... 365	
\glstoklabel 274		\glsserivaccessdisplay: new 366	
\Glsentrytitlecase: added existence		\glsservaccessdisplay: new 366	
check 161		\glsserviaccessdisplay: new 366	
\Glsentryuseriaccess: new 361		\ifglshaschildren: made robust 62	
\Glsentryuseriiaccess: new 361		\ifglshasfield: made robust 64	
\Glsentryuseriiiaccess: new 361		\ifglshaslong: made robust 63	
\Glsentryuserivaccess: new 361		\ifglshasprefix: added	
\Glsentryuservaccess: new 361		\glstoklabel 275	
\Glsentryuserviaccess: new 361		\ifglshasprefixfirst: added	
\glstfieldaccsupp: new 362		\glstoklabel 275	
\glstfirstaccessdisplay: added check		\ifglshasprefixfirstplural: added	
for existence 364		\glstoklabel 275	
\glstfirstpluralaccessdisplay:		\ifglshasprefixplural: added	
added check for existence 364		\glstoklabel 275	
\glstlongaccessdisplay: added check		\ifglshashshort: made robust 63	
for existence 365		\ifglshassymbol: made robust 63	
\glstlongpluralaccessdisplay: added		disablemakegloss: new 31	
check for existence 365		\makeglossaries: let \@makeglossary	
\glstnameaccessdisplay: added check		to \@gobble instead of \relax 183	
for existence 363		writelabels: new 33	
\glstpluralaccessdisplay: added		user1access: new 356	
check for existence 364		user2access: new 356	
\glstprefixsep: new 275		user3access: new 356	
\glstsee: switched to \newrobustcmd . 198		user4access: new 357	
\glstseeformat: switched to		user5access: new 357	
\newrobustcmd 198		user6access: new 357	
\glstseeitem: switched to		\xglstfieldaccsupp: new 362	
\newrobustcmd 199	4.46 (2020-03-19)		
\glstseelist: switched to		\@ifglossaryexists: new 60	
\newrobustcmd 198		\@printgloss@checkexists: new ... 200	
\glstshortaccessdisplay: added check		\@printgloss@checkexists@allowignored:	
for existence 365		new 200	
\glstshortaccsupp: new 362		\@printgloss@checkexists@noignored:	
\glstshortplaccsupp: new 363		new 200	
\glstshortpluralaccessdisplay:		\@printglossary: replaced	
added check for existence 365		\@ifglossaryexists with	
\glstshowaccsupp: new 7		\@printgloss@checkexists 202	
\glstshowtargetfont: new 7		\doifglossarynoexistsordo: switched	
\glstshowtargetouter: new 6		to starred form of	
\glstshowtargetsymbol: new 6		\@ifglossaryexists 62	
\glssymbolaccessdisplay: added		\@ifglossaryexists: added starred form 60	
check for existence 364		\s@ifglossaryexists: new 60	

\setglossarypreamble: switched to starred form of \ifglossaryexists	46	\glsdisplaynumberlist: changed \edef to \protected@edef	165
4.47 (2021-09-20)		\glsfielddef: changed \edef to \protected@edef	86
\@do@esc@wrglossary: changed \edef to \protected@edef	196	\glsfieldedef: changed \edef to \protected@edef	85
\@do@noesc@wrglossary: changed \edef to \protected@edef	193	\glsfieldfetch: changed \edef to \protected@edef	86
\@do@write@glslabels: fully expand list	32	\glsfieldgdef: changed \edef to \protected@edef	85
\@gls@default@acr@descplural: new	249	\glsfieldxdef: changed \edef to \protected@edef	85
\@gls@default@acr@firstplural: new	249	\glsIfListOfAcronyms: changed \edef to \protected@edef	19
\@gls@default@acr@plural: new	248	\glsmoveentry: changed \edef to \protected@edef	95
\@gls@noexpand@assign@default@plurals: new	248	\glsnavhyperlink: changed \edef to \protected@edef	280
\@glo@do@sortentries: changed \edef to \protected@edef	206	\GlsSetQuote: added check for backslash	177
\@glo@sortentries: changed \edef to \protected@edef	205	\glsstepsubentry: changed \edef to \protected@edef	217
\@glo@storeentry: changed \edef to \protected@edef	95	\ifglsfieldcseq: changed \edef to \protected@edef	87
\@gls@checkesc: new	128	\ifglshaschildren: changed \edef to \protected@edef	62
\@gls@deprecated@acr@warning: new	263	\ifignoredglossary: changed \edef to \protected@edef	69
\@gls@link: changed \edef to \protected@edef	119	\makenoidxglossaries: changed \edef to \protected@edef	185
\@gls@setupsort@use: changed \edef to \protected@edef	16	\new@glossaryentry: allow overwrite if defined in .glsdefs	78
\@gls@writedef: added a warning if entry has already been defined	80	\newignoredglossary: changed \edef to \protected@edef	69
\@write@glslabelsname: new	33	\ns@newglossary: changed \edef to \protected@edef	67
General: changed \x to \@gls@x	227	writeglslabelnames: new	33
\do@write@glslabelsname: new	33	\SetDescriptionDUAAcronymStyle: added deprecated warning	253
\forglsentries: changed \edef to \protected@edef	59	\SetDescriptionFootnoteAcronymStyle: added deprecated warnings	251
\gls@defdocnewglossaryentry: removed see warning	77	\SetDUASyle: added deprecated warnings	263
\gls@defglossaryentry: changed \edef to \protected@edef	89, 90	\SetFootnoteAcronymStyle: added deprecated warnings	258
\gls@docdef@addlabel: new	78	\SetSmallAcronymStyle: added deprecated warnings	261
\gls@docdef@labellist: new	78	4.47 (??)	
\gls@gldefs@addlabel: new	78	\@makeglossaries@warn@noprintglossary: new	184
\gls@gldefs@labellist: new	78		
\gls@save@numberlist: changed \edef to \protected@edef	199		
\glsadd: changed \edef to \protected@edef	166		
\glsaddall: changed \edef to \protected@edef	167		
\glsdefs@newdocentry: new	80		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	
\!	123, 126, 177, 479
\"	24, 123–126, 128, 129, 415, 476–479, 481
\#	128, 172, 505
\%	168, 174, 337, 338, 502, 506, 507
\&	41, 165, 425, 501
\'	24, 415
\.	12, 24, 407, 415
\=	24, 415
\?	123, 125, 177, 476, 478, 509
\@	79, 448
\@delimN	226, 540
\@do@wrglossary	185, 193, 196, 515, 520, 521
\@do@esc@wrglossary	193, 520
\@do@noesc@wrglossary	193, 520
\@do@wrglossary	166, 191, 501, 519, 520
\@do@write@glslabels	32, 421
\@glo@assign@sortkey	186, 515, 534
\@glo@list	59, 437
\@glo@sort	24, 414
\@glo@type	201, 524
\@glossarysec	8, 48, 49, 406, 429, 430
\@glossaryseclabel	9, 48, 49, 215, 406, 429, 430, 533
\@glossarysecstar	9, 48, 215, 406, 429, 430, 533
\@gls@@default@acr@descplural	248
\@gls@@default@acr@firstplural	248
\@gls@@default@acr@plural	248
\@gls@checkactual	127, 128, 480, 481
\@gls@checkbar	126, 127, 479, 480
\@gls@checkescactual	125, 478
\@gls@checkescbar	125, 126, 478, 479
\@gls@checkesclevel	126, 479
\@gls@checkescquote	124, 178, 477, 478, 509, 510
\@gls@checklevel	127, 480
\@gls@checkquote	124, 176, 477, 508
\@gls@default@entryfmt	106, 115, 465, 471, 472
\@gls@expand@field	22, 76, 77, 82, 83, 248, 251, 253, 255, 257, 258, 260–263, 395–399, 413, 446, 447, 450, 554, 556, 557, 559, 561, 563, 565
\@gls@extramakeindexopts	175, 183, 508, 513
\@gls@fixbraces	198, 522
\@gls@noexpand@assign@default@plurals	248, 395
\@gls@noexpand@field	22, 75, 76, 413, 446
\@gls@noidx@no@sanitizesort	24, 414
\@gls@noidx@nosanitizesort	187, 414, 517
\@gls@noidx@sanitizesort	414
\@gls@nosanitizesort	23, 187, 414, 517
\@gls@sanitizesort	23, 187, 414, 517
\@gls@xdycheckbackslash	129, 481, 482
\@gls@xdycheckquote	128, 129, 481
\@glslocalreset	99, 101, 461, 463
\@glslocalunset	98, 101, 461, 463
\@glsreset	99, 101, 461–463
\@glsshowtarget	6, 7, 404, 405
\@glsunset	98, 101, 461, 463
\@newglossaryentry@defcounters	100, 462
\@this@glo@	59, 437
\@ACRfull	231, 542
\@ACRfullpl	232, 542
\@ACRlong	154, 231, 495, 542
\@ACRlongpl	156, 232, 496, 542
\@ACRshort	151, 231, 492, 542
\@ACRshortpl	152, 232, 493, 542
\@Acrfull	230, 541, 542
\@Acrfullpl	231, 232, 542
\@Acrlong	154, 230, 494, 542
\@Acrlongpl	156, 232, 495, 542

\@Acrshort	150, 492	\@Glsfirst	138, 486
\@Acrshortpl	152, 493	\@Glsfirst@	138, 486
\@Alpha	192, 195, 519–521	\@Glsfirstplural	140, 487
\@GLS	133, 484	\@Glsfirstplural@	140, 487
\@GLS@	133, 279, 484	\@Glsname	141, 487, 488
\@GLSdesc	142, 488	\@Glsname@	141, 488
\@GLSdesc@	142, 488	\@Glspl	134, 484
\@GLSdescplural	143, 488	\@Glspl@	102, 104, 134, 278, 463, 465, 484, 485
\@GLSdescplural@	143, 488	\@Glsplural	139, 487
\@GLSfirst	138, 139, 487	\@Glsplural@	139, 487
\@GLSfirst@	139, 487	\@Glsymbol	143, 489
\@GLSfirstplural	140, 487	\@Glsymbol@	143, 489
\@GLSfirstplural@	140, 487	\@Glsymbolplural	144, 489
\@GLSname	141, 488	\@Glsymbolplural@	144, 489
\@GLSname@	141, 488	\@Glstext	138, 486
\@GLSpl	135, 485	\@Glstext@	138, 486
\@GLSpl@	135, 279, 485	\@Glsuser@i	382, 383
\@GLSplural	139, 487	\@Glsuseri	145, 489
\@GLSplural@	139, 487	\@Glsuseri@	145, 489
\@GLSsymbol	143, 489	\@Glsuserii	146, 490
\@GLSsymbol@	143, 144, 489	\@Glsuserii@	146, 490
\@GLSsymbolplural	144, 489	\@Glsuseriii	146, 490
\@GLSsymbolplural@	144, 489	\@Glsuseriii@	146, 147, 490
\@GLStext	137, 486	\@Glsuseriv	147, 490
\@GLStext@	137, 138, 486	\@Glsuseriv@	147, 490
\@GLSuseri	145, 489	\@Glsuserv	148, 491
\@GLSuseri@	145, 489, 490	\@Glsuserv@	148, 491
\@GLSuserii	146, 490	\@Glsuservi	149, 491
\@GLSuserii@	146, 490	\@Glsuservi@	149, 491
\@GLSuseriii	147, 490	\@Mi	298
\@GLSuseriii@	147, 490	\@PGLS	278
\@GLSuseriv	147, 148, 491	\@PGLS@	278
\@GLSuseriv@	148, 491	\@PGLSpl	279
\@GLSuserv	148, 491	\@PGLSpl@	279
\@GLSuserv@	148, 491	\@Pgls	277
\@GLSuservi	149, 491	\@Pgls@	277
\@GLSuservi@	149, 491	\@Pglspl	277
\@Gls	132, 483	\@Pglspl@	277
\@Gls@	101, 103, 132, 277, 463, 464, 483	\@Roman	192, 195, 519, 520
\@Gls@acrentryname	233, 497, 543	\@acrfull	229, 541
\@Gls@entry@field	83, 158–164, 451, 496–500	\@acrfullpl	231, 542
\@Gls@entryname	158, 233, 497, 543	\@acrlong	153, 230, 494, 541
\@GlsSetXdyFirstLetterAfterDigits		\@acrlongpl	155, 231, 495, 542
	168, 502	\@acrshort	149, 230, 491, 541, 542
\@GlsSetXdyNumberGroupOrder	169, 502, 503	\@acrshortpl	151, 231, 232, 493, 542
\@Glsdesc	141, 142, 488	\@addtoacronymlists	19, 411, 412
\@Glsdesc@	142, 488	\@afterheading	286, 341
\@Glsdescplural	142, 488	\@alpha	192, 195, 519–521
\@Glsdescplural@	142, 488	\@arabic	192, 195, 519, 520

\@auxout	66, 67, 102, 182, 183, 185, 188, 199, 204, 281, 440, 441, 464, 513, 515, 517, 523, 525, 526	\@empty	12, 16, 19, 34, 37–39, 50, 52, 55, 58, 59, 90, 91, 96, 120, 121, 131–135, 150–156, 170, 173, 175, 180, 181, 189, 190, 197, 223, 249, 251, 253, 254, 256, 258, 259, 261, 263, 265, 335, 337, 339, 376–378, 408, 410, 411, 421, 423–425, 431, 432, 434, 436, 437, 456, 459, 460, 474, 475, 483–485, 492–496, 503, 505, 507, 511, 512, 518, 522, 538, 554, 556, 558–562, 564, 565, 567
\@backslashchar	122, 123, 129, 177, 476, 481, 482	\@end@fixbraces	198, 522
\@bsphack	190, 518	\@endfortrue ...	29, 62, 81, 281, 418, 438, 449
\@cGls	103, 464	\@esphack	191, 519
\@cGls@	101, 103, 463, 464	\@expandtwoargs	37, 424
\@cGlspl	104, 465	\@firstofone	24, 32, 414, 415, 420
\@cGlspl@	102, 104, 463, 465	\@firstofthree	117, 131, 134, 136, 150, 151, 153, 155, 376–378, 473, 483–485, 492–495
\@cclv	298, 299	\@firstoftwo	27, 28, 80, 81, 117, 134, 135, 151–153, 155, 156, 417, 418, 448, 449, 473, 484, 485, 493–496
\@cgls	103, 464	\@for	29, 37, 39, 51, 52, 59, 80, 81, 122, 170, 172, 182– 184, 191, 198, 205, 235, 249, 252, 254, 256, 258, 261, 263, 265, 281, 282, 335, 418, 424, 425, 431, 432, 437, 448, 449, 476, 503, 504, 513, 514, 519, 523, 526, 544, 554, 556, 558, 560, 561, 564, 566, 567
\@cgls@	101, 103, 463, 464	\@glo@desc	93, 458
\@cglspl	104, 465	\@glo@symbol	93, 458
\@cglspl@	101, 104, 463, 465	\@glo@access	355, 358, 363
\@chapter	39, 425	\@glo@addchildren ..	205, 209, 526, 527, 530
\@classoptionslist	37, 424	\@glo@assign@sortkey	184, 186, 216, 514, 515, 534
\@closegls	180, 511	\@glo@autosee	94, 458
\@colht	298	\@glo@autoseehook	94, 459
\@colroom	298, 299	\@glo@check@mkidxrangechar	121, 122, 196, 334, 335, 475, 521
\@currentlabelname	9, 215, 406, 533	\@glo@check@sortallowed	14–17, 184, 187, 409, 410, 514, 517
\@curroptions	37, 424	\@glo@childlist	205, 526
\@declaredoptions	37, 424	\@glo@counter	72, 89, 92, 443, 455, 457
\@delimN	226, 540	\@glo@counterprefix	188, 193, 195–197, 223, 227, 517, 520–522, 538, 540
\@delimR	226, 539, 540	\@glo@default@sorttype	14, 186, 207–209, 408, 409, 515, 528, 529
\@disable@onlypremakeg	183, 425, 514	\@glo@defaultcounter	92, 457
\@disable@premakecs	39, 425	\@glo@desc ..	70, 88–90, 93, 443, 454–456, 458
\@disabled@gl saddxdycounters	52, 431, 432	\@glo@descaccess	356, 358, 359
\@do@addcounter	51, 431	\@glo@descplural ...	70, 88, 89, 443, 454, 455
\@do@auxoutstuff	203, 204, 525, 526		
\@do@glossentry	219, 353, 354, 536		
\@do@gls@checkesc	123		
\@do@gls@getcounterprefix	193, 196, 520, 521		
\@do@gls@islistofacronyms	19, 412		
\@do@glssee	94, 459		
\@do@ifinlist	50, 431		
\@do@newglossaryentry ..	233, 247, 248, 250–255, 257, 258, 260–263, 265, 395– 399, 543, 553–557, 559–561, 563–565, 567		
\@do@seeglossary ...	185, 198, 515, 522, 523		
\@do@subglossentry	220, 221, 354, 537		
\@do@wrglossary	120, 474, 519		
\@do@write@glslabels	33, 421		
\@do@writeaux@info	199, 523		
\@domakeglossaries	31, 32, 182, 185, 420, 513, 514		
\@ehc	298		

\@glo@descpluralaccess	356, 358, 359	\@glo@sort	14, 17, 23, 24, 70, 89, 92, 95, 96, 409, 410, 414, 443, 455, 457, 459, 460
\@glo@do@sortentries	205, 526, 527	\@glo@sortedinsert	206, 527
\@glo@entry	167, 502	\@glo@sortentries ..	208, 209, 526, 528, 529
\@glo@entryprefix	273	\@glo@sorthandler@case	208, 528, 529
\@glo@entryprefixfirst	273	\@glo@sorthandler@letter ...	208, 527, 528
\@glo@entryprefixfirstplural ..	273, 274	\@glo@sorthandler@nocase ...	209, 528, 529
\@glo@entryprefixplural	273	\@glo@sorthandler@word	208, 527, 528
\@glo@esclabel	95, 96, 459, 460	\@glo@sortinghandler ...	205, 206, 526, 527
\@glo@etext	106–108, 466, 467	\@glo@sortinglist	205, 206, 209, 526, 527, 529, 530
\@glo@first	71, 89, 92, 93, 260, 398, 443, 455, 457, 458, 563	\@glo@sortmacro@case	529
\@glo@firstaccess	355, 358	\@glo@sortmacro@def	529
\@glo@firstplural	71, 89, 92, 399, 443, 455, 457	\@glo@sortmacro@def@do	530
\@glo@firstpluralaccess	355, 358	\@glo@sortmacro@letter	528
\@glo@grabfirst	211, 531	\@glo@sortmacro@nocase	529
\@glo@label	74, 75, 82, 83, 85–87, 89–94, 100, 165, 166, 273, 274, 329, 358, 359, 445, 450–459, 462, 501	\@glo@sortmacro@standard	529
\@glo@list	94, 459	\@glo@sortmacro@use	530
\@glo@long ..	63, 74, 90, 92, 439, 445, 455, 457	\@glo@sortmacro@word	528
\@glo@longaccess	356, 358, 359	\@glo@sorttype	186, 210, 211, 216, 515, 530, 531, 534
\@glo@longpl	74, 90, 92, 248– 250, 252, 255, 257, 260, 262, 395, 445, 455, 457, 553–555, 557, 559, 561, 563, 565	\@glo@storeentry	14–16, 409, 410, 459
\@glo@longpluralaccess	356, 358, 359	\@glo@suffix	121, 122, 196, 335, 475, 476, 521
\@glo@name	14, 70, 75, 89, 91–93, 409, 443, 446, 455–457	\@glo@symbol	63, 71, 89, 93, 254, 259, 260, 439, 443, 455, 458, 558, 562, 563
\@glo@no@assign@sortkey	184, 514, 534	\@glo@symbolaccess	355, 358, 398
\@glo@nonumberlist	73, 444	\@glo@symbolplural ..	71, 89, 93, 443, 455, 458
\@glo@numfmt	196, 335, 521	\@glo@symbolpluralaccess	355, 358
\@glo@parent	16, 72, 89, 91, 92, 96, 206, 410, 444, 455–457, 459, 460, 527	\@glo@text	70, 89, 92, 93, 132–136, 157–159, 255, 274, 397, 443, 455, 457, 458, 483–486, 496, 497, 559
\@glo@plural ..	70, 89, 91, 92, 397, 443, 455–457	\@glo@textaccess	355, 358, 395–397
\@glo@pluralaccess	355, 358	\@glo@thislabel	95, 459
\@glo@prefix	10, 72, 73, 89, 96, 121, 122, 196, 334, 335, 407, 444, 455, 459, 460, 475, 476, 521	\@glo@thislettergrp	211, 212, 530–532
\@glo@range	196, 334, 335, 521	\@glo@thisvalue	64, 65, 439, 440
\@glo@see	72, 89, 94, 444, 455, 459	\@glo@tmp	82, 83, 197, 450, 451, 522
\@glo@seeautonumberlist ..	10, 72, 407, 444	\@glo@type ..	9, 16, 71, 89, 90, 92–94, 166, 167, 183, 189, 201–206, 209, 210, 214, 215, 233, 249, 251, 253, 256, 258, 261, 263, 265, 280, 282, 406, 410, 443, 455–458, 501, 502, 513, 518, 524–527, 530, 533, 543, 554, 556, 558, 559, 561, 564, 565, 567
\@glo@short	64, 74, 90, 92, 398, 439, 445, 455, 457	\@glo@types	32, 59, 67, 99, 100, 167, 182–184, 271, 329, 421, 437, 440, 441, 462, 502, 513, 514, 569
\@glo@shortaccess ..	356, 358, 359, 395–398	\@glo@useri	73, 89, 92, 445, 455, 457
\@glo@shortpl	74, 90, 92, 247–250, 252, 255, 257, 260, 262, 395, 398, 445, 455, 457, 553, 555, 557, 559, 561, 563, 565	\@glo@useriaccess	356, 358, 359
\@glo@shortpluralaccess	356, 358, 359	\@glo@userii	73, 89, 92, 445, 455, 457
		\@glo@useriaccess	356, 358, 359

<code>\@glo@useriii</code>	74, 89, 92, 445, 455, 457	<code>\@gls@classI</code>	170, 171, 503, 504
<code>\@glo@useriiiaccess</code>	356, 358, 359	<code>\@gls@classII</code>	170, 171, 504
<code>\@glo@useriv</code>	74, 89, 92, 445, 455, 457	<code>\@gls@codepage</code>	204, 436, 526
<code>\@glo@userivaccess</code>	357–359	<code>\@gls@counter</code>	116, 119–121, 166, 188, 196, 197, 335, 472, 474, 475, 501, 517, 521, 522
<code>\@glo@userv</code>	74, 89, 92, 445, 455, 457	<code>\@gls@counterwithin</code>	12, 13, 408
<code>\@glo@uservaccess</code>	357–359	<code>\@gls@ctr</code>	51, 431
<code>\@glo@uservi</code>	74, 89, 92, 445, 455, 457	<code>\@gls@currentlettergroup</code>	210, 212, 530–532
<code>\@glo@userviaccess</code>	357–359	<code>\@gls@debugfalse</code>	5, 404
<code>\@glodesc</code>	93, 458	<code>\@gls@debugtrue</code>	5, 6, 404
<code>\@glolist@</code>	90, 456	<code>\@gls@declareoption</code> 10, 11, 17, 18, 21, 22, 27, 30– 33, 36, 37, 407, 411, 413, 417, 419–421, 423
<code>\@glo@name</code>	93, 457	<code>\@gls@default</code>	105, 465
<code>\@glossary@default@style</code>	9, 11, 201, 224, 266, 406, 407, 524, 538, 539, 567	<code>\@gls@default@value</code> 63–65, 75, 76, 89, 91–93, 259, 273, 439, 440, 443, 446, 447, 455–458, 562
<code>\@glossaryentryfield</code>	96, 459, 460	<code>\@gls@deffile</code>	78–80, 448, 449
<code>\@glossarysection</code>	47, 428, 429	<code>\@gls@define@glossaryentrycounter</code> 13, 38, 215, 217, 408, 424, 534
<code>\@glossarystyle</code>	201, 202, 214, 524, 525, 533	<code>\@gls@define@glossarysubentrycounter</code> 38, 215, 217, 408, 424, 534
<code>\@glossarysubentryfield</code>	96, 459, 460	<code>\@gls@defsort</code>	14–16, 93, 409, 410, 458
<code>\@gls</code>	131, 483	<code>\@gls@defsortcount</code>	14–16, 68, 409, 410, 441
<code>\@gls@</code> ...	101, 103, 131, 276, 277, 463, 464, 483	<code>\@gls@deprecated@acr@warning</code> 251, 253, 255, 258, 261, 263
<code>\@gls@@automake@immediate</code>	182, 513	<code>\@gls@do@acronymsdef</code>	18, 38, 69, 411, 424, 442
<code>\@gls@@link</code>	118, 473	<code>\@gls@do@indexdef</code>	37–39, 69, 423–425, 442
<code>\@gls@Hcounter</code>	120, 121, 474, 475	<code>\@gls@do@numbersdef</code>	36–38, 69, 423, 424, 442
<code>\@gls@ReturnAfterFi</code>	227, 540	<code>\@gls@do@symbolsdef</code>	36, 69, 423, 442
<code>\@gls@actualchar</code> 96, 125, 127, 174, 338, 460, 478, 480, 506	<code>\@gls@do@symbolssdef</code>	38, 39, 424, 425
<code>\@gls@addpredefinedattributes</code> 169, 178, 432, 433, 503, 510	<code>\@gls@doautomake</code> 33, 34, 182, 184, 421, 513, 514
<code>\@gls@adjustmode</code>	166, 501, 502	<code>\@gls@docheckquotedef</code>	176–178, 508–510
<code>\@gls@after</code>	19, 412	<code>\@gls@docloadedfalse</code>	5, 403
<code>\@gls@automake</code>	184, 511, 514	<code>\@gls@docloadedtrue</code>	4, 5, 403
<code>\@gls@automake@immediate</code> ...	182, 512, 513	<code>\@gls@dodeflistparser</code>	183, 514
<code>\@gls@before</code>	19, 412	<code>\@gls@doentrycounterdef</code>	38, 39, 424, 425
<code>\@gls@between</code>	282	<code>\@gls@doentrydef</code>	115, 471, 472
<code>\@gls@body</code>	158, 159, 497	<code>\@gls@dolast</code>	198, 199, 523
<code>\@gls@checkactual</code>	123, 177, 477, 480, 481, 509	<code>\@gls@donext</code>	198, 199, 523
<code>\@gls@checkbar</code> ..	123, 177, 477, 479, 480, 509	<code>\@gls@donext@def</code>	165, 501
<code>\@gls@checkedmkidx</code> 122–129, 176–178, 476–482, 508–510	<code>\@gls@dosubentrycounterdef</code>	38, 39, 424, 425
<code>\@gls@checkesc</code>	123, 177	<code>\@gls@dothiswrite</code>	180–182, 511–513
<code>\@gls@checkescactual</code>	123, 177, 476, 478, 509	<code>\@gls@elem</code>	281
<code>\@gls@checkescbar</code> ..	123, 177, 477–479, 509	<code>\@gls@enablesavenonumberlist</code>	79, 444, 448
<code>\@gls@checkesclevel</code>	123, 177, 479	<code>\@gls@encapchar</code>	125, 126, 174, 196, 198, 335, 338, 478–480, 506, 521, 522
<code>\@gls@checkescquote</code> 123, 176–178, 476–478, 508–510	<code>\@gls@entry@count</code>	102, 464
<code>\@gls@checklevel</code> ...	123, 177, 477, 480, 509		
<code>\@gls@checkmkidxchars</code> 95, 96, 121, 176, 185, 195–197, 334, 335, 459, 475, 476, 508, 515, 521, 522		
<code>\@gls@checkquote</code> ...	123, 176, 476, 477, 508		

<code>\@gls@entry@field</code>	82, 83, 100, 101, 158–164, 359–361, 450, 451, 462, 496–500	<code>\@gls@link@nocheckfirsthyper</code>	137, 150–156, 486, 492–496
<code>\@gls@escbsdq</code>	123, 175, 339, 476, 507	<code>\@gls@link@opts</code> .	119, 250, 257, 474, 555, 560
<code>\@gls@expand@fields</code>	76, 77, 446, 447	<code>\@gls@list</code>	281, 282
<code>\@gls@expandonce</code>	77, 447	<code>\@gls@listsuffix</code>	50, 431
<code>\@gls@extramakeindexopts</code> ...	183, 510, 513	<code>\@gls@loadlist</code>	11, 266, 407, 567
<code>\@gls@fetchfield</code>	64, 440, 449	<code>\@gls@loadlong</code>	11, 266, 407, 567
<code>\@gls@field@link</code>	84, 137–149, 378–383, 451, 452, 486–491	<code>\@gls@loadsuper</code>	11, 266, 407, 567
<code>\@gls@fieldaccess@display</code>	363–366	<code>\@gls@loadtree</code>	11, 266, 407, 567
<code>\@gls@firsttok</code>	210, 211, 530, 531	<code>\@gls@local@increment@currcount</code> ...	101, 463, 464
<code>\@gls@fixbraces</code>	94, 459, 522	<code>\@gls@loclist</code>	186, 187, 211, 212, 515, 516, 531, 532
<code>\@gls@forbidtexext</code>	67, 441	<code>\@gls@map</code>	80, 81, 448, 449
<code>\@gls@fulllist</code>	32	<code>\@gls@missinglang@warn</code> ...	21, 42, 413, 426
<code>\@gls@get@counterprefix</code>	197, 522	<code>\@gls@missingnumberlist</code>	93, 446, 458
<code>\@gls@getbody</code>	158, 497	<code>\@gls@noaccess</code>	363
<code>\@gls@getcounter</code>	442	<code>\@gls@noexpand@fields</code>	77, 446, 447
<code>\@gls@getcounterprefix</code> .	193, 196, 520–522	<code>\@gls@nohyperlist</code> .	20, 69, 119, 412, 442, 474
<code>\@gls@getgrouptitle</code>	185, 222, 282, 515, 537, 538	<code>\@gls@noidx@do</code>	210, 530, 531
<code>\@gls@glossary</code>	190, 518	<code>\@gls@noidx@getgrouptitle</code> ..	185, 515, 538
<code>\@gls@gobbleopt</code>	66, 441	<code>\@gls@noidx@sanitizesort</code> 23,	187, 414, 517
<code>\@gls@grptitle</code>	222, 280, 282, 537	<code>\@gls@noidx@setsanitizesort</code>	26, 187, 416, 517
<code>\@gls@hyp@opt</code>	84, 103, 104, 118, 131–156, 229–232, 275–279, 451, 452, 464, 465, 473, 483–496, 541, 542	<code>\@gls@noidx@loc@list@finalsep</code> ...	187, 516
<code>\@gls@hyp@opt@cs</code>	117, 473	<code>\@gls@noidx@loc@list@prev</code> 187,	213, 516, 532
<code>\@gls@hypergroup</code>	281	<code>\@gls@noidx@loc@list@sep</code> .	186, 213, 516, 532
<code>\@gls@ifinlist</code>	51, 431	<code>\@gls@noref@warn</code> ...	185, 210, 515, 517, 530
<code>\@gls@ifnotmeasuring</code>	97, 460	<code>\@gls@numberlink</code>	226, 227, 540
<code>\@gls@igtype</code>	69, 443	<code>\@gls@numbersdef</code>	36, 423
<code>\@gls@increment@currcount</code> ..	101, 463, 464	<code>\@gls@numlist@lastsep</code>	165, 501
<code>\@gls@indexdef</code>	37, 423	<code>\@gls@numlist@nextsep</code>	165, 501
<code>\@gls@initnonumberlist</code> ...	73, 89, 444, 455	<code>\@gls@numlist@sep</code>	165, 501
<code>\@gls@islistofacronyms</code>	19, 412	<code>\@gls@old@chapter</code>	39, 425
<code>\@gls@keylist</code>	394	<code>\@gls@oldnewglossaryentryposthook</code> .	358
<code>\@gls@keymap</code> 73, 80–83, 273, 357, 445, 448–450		<code>\@gls@oldnewglossaryentryprehook</code> ..	357, 358
<code>\@gls@label</code>	185, 188, 193, 196, 515, 517, 520, 521	<code>\@gls@onlypremakeg</code>	39, 425
<code>\@gls@labelsfile</code>	32, 33, 421	<code>\@gls@order</code>	180, 181, 511, 512
<code>\@gls@langmod</code>	180, 181, 511, 512	<code>\@gls@org@LT@output</code>	298
<code>\@gls@levelchar</code>	96, 126, 127, 174, 338, 460, 479, 480, 506	<code>\@gls@org@glsnoidx@displayloc</code> 187,	516, 517
<code>\@gls@link</code>	118, 132–137, 150–157, 376–378, 473, 474, 483–486, 492–496	<code>\@gls@org@glsseeformat</code>	187, 516, 517
<code>\@gls@link@checkfirsthyper</code>	131–136, 473, 483–485	<code>\@gls@override@glossary</code> ..	35, 36, 422, 423
<code>\@gls@link@label</code> 119, 250, 257, 474, 555, 560		<code>\@gls@patchtabularx</code>	97, 460
		<code>\@gls@preglossaryhook</code>	203, 524, 525
		<code>\@gls@prevlevel</code> .	309, 310, 329–332, 347, 348
		<code>\@gls@provide@newglossary</code> ...	67, 440, 441
		<code>\@gls@quotechar</code>	124–127, 174–176, 178, 338, 477–480, 506–510

\@gls@reference 185, 188, 515, 517, 532	\@gls@updatechecked
\@gls@removespaces 227, 540 122, 123, 176, 177, 476, 477, 508, 509	
\@gls@renewglossary 179, 510, 518	\@gls@usetranslator	.. 28, 41, 417, 418, 426
\@gls@replacementtext 362, 363	\@gls@value 76, 161, 446, 447, 498
\@gls@rest 158, 497	\@gls@warnonglossdefined	22, 200, 413, 523
\@gls@restoreat 79, 448	\@gls@warnontheGLOSSdefined
\@gls@roman 54, 335, 336, 433 22, 218, 413, 536	
\@gls@sanitized@tmp 122, 476	\@gls@write@entrycounts 102, 464
\@gls@sanitizedesc 29, 413, 419	\@gls@writedef 79, 448
\@gls@sanitizename 414	\@gls@writeisthook 173, 175, 506, 507
\@gls@sanitizesort 14, 409, 414	\@gls@x 227
\@gls@sanitizesymbol 29, 30, 414, 419	\@gls@xdy@locationlist	. 170, 433, 503, 504
\@gls@saveentrycounter	. 120, 166, 474, 501	\@gls@xdycheckbackslash	122, 476, 481, 482
\@gls@savenonumberlist 73, 444	\@gls@xdycheckquote 122, 476, 481
\@gls@see@noindex 8, 72, 405, 406, 444	\@gls@xref 197, 198, 522
\@gls@setacrstyle 29, 30, 38, 419, 424	\@gls@Alphacompositor	. 45, 54, 336, 428, 434
\@gls@setcounter 68, 441, 442	\@gls@Hlocref 193, 196, 520, 521
\@gls@setdefault@glslink@opts	. 119, 474	\@gls@acronymlists
\@gls@setsort	19, 20, 59, 233–235, 249, 251–254, 256,	
..... 14–17, 120, 166, 409, 410, 474, 501		258, 261, 263, 265, 270, 411, 412, 437,	
\@gls@setupshortcuts 38, 424	543, 544, 554, 556, 558–561, 564–567, 569	
\@gls@setupsort@def 409	\@gls@addkey 83, 450
\@gls@setupsort@none 410	\@gls@addstoragekey 82, 450
\@gls@setupsort@standard 409	\@gls@addxdyattribute 51, 52, 431, 432
\@gls@setupsort@use 410	\@gls@defaultplural 445
\@gls@sort 212, 531	\@gls@defaultsort 14, 409, 446
\@gls@sort@A 206, 207, 527, 528	\@gls@desc 141, 488
\@gls@sort@B 207, 527, 528	\@gls@desc@ 141, 488
\@gls@startswithexpandonce 76, 447	\@gls@descplural 142, 488
\@gls@storenonumberlist	.. 73, 93, 444, 458	\@gls@descplural@ 142, 488
\@gls@symbolsdef 36, 423	\@gls@disp 136, 485
\@gls@this 191, 519	\@gls@entry	. 32, 33, 99, 100, 102, 421, 462, 464
\@gls@thisHloc 197, 522	\@gls@entrytitlecase 161, 498, 499
\@gls@thisentryname 32, 33	\@gls@first 138, 486
\@gls@thisfield 64, 65, 440	\@gls@first@ 138, 486
\@gls@thislabel	\@gls@firstletter 168, 169, 502
..... 62, 198, 199, 209, 438, 523, 529		\@gls@firstplural 140, 487
\@gls@thislist 165, 501	\@gls@firstplural@ 140, 379, 487
\@gls@thisloc 197, 522	\@gls@hypernumber 226, 539, 540
\@gls@thisval 81, 449	\@gls@isacronymlistfalse 20, 412
\@gls@title 47, 428	\@gls@isacronymlisttrue 20, 412
\@gls@tmp	... 7, 16, 42, 54, 55, 77, 122, 191,	\@gls@link
281, 282, 405, 410, 426, 434, 447, 476, 518		120, 130, 131, 166, 280, 474, 482, 483, 501	
\@gls@tmpb	124–129, 176–178, 477–482, 508–510	\@gls@localreset 98, 101, 461, 463
\@gls@toc 48, 429, 430	\@gls@localunset 98, 101, 461, 463
\@gls@type	... 182, 184, 235, 249, 252, 254,	\@gls@locref
256, 258, 261, 263, 265, 329, 513, 514,		188, 193, 195, 196, 334, 335, 517, 520, 521	
544, 554, 556, 558, 560, 561, 564, 566, 567		\@gls@minrange 169–171, 336, 503, 504
		\@gls@name 140, 487

<code>\@glsname@</code>	140, 141, 487	<code>\@ifclassloaded</code>	5, 12, 47, 403, 408, 429
<code>\@glsnavhypertarget</code>	280	<code>\@ifglossaryexists</code>	60, 200, 437, 523
<code>\@glsnextpages</code>	202, 525, 534	<code>\@ifl@t@r</code>	25, 415
<code>\@glsnodelsc</code>	89, 90, 93, 445, 455, 456, 458	<code>\@ifnextchar</code>	68, 117, 441, 473
<code>\@glsnoname</code>	89, 91, 93, 445, 455–457	<code>\@ifpackageloaded</code>	
<code>\@glsnonextpages</code>	202, 525, 534	..	4, 9, 27–29, 41, 58, 97, 164, 175, 353,
<code>\@glsnumberformat</code>	116, 119, 166,		403, 406, 417, 418, 426, 436, 460, 500, 507
	188, 196, 334, 335, 472, 474, 501, 517, 521	<code>\@ifstar</code>	60, 67, 82, 83, 117,
<code>\@glsopenfile</code>	179, 189, 510, 511, 518		168, 169, 228, 437, 441, 450, 473, 502, 541
<code>\@glsorder</code>	182, 183, 419, 513	<code>\@ifundefined</code>	41,
<code>\@glspl</code>	134, 484		281, 288, 299, 310, 317, 330, 331, 347, 426
<code>\@glspl@</code> .	101, 104, 134, 276, 278, 463, 465, 484	<code>\@ignored@glossaries</code> .	32, 69, 421, 442, 443
<code>\@glsplural</code>	139, 487	<code>\@input@</code>	203, 525
<code>\@glsplural@</code>	139, 487	<code>\@istfilename</code>	182, 183, 428, 513
<code>\@glsprefix@record@hook</code>	276–279	<code>\@makecol</code>	298, 299
<code>\@glsreset</code>	97, 101, 461, 463	<code>\@makeglossaries@warn@noprintglossary</code>	
<code>\@glssee</code>	94, 198, 459, 523	183
<code>\@glsshowacccsupp</code>	6, 355, 404, 405	<code>\@makeglossary</code>	31, 32, 183, 420, 510, 513, 514
<code>\@glsshowtarget</code>	5, 6, 130, 404, 405, 482	<code>\@minus</code>	285, 305, 323
<code>\@glssymbol</code>	143, 488	<code>\@mkboth</code>	47, 48, 429
<code>\@glssymbol@</code>	143, 489	<code>\@newglossary</code>	66, 67, 440–442
<code>\@glssymbolplural</code>	144, 489	<code>\@newglossaryentry@defcounters</code>	
<code>\@glssymbolplural@</code>	144, 489	94, 100, 458, 459, 462
<code>\@glstarget</code> .	130, 131, 219, 281, 482, 483, 536	<code>\@newglossaryentryposthook</code>	
<code>\@glstext</code>	137, 486	82, 83, 94, 273, 358, 450, 458, 459
<code>\@glstext@</code>	137, 486	<code>\@newglossaryentryprehook</code>	82,
<code>\@glsunset</code>	98, 101, 461, 463		83, 88, 90, 273, 357, 450, 454, 455, 459
<code>\@glsuseri</code>	145, 489	<code>\@nil</code>	19, 94, 121–
<code>\@glsuseri@</code>	145, 489		123, 158, 176, 177, 196, 198, 210–212,
<code>\@glsuserii</code>	145, 490		226, 227, 334, 335, 412, 459, 475–477,
<code>\@glsuserii@</code>	145, 490		497, 508, 509, 521, 522, 530, 531, 539, 540
<code>\@glsuseriii</code>	146, 490	<code>\@nnil</code>	19, 198, 412, 523
<code>\@glsuseriii@</code>	146, 490	<code>\@no@makeglossaries</code>	
<code>\@glsuseriv</code>	147, 490	31, 32, 184, 186, 420, 514, 515, 517
<code>\@glsuseriv@</code>	147, 490	<code>\@no@post@desc</code>	340, 427
<code>\@glsuserv</code>	148, 491	<code>\@nopostdesc</code>	202, 427, 525
<code>\@glsuserv@</code>	148, 491	<code>\@onelevel@sanitize</code>	7,
<code>\@glsuservi</code>	149, 491		23, 33, 54, 80, 96, 122, 168, 169, 173,
<code>\@glsuservi@</code>	149, 491		197, 199, 211, 336, 337, 405, 414, 433,
<code>\@glswidestname</code>	329–331, 347		448, 460, 476, 502, 503, 505, 522, 523, 530
<code>\@glswritefiles</code>	34, 421, 518	<code>\@onlypreamble</code> .	68, 78, 88, 102, 105, 184,
<code>\@glsextr@doacccsupp</code>	353		188, 442, 448, 454, 455, 464, 465, 514, 517
<code>\@glsextr@record</code>	275	<code>\@onlypremakeg</code>	44, 45, 51, 52, 56, 68,
<code>\@gobble</code>	5,		175, 425, 427, 428, 431, 432, 435, 442, 507
	6, 14–16, 31–33, 80, 97, 122, 168, 172,	<code>\@org@glossaryentrynumbers</code>	
	183, 185, 333, 337, 338, 404, 409, 410,	202, 203, 524, 525
	420, 448, 449, 460, 476, 502, 505, 514, 515	<code>\@org@gl@assign@descplural</code>	
<code>\@gobblethree</code>	6, 404	248, 257, 258,
<code>\@idxitem</code>	323		260–263, 395, 398, 399, 554, 561, 563–565

<code>\@org@gl@assign@firstpl</code>	<code>\@tmp</code>
.... 248, 251, 253, 255, 257, 258, 260–	336, 433
263, 395–399, 554–557, 559, 561, 563, 565	
<code>\@org@gl@assign@plural</code>	<code>\@use@ption</code>
. 248, 251, 253, 255, 257, 258, 260–263,	37, 424
395–399, 554, 556, 557, 559, 561, 563–565	<code>\@warn@nomakeglossaries</code> 182, 204, 513, 526
<code>\@org@gl@assign@symbolplural</code>	<code>\@wrglossary@pageformat</code> 192, 519, 520
..... 248, 251, 253, 255, 260–263,	<code>\@wrglossarynumberhook</code> . 192, 195, 519–521
395–397, 399, 554, 556, 557, 559, 563–565	<code>\@write@gl@labelsname</code>
<code>\@org@gl@numberformat</code>	33
165, 501	<code>\@xdy@main@language</code>
<code>\@org@newglossaryentryprehook</code> .. 88, 454 30, 180, 181, 204, 419, 511, 512, 525
<code>\@outputpage</code>	<code>\@xdy@attributelist</code> .. 52, 172, 431, 432, 504
298, 299	<code>\@xdy@attributes</code>
<code>\@p@glossarysection</code> 51, 170, 333, 335, 431, 432, 503
47, 428, 429	<code>\@xdy@counters</code> 50–52, 172, 431, 432, 504
<code>\@pgls</code>	<code>\@xdy@crossrefhook</code>
275	171, 434, 504
<code>\@pgls@</code>	<code>\@xdy@language</code>
275, 276	204, 436, 525, 526
<code>\@pglspl</code>	<code>\@xdy@lettergroups</code> 58, 173, 338, 436, 437, 506
276	<code>\@xdy@locationclassorder</code>
<code>\@pglspl@</code> 56, 171, 337, 435, 504
276	<code>\@xdy@locref</code> .. 51, 173, 333, 337, 431, 432, 505
<code>\@plus</code>	<code>\@xdynumbergrouporder</code> 58, 169, 436, 502, 503
285, 305, 323	<code>\@xdy@requiredstyles</code>
<code>\@print@glossary</code> 57, 170, 335, 435, 436, 503
200, 524, 525	<code>\@xdysortrules</code>
<code>\@print@noidx@glossary</code> 201, 524, 530	56, 173, 338, 435, 506
<code>\@print@gloss@checkexists</code> ... 200–202, 524	<code>\@xdystyle</code>
<code>\@print@gloss@checkexists@allowignored</code>	170, 335, 503
..... 524	<code>\@xdy@useralphabets</code> .. 53, 170, 335, 433, 503
<code>\@print@gloss@checkexists@noignored</code>	<code>\@xdy@userlocationdefs</code>
..... 200, 201, 523, 524 55, 171, 334, 336, 434, 504
<code>\@print@gloss@setsort</code>	<code>\@xdy@userlocationnames</code> 55, 56, 334, 434, 435
..... 184, 186, 202, 514, 515, 524	<code>\@xfor@nextelement</code>
<code>\@print@glossary</code>	198, 523
200, 201, 524	<code>\@</code>
<code>\@roman</code>	95, 122, 128, 168,
54, 335, 433	174, 175, 226, 227, 338, 339, 341–343,
<code>\@secondofthree</code> . 117, 131, 132, 135, 150,	351, 352, 459, 476, 502, 506, 507, 539, 540
152, 154, 156, 376, 473, 483, 485, 492–495	
<code>\@secondoftwo</code>	<code>\{</code> 80, 168, 174, 175, 333, 338, 339, 448, 502, 507
24, 27,	<code>\}</code> 80, 168, 174, 175, 333, 339, 448, 449, 502, 507
28, 42, 80, 81, 130–133, 136, 150, 151,	<code>\~</code>
153, 154, 376–378, 402, 414, 415, 417,	24, 33, 415
418, 426, 448, 449, 482–485, 492, 494, 495	<code>\‘</code>
<code>\@set@glo@numformat</code> 196, 335, 475, 521	24, 415
<code>\@sglsaddkey</code>	<code>\ </code>
83, 450	123, 125, 128, 177, 477–479, 509
<code>\@sglsaddstoragekey</code>	<code>\~</code>
82, 450	24, 415
<code>\@tabacckludge</code>	
24, 415	
<code>\@text@composite@x</code>	
24, 415	
<code>\@thirdofthree</code> .. 117, 133, 135, 151, 153,	
154, 156, 377, 473, 484, 485, 492, 494–496	
<code>\@this@attr</code>	
172, 504, 505	
<code>\@this@childlabel</code>	
205, 526, 527	
<code>\@this@counter</code>	
52, 432	
<code>\@this@ctr</code>	
172, 504, 505	
<code>\@this@key</code>	
81, 449	
<code>\@this@label</code>	
205, 526	
<code>\@this@cs</code>	
39, 425	

A

<code>\a</code>	24, 415
<code>\AA</code>	25, 415
<code>\aa</code>	25, 415
<code>\Ac</code>	553
<code>\ac</code>	553
<code>accsupp package</code>	353, 354
<code>\accsuppglossaryentryfield</code>	353
<code>\accsuppglossarysubentryfield</code>	354
<code>\Acf</code>	553
<code>\acf</code>	553

<code>\Acfp</code>	553	<code>\acronymtype</code>	18, 233, 235, 247, 249–253, 255–258, 260–263, 265, 395– 398, 411, 543, 544, 553–561, 563–565, 567
<code>\acfp</code>	553	<code>\acrpluralsuffix</code>	233, 236–238, 242–245, 248–250, 252, 255–258, 260– 262, 265, 386, 391–393, 395–399, 541, 543, 545–547, 550–552, 554–557, 559–567
<code>\Acl</code>	553	<code>\ACRshort</code>	492
<code>\acl</code>	553	<code>\Acrshort</code>	246, 492, 553
<code>\Aclp</code>	553	<code>\acrshort</code>	246, 491, 553
<code>\aclp</code>	553	<code>\ACRshorttpl</code>	493
<code>\Acp</code>	553	<code>\Acrshorttpl</code>	246, 493, 553
<code>\acp</code>	553	<code>\acrshorttpl</code>	246, 493, 553
<code>\acrfootnote</code>	250, 257, 554, 555, 560	<code>\Acs</code>	553
<code>\ACRfull</code>	542	<code>\acs</code>	553
<code>\Acrfull</code>	246, 541, 553	<code>\Acsp</code>	553
<code>\acrfull</code>	246, 541, 553	<code>\acsp</code>	553
<code>\ACRfullfmt</code>	231, 234, 242, 244, 390, 392, 542, 543, 550, 551	<code>\add@accent@</code>	24, 414
<code>\Acrfullfmt</code>	230, 234, 242, 244, 390, 392, 542, 543, 550, 551	<code>\addcontentsline</code>	49, 430
<code>\acrfullfmt</code>	230, 234, 242, 243, 390, 392, 541, 543, 550, 551	<code>\addglossarytocaptions</code>	42, 426
<code>\acrfullformat</code>	164, 230, 247, 248, 264, 500, 541, 553, 566	<code>\addtolength</code>	331, 347
<code>\ACRfullpl</code>	542	<code>\advance</code>	15, 16, 91, 120, 298, 409, 410, 456, 475
<code>\Acrfullpl</code>	247, 542, 553	<code>\AE</code>	24, 415
<code>\acrfullpl</code>	246, 542, 553	<code>\ae</code>	24, 415
<code>\ACRfullplfmt</code>	232, 234, 242, 244, 390, 392, 542, 544, 550, 551	<code>\altnewglossary</code>	442
<code>\Acrfullplfmt</code>	232, 234, 242, 244, 390, 392, 542, 544, 550, 551	<code>amsgen package</code>	<u>4</u> , <u>116</u>
<code>\acrfullplfmt</code>	231, 234, 242, 244, 390, 392, 542, 543, 550, 551	<code>amsmath package</code>	<u>97</u>
<code>\acrlinkfootnote</code>	249, 554	<code>\andname</code>	199, 425, 523
<code>\acrlinkfullformat</code>	230–232, 541, 542	<code>\AnyTrackedLanguages</code>	42, 402, 426
<code>\ACRlong</code>	494	<code>\appto</code>	20, 25, 73, 82, 83, 273, 357, 412, 415, 445, 450
<code>\Acrlong</code>	246, 494, 553	<code>array package</code>	<u>295</u> , <u>299</u> , <u>317</u>
<code>\acrlong</code>	246, 494, 553	<code>article class</code>	<u>197</u>
<code>\ACRlongpl</code>	496	<code>\AtBeginDocument</code>	18, 58, 79, 97, 167, 185, 411, 436, 448, 460, 502, 515
<code>\Acrlongpl</code>	246, 495, 553	<code>\AtEndDocument</code>	32, 34, 79, 102, 184, 189, 203, 204, 281, 421, 448, 464, 514, 517, 525, 526
<code>\acrlongpl</code>	246, 495, 553		
<code>\acrnameformat</code>	255, 396, 543, 559		
<code>\acrno linkfootnote</code>	555		
<code>\acronymentry</code>	233, 236, 237, 239, 240, 242–245, 385–388, 391–394, 543–553		
<code>\acronymfont</code>	112, 150–153, 158, 164, 232, 234, 236–240, 242–245, 250, 252, 254, 256–261, 373, 374, 376–378, 386– 388, 390–394, 396–398, 470, 492–494, 497, 500, 543–553, 555, 556, 558, 560–564		
<code>\acronymname</code>	18, 43, 411, 425, 427		
<code>\acronymsort</code>	233, 236, 237, 239, 240, 242, 243, 245, 386–388, 391–394, 543–553		

B

<code>\b</code>	24, 415
<code>babel package</code>	<u>27</u> , <u>40</u> , <u>42</u> , <u>57</u>
<code>\begin</code>	172, 210, 285, 289–294, 297–323, 337, 505, 530
<code>\BeginAccSupp</code>	354
<code>\beginingroup</code>	5, 7, 33, 190, 194, 227, 403, 405, 518, 520, 540
<code>\bfseries</code>	290–293, 295, 296, 300–302, 304, 312–317, 319–323

<code>\bgroup</code>	24, 88, 128, 165, 202, 205, 414, 454, 500, 524, 526	
<code>bib2gls</code>	24, 62, 194	
<code>booktabs</code> package	294–297	
<code>\boolean</code>	264, 566	
<code>\boolfalse</code>	34, 421, 422	
<code>\booltrue</code>	34, 422	
<code>\bottomrule</code>	295, 296	
<code>\box</code>	298	
C		
<code>\c</code>	24, 415	
<code>\c@equation</code>	120, 475	
<code>\c@glossaryentry</code>	12, 408	
<code>\c@glossarysubentry</code>	13, 408	
<code>\c@page</code>	192, 195, 519–521	
<code>\catcode</code>	33, 79, 128, 448	
<code>\cGls</code>	103, 464	
<code>\cgl</code>	103, 464	
<code>\cGlsformat</code>	101, 463, 464	
<code>\cglformat</code>	101, 463, 464	
<code>\cGlspl</code>	104, 465	
<code>\cglsp</code>	104, 465	
<code>\cGlsplformat</code>	102, 463, 465	
<code>\cglspformat</code>	101, 463, 465	
<code>\char</code>	223, 538	
<code>classicthesis</code> package	9	
<code>\cleardoublepage</code>	49, 430	
<code>\clearpage</code>	49, 430	
<code>\closeout</code>	33, 79, 173, 175, 179, 189, 421, 448, 506, 507, 511, 518	
<code>\compatglossarystyle</code>	340–352	
<code>\compatibleglossentry</code>	221, 536, 537	
<code>\compatiblesubglossentry</code>	221, 537	
<code>\copy</code>	298, 299	
<code>\count@</code>	211, 531	
<code>\csdef</code>	22, 82–84, 94, 95, 100, 101, 205, 206, 225, 235, 339, 413, 450– 452, 458, 459, 462, 463, 526, 527, 539, 544	
<code>\csedef</code>	102, 192, 464, 519	
<code>\csgdef</code>	46, 66, 69, 101, 102, 199, 214, 428, 441, 442, 463, 464, 523, 533	
<code>\cslet</code>	73, 88, 95, 209, 444, 454, 459, 530	
<code>\csname</code>	14– 17, 24, 38, 40, 42, 43, 48, 51, 52, 54, 55, 57, 59, 62, 67, 68, 75, 76, 82–84, 86, 87, 90–96, 98, 99, 115, 119–121, 132– 136, 150–157, 165, 166, 170–172, 176– 179, 185, 188, 189, 191, 192, 196–198,	
	201–204, 206, 214, 219, 221, 224, 225, 228, 248, 266–274, 281, 282, 329–331, 333–335, 347, 353, 354, 358, 359, 362, 366, 376–378, 400, 401, 409, 410, 415, 424–427, 429, 430, 432–434, 436–438, 441, 442, 446, 450–454, 456–462, 471, 472, 474, 475, 483–486, 492–496, 501, 504, 505, 508–511, 515, 517–519, 521, 522, 524–527, 533, 536–539, 541, 567–569	
<code>\csshow</code>	270, 569	
<code>\csuse</code>	43, 46, 66, 76, 83, 84, 115, 180–182, 206, 208, 210, 211, 213, 215, 224, 235, 274, 340– 352, 427, 428, 441, 446, 447, 451, 452, 472, 511–513, 527, 529–532, 534, 539, 545	
<code>\csxdef</code>	93, 102, 248, 458, 464	
<code>\currentglossary</code>	12, 13, 46, 202, 408, 428, 525	
<code>\currentglssubentry</code>	13, 217, 408, 535	
<code>\CurrentOption</code>	37, 38, 273, 353, 424	
<code>\CurrentTrackedLanguage</code>	42, 402, 403, 426	
<code>\CurrentTrackedTag</code>	42, 402, 403, 426	
<code>\CustomAcronymFields</code>	265, 566, 567	
<code>\CustomNewAcronymDef</code>	265, 566, 567	
D		
<code>\d</code>	24, 415	
<code>datatool</code> package	206	
<code>datatool-base</code> package	4	
<code>\day</code>	170, 174, 335, 338, 503, 506	
<code>\DeclareAcronymList</code>	18, 20, 233, 235, 249, 251, 253, 256, 258, 261, 263, 265, 411, 412, 543, 544, 554, 556, 558, 559, 561, 564, 565, 567	
<code>\DeclareCurrentRelease</code>	4	
<code>\DeclareListParser</code>	183, 514	
<code>\DeclareOption</code>	10, 273, 353, 407	
<code>\DeclareOptionX</code>	10, 407	
<code>\DeclareRelease</code>	4	
<code>\DeclareRobustCommand</code>	43, 259, 427, 562	
<code>\def</code>	9, 10, 12–19, 23–25, 30– 32, 35, 36, 38–40, 43, 44, 47, 50, 53–58, 62, 66–68, 70–74, 77, 86, 88–93, 95, 97, 101–105, 115, 116, 119–129, 131–157, 165, 166, 169, 174–178, 180, 181, 185– 187, 189, 192, 193, 195–198, 201, 202, 205, 209–211, 213, 214, 216, 223–228, 230–233, 248, 249, 251, 253–263, 265, 273, 276–279, 282–284, 309, 310, 329– 332, 334, 335, 338, 340, 347, 348, 353– 358, 376–383, 395–399, 407–412, 414,	

415, 419, 422–425, 427, 428, 431, 433– 436, 438, 441–445, 447, 453–460, 463– 465, 471, 472, 474–496, 501, 503, 506, 508–512, 514–516, 518–524, 526, 529– 534, 538–543, 554, 556–559, 561–565, 567	\do@gl@xdycheckbackslash 122, 476 \do@gl@xdydisablehyperinlist 119, 474 \do@gl@shaschildren 62, 438 \do@write@gl@labelsname 32, 33 doc package 4, 5, 17
\def@gl@xdycheckbackslash 129, 130, 481, 482	\doifglossarynoexistsordo ... 67, 438, 441 \dtl@ifsingle 223, 538 \dtl@insertinto 206, 527 \dtl@sortresult 207, 527, 528 \dtlcompare 207, 528 \dtlicompare 207, 528 \DTLifinlist 69, 119, 443, 474 \DTLifint 223, 538 \dtlletterindexcompare 207, 528 \DTLsubstituteall 122, 476 \dtlwordindexcompare 207, 527 \DUANewAcronymDef 263, 564, 565
\define@boolkey 8, 10, 12, 13, 18, 25, 26, 29, 30, 34, 116, 216, 406– 408, 411, 415, 416, 419, 421, 422, 473, 534	
\define@choicekey 5, 8, 9, 14, 26, 28, 30, 33, 36, 72, 214, 215, 404–406, 409, 416, 417, 419, 421, 422, 444, 533, 534	
\define@key 9, 13, 20, 26, 30, 31, 70–74, 82, 83, 116, 166, 214, 216, 273, 355–357, 407, 408, 412, 417, 419, 443–445, 450, 472, 501, 533, 534	
\DefineAcronymSynonyms .. 38, 247, 424, 553	
\delimN 172, 183, 213, 226, 337, 428, 505, 514, 532, 540	
\delimR 172, 226, 337, 428, 505, 539, 540	
\DescriptionDUANewAcronymDef 253, 557, 558	
\DescriptionFootnoteNewAcronymDef 251, 555, 556	
\descriptionname . 43, 290–293, 295, 296, 300–302, 304, 312–317, 319–323, 425, 427	
\DescriptionNewAcronymDef .. 256, 559, 560	
\DH 25, 415	
\dh 25, 415	
\dimen@ 237, 298, 329, 546	
\disable@keys 38, 424	
\do 29, 37, 39, 51, 52, 59, 80, 81, 122, 165, 170, 172, 182–184, 191, 198, 205, 235, 249, 252, 254, 256, 258, 261, 263, 265, 281, 282, 335, 418, 424, 425, 431, 432, 437, 448, 449, 476, 501, 503, 504, 513, 514, 519, 523, 526, 544, 554, 556, 558, 560, 561, 564, 566, 567	
\do@gl@storeentry . 14–16, 94, 409, 410, 458	
\do@gl@link@checkfirsthyper 118, 119, 131–137, 150– 156, 376–378, 473, 474, 483–486, 492–496	
	E \eappto 442, 459, 519 \edef 39, 42, 50–55, 57, 67, 76, 79, 81, 115, 120–129, 168, 174–178, 180, 181, 183, 189, 196, 197, 203, 204, 207, 211, 223, 227, 228, 247, 250, 252, 254, 257, 260, 262, 333, 334, 336, 338, 395– 398, 410, 412, 425, 426, 431–438, 441– 443, 446–449, 452–455, 459, 471, 472, 474–482, 501, 502, 506–512, 514, 515, 518, 520–523, 525–528, 531, 535, 538, 540, 541, 553, 555, 557, 559, 560, 563, 564 \egroup 24, 88, 165, 203, 206, 414, 454, 501, 525, 527 \else 6, 12, 13, 16–19, 21, 23, 25, 26, 32–35, 37–39, 43–45, 47– 59, 73, 75, 91, 92, 95–97, 101, 102, 118– 120, 122–129, 132–137, 158, 159, 168– 170, 173, 175, 176, 178–181, 189, 191– 193, 195–199, 202, 211, 216, 218, 223, 226, 227, 237, 251–253, 255, 256, 258, 259, 261, 262, 264, 266, 281, 286, 289, 291, 292, 295, 296, 298, 300, 302, 303, 311, 313, 315, 318, 320, 322, 325, 326, 328, 330, 331, 334–340, 345–348, 363, 405, 407, 408, 410–414, 416, 421, 422, 424, 425, 427–437, 444, 446, 456, 457, 459, 460, 463, 473–486, 497, 502, 503, 505–512, 518–524, 531, 534, 535, 538– 540, 546, 556, 557, 560, 562, 564, 566, 567 \emph 198, 228, 523, 541

<code>\empty</code>	227, 353, 540	<code>\expandonce</code>	76, 77, 122, 123, 176–178, 192, 207, 219, 221, 233, 247–250, 252, 255, 257, 260, 262, 353, 354, 446, 447, 476, 508–510, 519, 527, 528, 536, 537, 543, 553–555, 557, 559, 561, 563, 565
<code>\encodingdefault</code>	24, 415		
<code>\end</code>	172, 210, 285, 289–294, 297–323, 337, 505, 530		
<code>\end@doifinlist</code>	50, 431		
<code>\end@getprefix</code>	197, 522		
<code>\end@glsgls@islistofacronyms</code>	19, 412		
<code>\EndAccSupp</code>	354		
<code>\endcsname</code>	14–17, 24, 38, 40, 42, 43, 48, 51, 52, 54, 55, 57, 59, 62, 67, 68, 75, 76, 82–84, 86, 87, 90–96, 98, 99, 115, 119–121, 132–136, 150–157, 165, 166, 170–172, 176–179, 185, 188, 189, 191, 192, 196–198, 201–204, 206, 214, 219, 221, 224, 225, 228, 248, 266–274, 281, 282, 329–331, 333–335, 347, 353, 354, 358, 359, 362, 366, 376–378, 400, 401, 409, 410, 415, 424–427, 429, 430, 432–434, 436–438, 441, 442, 446, 450–454, 456–462, 471, 472, 474, 475, 483–486, 492–496, 501, 504, 505, 508–511, 515, 517–519, 521, 522, 524–527, 533, 536–539, 541, 567–569		
<code>\endfoot</code>	289–291, 293, 295, 296, 300–302, 304		
<code>\endgroup</code>	5, 7, 33, 191, 195, 227, 404, 405, 519, 521, 540		
<code>\endhead</code>	289–291, 293, 295, 296, 300–302, 304		
<code>\endtheglossary</code>	5, 403, 404		
<code>\entryname</code>	43, 290–293, 295, 296, 300–302, 304, 312–317, 319–323, 425, 427		
<code>\equal</code>	26, 39, 49, 120, 183, 223, 281, 417, 425, 430, 474, 513, 538		
<code>equation (counter)</code>	120, 121		
<code>etoolbox package</code>	4		
<code>\expandafter</code>	7, 14–17, 23, 24, 38, 39, 42, 51, 52, 54–59, 62, 67–69, 75, 76, 80–84, 86, 87, 90–94, 96–99, 115, 119–129, 158, 159, 165, 168, 172, 176, 177, 179, 189, 191, 193, 195, 196, 199, 202, 206, 211, 212, 219, 221, 225, 227, 228, 248, 250, 257, 266–271, 273, 274, 281, 282, 329, 333–335, 337, 338, 353, 354, 358, 359, 362, 363, 394, 400, 401, 405, 409, 410, 414, 415, 424–426, 432–438, 441–443, 446–454, 456–462, 471, 472, 474–481, 497, 501, 502, 505, 508–510, 518–521, 523, 524, 527, 531, 536, 537, 539–541, 555, 560, 567–569		
		F	
<code>\fi</code>	5, 6, 8, 9, 12–19, 21, 23, 25, 26, 28, 33–39, 43–45, 47–59, 68, 73, 75, 90–93, 95–97, 101, 102, 118–129, 132–137, 159, 167–170, 173, 175, 176, 178, 179, 181–184, 189–193, 195–199, 202, 204, 211, 215–218, 223, 224, 226, 227, 237, 247–249, 251–253, 256, 258, 259, 261–266, 272, 281, 286, 289, 291, 292, 295, 296, 298–300, 302, 303, 311, 313, 315, 318, 320, 322, 325, 326, 328–331, 333–337, 339, 340, 345–348, 353, 363, 404–414, 416–418, 421–425, 427–437, 442, 444, 446, 456–460, 463, 473–486, 497, 502, 503, 505, 507, 508, 510, 512–514, 518–526, 531, 533–536, 538–540, 546, 553, 554, 556–562, 564–567, 570		
file types			
<code>.aux</code>	203		
<code>.glo</code>	95		
<code>.ist</code>	167, 168, 178, 179		
<code>.toc</code>	49		
<code>.xdy</code>	44		
<code>glo</code>	271		
<code>\findrootlanguage</code>	436		
<code>\firstacronymfont</code>	114, 236–238, 243, 250, 254, 257, 260, 375, 376, 385, 386, 391, 392, 471, 543, 545, 546, 551, 555, 558, 560, 562, 563		
<code>\fmtversion</code>	25, 415		
<code>\footnote</code>	243, 249, 391, 392, 551, 554, 555		
<code>\FootnoteNewAcronymDef</code>	258, 560, 561		
<code>\footnotesize</code>	7, 405		
<code>\forallacronyms</code>	437		
<code>\forallglossaries</code>	59, 189, 201, 329, 437, 518, 524		
<code>\forallglsentries</code>	32, 99, 100, 102, 167, 421, 437, 462, 464, 502		
<code>\ForEachTrackedDialect</code>	42, 402, 403, 426		
<code>\forglentries</code>	59, 62, 95, 209, 329, 437, 438, 459, 529		
<code>\forlistcsloop</code>	205, 210, 526, 530		
<code>\forlistloop</code>	187, 213, 516, 532		

G

- garamondx package [229](#)
- \gdef ... [16](#), [33](#), [52](#), [67](#), [86](#), [91](#), [92](#), [190](#), [216](#),
[281](#), [410](#), [432](#), [441](#), [453](#), [456](#), [457](#), [518](#), [534](#)
- \Genacrfullformat ... [113](#), [234](#), [236](#), [237](#),
[243](#), [375](#), [385](#), [386](#), [391](#), [471](#), [543–546](#), [551](#)
- \genacrfullformat [113](#), [114](#), [234](#), [236](#), [237](#),
[243](#), [375](#), [385](#), [386](#), [391](#), [471](#), [543–546](#), [551](#)
- \GenericAcronymFields
.... [233](#), [236](#), [237](#), [239](#), [240](#), [242](#), [243](#),
[245](#), [385–388](#), [390](#), [391](#), [393](#), [394](#), [543–553](#)
- \Genplacrfullformat
..... [113](#), [234](#), [236](#), [237](#), [243](#),
[375](#), [385](#), [386](#), [391](#), [470](#), [471](#), [544–546](#), [551](#)
- \genplacrfullformat
.... [113](#), [114](#), [234](#), [236](#), [237](#), [243](#), [374](#),
[375](#), [385](#), [386](#), [391](#), [470](#), [471](#), [544–546](#), [551](#)
- \glo@desc [340](#)
- \glo@do@compare [207](#), [527](#), [528](#)
- \glo@grabfirst [212](#), [530](#), [531](#)
- \glo@label [62](#), [95](#), [438](#), [459](#)
- \glo@list [95](#), [459](#)
- \glo@name [219](#), [536](#)
- \glo@parent [62](#), [438](#)
- \glo@type [95](#), [459](#)
- \glo@value [80](#), [448](#)
- \global [15–](#)
[17](#), [75](#), [78](#), [88](#), [93](#), [98](#), [99](#), [191](#), [203](#), [211](#),
[212](#), [216](#), [298](#), [299](#), [409](#), [410](#), [446](#), [448](#),
[454](#), [458](#), [461](#), [462](#), [518](#), [525](#), [531](#), [532](#), [534](#)
- \glolinkprefix .. [120](#), [166](#), [219](#), [474](#), [501](#), [536](#)
- \glosortentrieswarning .. [21](#), [205](#), [413](#), [526](#)
- glossaries package
[35](#), [37](#), [57](#), [58](#), [169](#), [266](#), [273](#), [285](#), [333](#), [353](#)
- glossaries-accsupp package [95](#), [353](#), [354](#)
- glossaries-extra package
..... [79](#), [171](#), [179](#), [201](#), [275](#), [353](#), [362](#)
- \GlossariesWarning [5](#), [6](#), [8](#), [21](#),
[22](#), [25](#), [26](#), [31](#), [32](#), [35](#), [36](#), [46](#), [50](#), [61](#), [65](#),
[72](#), [75](#), [78–80](#), [103](#), [104](#), [115](#), [117](#), [164](#),
[180–182](#), [185–188](#), [191](#), [197](#), [200](#), [201](#),
[221](#), [224](#), [263](#), [333](#), [353](#), [404](#), [406](#), [412](#),
[413](#), [415–417](#), [420](#), [422](#), [428](#), [430](#), [438](#),
[440](#), [444](#), [446](#), [464](#), [465](#), [471–473](#), [500](#),
[511–513](#), [515–517](#), [519](#), [522–524](#), [537](#), [539](#)
- \GlossariesWarningNoLine
..... [5](#), [6](#), [21](#), [184](#), [185](#),
[189](#), [204](#), [281](#), [404](#), [413](#), [514](#), [515](#), [518](#), [526](#)
- \glossary [34–36](#), [334](#), [335](#), [422](#), [423](#)
- glossary package [1](#), [35](#), [228](#)
- glossary styles:
 - altlist [286](#), [287](#), [341](#)
 - altlistgroup [287](#), [341](#)
 - altlisthypergroup [287](#), [341](#)
 - altlong4col [293](#), [294](#), [302](#), [343](#)
 - altlong4col-booktabs [296](#), [298](#)
 - altlong4colborder [294](#), [343](#)
 - altlong4colheader [294](#), [296](#), [343](#)
 - altlong4colheaderborder [294](#), [343](#)
 - altlongragged4col ... [297](#), [302–304](#), [344](#)
 - altlongragged4col-booktabs [297](#)
 - altlongragged4colborder [304](#), [344](#)
 - altlongragged4colheader [303](#), [344](#)
 - altlongragged4colheaderborder [304](#), [345](#)
 - altsuper4col [316](#), [321](#), [352](#)
 - altsuper4colborder [316](#), [352](#)
 - altsuper4colheader [316](#), [352](#)
 - altsuper4colheaderborder ... [316](#), [352](#)
 - altsuperragged4col [321](#), [322](#), [350](#)
 - altsuperragged4colborder ... [322](#), [350](#)
 - altsuperragged4colheader ... [322](#), [350](#)
 - altsuperragged4colheaderborder .
..... [322](#), [350](#)
 - alttree [309](#), [324](#), [329](#), [331](#), [347](#)
 - alttreegroup [332](#), [348](#)
 - alttreehypergroup [332](#), [348](#)
 - index [9](#), [305](#), [323–326](#), [345](#)
 - indexgroup [325](#), [345](#)
 - indexhypergroup [325](#), [345](#)
 - inline [340](#)
 - list [9](#), [11](#), [285–287](#), [340](#)
 - listdotted [287](#), [288](#), [341](#)
 - listgroup [286](#), [340](#)
 - listhypergroup [286](#), [341](#)
 - long [289](#), [290](#), [295](#), [299](#), [341](#), [343](#)
 - long-booktabs [295](#), [297](#)
 - long3col [290](#), [291](#), [295](#), [342](#)
 - long3col-booktabs [295](#), [297](#)
 - long3colborder [291](#), [342](#)
 - long3colheader [291](#), [295](#), [342](#)
 - long3colheaderborder [291](#), [342](#)
 - long4col [292](#), [293](#), [296](#), [342](#)
 - long4col-booktabs [296](#)
 - long4colborder [293](#), [343](#)
 - long4colheader [292](#), [296](#), [343](#)
 - long4colheaderborder [293](#), [343](#)
 - longborder [289](#), [342](#)

longheader	289, 295, 342	treenonamegroup	328, 347
longheaderborder	290, 342	treenonamehypergroup	328, 347
longragged	297, 299–301	glossary-hypernav package	168
longragged-booktabs	297	glossary-list package	9, 11, 285
longragged3col	297, 301, 302, 344	glossary-long package	11, 288, 302, 310, 311
longragged3col-booktabs	297	glossary-longragged package	299
longragged3colborder	302, 344	glossary-mcols package	304
longragged3colheader	302, 344	glossary-super package	11, 288, 310, 317, 321
longragged3colheaderborder	302, 344	glossary-superragged package	317
longraggedborder	300, 343	glossary-tree package	11, 323
longraggedheader	300, 344	\glossaryentry	196, 198, 335, 521, 522
longraggedheaderborder	301, 344	glossaryentry (counter)	12, 13, 217, 218
mcolalmtree	309, 349	\glossaryentryfield	219, 340–347, 349–352, 385, 536, 537
mcolalmtreegroup	309, 349	\glossaryentrynumbers	10, 172, 202, 203, 212, 216, 337, 407, 505, 524, 525, 531, 532, 534
mcolalmtreehypergroup	309, 310, 349	\glossaryheader	172, 210, 283, 285–287, 289–293, 295, 296, 299–305, 307–309, 311, 312, 314, 318, 319, 321, 324–329, 332, 337, 505, 530, 536
mcolindex	305, 348	\glossarymark	47, 429
mcolindexgroup	305, 348	\glossaryname	17, 42, 43, 410, 425–427
mcolindexhypergroup	305, 306, 348	\glossarypostamble	172, 210, 337, 428, 505, 530
mcoltree	306, 348	\glossarypreamble	171, 210, 337, 428, 504, 530
mcoltreegroup	348	\glossarysection	171, 210, 337, 428, 504, 530
mcoltreehypergroup	307, 348	\glossarystyle	538, 539
mcoltreenoname	308, 349	glossarysubentry (counter)	13, 217, 218
mcoltreenonamegroup	308, 349	\glossarysubentryfield	220, 340–347, 349–352, 385, 537
mcoltreenonamehypergroup	308, 349	\glossarytitle	171, 201, 202, 210, 214, 337, 504, 524, 525, 530, 533
sublistdotted	341	\glossarytoctitle	9, 17, 18, 36, 37, 40, 43, 47, 171, 201, 210, 214, 215, 337, 406, 410, 411, 423, 425, 427, 429, 504, 524, 530, 533
super	311, 312, 319, 351	\glossentry	95, 203, 212, 221, 283, 285–290, 292, 300, 301, 303, 311, 313, 314, 318, 319, 321, 324, 326, 327, 330, 459, 525, 532, 537
super3col	312–314, 351	\Glossentrydesc	384, 536
super3colborder	313, 351	\glossentrydesc	283–290, 292, 300, 301, 303, 311, 313, 314, 318–321, 324–326, 328, 330, 331, 384, 536
super3colheader	313, 351	\Glossentryname	536
super3colheaderborder	314, 351	\glossentryname	283–290, 292, 300, 301, 303, 311, 313, 314, 318, 319, 321, 324–327, 330, 331, 384, 536
super4col	314–316, 352		
super4colborder	315, 352		
super4colheader	315, 352		
super4colheaderborder	315, 352		
superborder	311, 351		
superheader	312, 351		
superheaderborder	312, 351		
superragged	317, 319, 349		
superragged3col	319–321, 350		
superragged3colborder	320, 350		
superragged3colheader	320, 350		
superragged3colheaderborder	321, 350		
superraggedborder	318, 349		
superraggedheader	319, 349		
superraggedheaderborder	319, 350		
tree	306, 326, 327, 329, 345		
treegroup	307, 327, 346		
treehypergroup	327, 346		
treenoname	307, 324, 327, 328, 346		

<code>\Glossentrysymbol</code>	384, 537	<code>\gls@dosubst</code>	122, 476
<code>\glossentrysymbol</code> ...	283, 284, 292, 303, 314, 321, 324–326, 328, 330, 331, 384, 537	<code>\gls@dotocitle</code>	202, 214, 524, 525, 533
<code>\GLS</code>	484	<code>\gls@end@sanitizesort</code>	23, 24, 414
<code>\Gls</code>	103, 228, 247, 464, 483, 541, 553	<code>\gls@endcheck</code>	76, 77, 447
<code>\gls</code>	35, 103, 185, 218, 228, 247, 422, 464, 483, 515, 535, 541, 553	<code>\gls@glossary</code> 35, 36, 196–198, 422, 423, 518, 521, 522
<code>\gls@accessibility</code>	355	<code>\gls@glsdefs@addlabel</code>	80
<code>\gls@accsupp@engine</code>	354	<code>\gls@glsdefs@labellist</code>	78, 79
<code>\gls@Alphpage</code>	191, 195, 519, 521	<code>\gls@gobbleopt</code>	68, 441, 442
<code>\gls@alphpage</code>	191, 195, 519, 521	<code>\gls@grplabel</code>	280
<code>\gls@arabicpage</code>	191, 195, 519, 520	<code>\gls@hypergroup prerun</code>	281
<code>\gls@assign@desc</code>	88, 93, 454, 458	<code>\gls@ifnotmeasuring</code>	97, 98, 460, 461
<code>\gls@assign@descplural</code> .	248, 257, 258, 260–263, 395, 398, 399, 554, 561, 563–565	<code>\gls@inlinepostchild</code>	283, 284, 340
<code>\gls@assign@field</code>	77, 82, 83, 88, 90, 92, 93, 273, 274, 447, 450, 451, 454, 455, 457, 458	<code>\gls@inlinesep</code>	283, 340
<code>\gls@assign@firstpl</code> 248, 251, 253, 255, 257, 258, 260– 263, 395–399, 554–557, 559, 561, 563, 565	<code>\gls@inlinesubsep</code>	283, 284, 340
<code>\gls@assign@plural</code> 248, 251, 253, 255, 257, 258, 260–263, 395–399, 554, 556, 557, 559, 561, 563–565	<code>\gls@islistofacronyms</code>	19, 412
<code>\gls@assign@symbolplural</code> 248, 251, 253, 255, 260–263, 395–397, 399, 554, 556, 557, 559, 563–565	<code>\gls@istfilebase</code> 43, 44, 180, 181, 427, 511, 512
<code>\gls@automake@nr</code>	33, 182, 421, 513	<code>\gls@label</code>	228, 541
<code>\gls@automake@val</code>	33, 421	<code>\gls@level</code>	91, 92, 211, 212, 446, 456, 457, 531
<code>\gls@begindocdefs</code>	79, 448	<code>\gls@noidxglossary</code>	185, 515, 517
<code>\gls@checkisacronymlist</code>	118, 412, 473	<code>\gls@nonumberlist@nr</code>	72, 444
<code>\gls@checkseeallowed</code> 72, 183, 185, 444, 447, 514, 515	<code>\gls@nonumberlist@val</code>	72, 444
<code>\gls@checkseeallowed@preambleonly</code> 444, 447	<code>\gls@nosetquote</code> 89, 174, 175, 178, 455, 506, 507, 510
<code>\gls@codepage</code> 58, 180, 181, 204, 419, 436, 511, 512, 526	<code>\gls@number</code>	195, 520
<code>\gls@debug@nr</code>	5, 36, 404, 422	<code>\gls@numberedsection@nr</code>	9, 214, 215, 406, 533
<code>\gls@debug@val</code>	5, 36, 404, 422	<code>\gls@numberedsection@val</code> .	9, 214, 406, 533
<code>\gls@defdocnewglossaryentry</code> 79, 100, 447, 448, 462	<code>\gls@numberpage</code>	191, 195, 519, 520
<code>\gls@defglossaryentry</code> 77–80, 88, 447, 448, 454, 455	<code>\gls@org@glossaryentryfield</code> ...	203, 525
<code>\gls@disablepagerefexpansion</code> 191, 195, 519, 521	<code>\gls@org@glossarysubentryfield</code>	203, 525
<code>\gls@do@addxdyattribute</code>	52, 432	<code>\gls@org@insert</code> 254, 256, 257, 259, 558, 560, 562
<code>\gls@docdef@addlabel</code>	79	<code>\gls@orgAlph</code>	195, 520, 521
<code>\gls@docdef@labellist</code>	78	<code>\gls@orgalph</code>	195, 520, 521
<code>\gls@docclearpage</code>	49, 430	<code>\gls@orgarabic</code>	195, 520
<code>\gls@doentryfmt</code>	441	<code>\gls@orgnumber</code>	195, 520
		<code>\gls@orgRoman</code>	195, 520, 521
		<code>\gls@orgromannumeral</code>	195, 520
		<code>\gls@orgthe</code>	195, 520
		<code>\gls@original@glossary</code>	36, 422, 423
		<code>\gls@original@makeglossary</code> ..	36, 422, 423
		<code>\gls@protected@pagefmts</code> 122, 191, 192, 476, 519
		<code>\gls@Romanpage</code>	191, 195, 519, 521
		<code>\gls@romanpage</code>	191, 195, 519, 520
		<code>\gls@save@numberlist</code>	10, 407, 523
		<code>\gls@seenoindex@nr</code>	8, 405
		<code>\gls@seenoindex@val</code>	8, 405

<code>\gls@set@xr@key</code>	72, 444	<code>\glsautoprefix</code>	9, 215, 406, 533
<code>\gls@suffixF</code>		<code>\glsbackslash</code>	502
.....	45, 173, 175, 337, 339, 428, 505, 507	<code>\glscapscase</code>	106,
<code>\gls@suffixFF</code>			107, 110–113, 131–136, 150–156, 241,
.....	45, 173, 175, 337, 339, 428, 505, 507		254, 260, 366, 368, 371–378, 389, 465,
<code>\gls@text</code>	114, 471		467–471, 483–485, 492–496, 549, 558, 562
<code>\gls@the</code>	195, 520	<code>\glscategory</code>	362
<code>\gls@thissty</code>	29, 418	<code>\glsclearpage</code>	48, 429, 430
<code>\gls@tmp</code>	189, 259, 518, 562	<code>\glsclosebrace</code>	
<code>\gls@tmplen</code>	130, 329–331, 347, 348, 482		. 55, 172, 173, 337, 338, 434, 502, 505, 506
<code>\gls@tr@set@acronym@toctitle</code> ...	18, 411	<code>\glscompositor</code>	
<code>\gls@tr@set@main@toctitle</code>	17, 410	...	44, 45, 55, 175, 336, 339, 428, 434, 507
<code>\gls@tr@set@numbers@toctitle</code> ...	37, 423	<code>\glscounter</code>	20, 39, 50,
<code>\gls@tr@set@symbols@toctitle</code> ...	36, 423		68, 92, 120, 333, 412, 425, 431, 441, 457, 474
<code>\gls@translate@nr</code>	28, 418	<code>\glscurrententrylabel</code> ..	199, 203, 523, 525
<code>\gls@translate@val</code>	28, 418	<code>\glscurrentfieldvalue</code>	64, 65, 440
<code>\gls@wrglossary</code>	190, 518	<code>\glscustomtext</code> ..	106, 109, 111, 112, 114,
<code>\gls@xdystring</code>	122, 476		131–136, 150–157, 241, 242, 250, 254,
<code>\gls@xindy@glsnumbersfalse</code>	31, 420		256, 257, 259, 260, 366, 370, 373, 375–
<code>\gls@xindy@glsnumberstrue</code>	30, 419		378, 388, 390, 465, 468, 470, 471, 483–
<code>\gls@xr@key</code>	7, 8, 72, 405, 406, 444		485, 492–496, 549, 555, 558, 560, 562, 563
<code>\glsaccessibility</code>	362, 363	<code>\GlsDeclareNoHyperList</code>	20, 412
<code>\glsacccsupp</code>	362, 363	<code>\glsdefaultshortaccess</code>	394–398
<code>\glsacronymtrue</code>	18, 411	<code>\glsdefaulttype</code>	17, 46, 57, 59, 66,
<code>\glsacrpluralsuffix</code> ..	41, 229, 238, 242,		67, 90, 105, 115, 189, 200, 201, 411, 428,
	243, 245, 249, 425, 541, 547, 550–552, 554		436, 437, 441, 455, 465, 471, 472, 518, 524
<code>\glsacrshortcutsfalse</code>	38, 424	<code>\glsdefmain</code>	17, 69, 410, 411, 442
<code>\glsacrshortcutstrue</code>	38, 424	<code>\glsdefs@newdocentry</code>	80
<code>\glsacspace</code>	236, 237, 239, 545, 546, 548	<code>\GLSdesc</code>	488
<code>\glsadd</code>	35, 167, 422, 501, 502	<code>\Glsdesc</code>	488
<code>\glsadd options</code>		<code>\glsdesc</code>	488
counter	166	<code>\GLSdescplural</code>	488
format	166, 225	<code>\Glsdescplural</code>	488
<code>\glsaddall</code>	502	<code>\glsdescplural</code>	488
<code>\glsaddall options</code>		<code>\glsdescriptionaccessdisplay</code>	
types	166, 167		368–370, 380, 384, 385
<code>\glsaddallunused</code>	502	<code>\glsdescriptionpluralaccessdisplay</code>	
<code>\glsaddkey</code>	450		367, 368, 380, 381
<code>\GlsAddLetterGroup</code>	437	<code>\glsdescwidth</code>	289–
<code>\glsaddprotectedpagefmt</code>	519		291, 293, 294, 297–304, 311–314, 316–323
<code>\GlsAddSortRule</code>	435	<code>\glsdetoklabel</code> ...	32, 60–65, 73, 80, 85–
<code>\glsaddstoragekey</code>	450		89, 95, 98–102, 119, 157, 158, 165, 166,
<code>\GlsAddXdyAlphabet</code>	433		185–187, 193, 196, 203, 206, 207, 211,
<code>\GlsAddXdyAttribute</code> ..	51–53, 333, 334, 431–433		212, 214, 217–219, 266–270, 274, 275,
<code>\GlsAddXdyCounters</code> ..	51, 52, 68, 431, 432, 442		329, 353, 354, 363–366, 400, 401, 437–
<code>\GlsAddXdyLocation</code>	434, 435		440, 444, 448, 452–455, 459, 461–464,
<code>\GlsAddXdyStyle</code>	435, 436		474, 496, 497, 501, 515, 516, 520, 521,
<code>\glsautomakefalse</code>	33, 34, 182, 421, 513		525, 527, 528, 531, 533, 535, 536, 567–569
<code>\glsautomaketrue</code>	33, 421	<code>\glsdisablehyper</code>	482

<code>\glsdisp</code>	485	<code>\Glsentryfull</code>	234, 242, 244, 390, 392, 500, 544, 550, 552
<code>\glsdisplay</code>	106, 115, 465, 471, 472	<code>\glsentryfull</code>	234, 242, 244, 390, 392, 500, 544, 550, 551
<code>\glsdisplayfirst</code> ...	106, 115, 465, 471, 472	<code>\Glsentryfullpl</code>	234, 242, 244, 390, 393, 500, 544, 550, 552
<code>\glsdisplaynumberlist</code>	186, 500, 516	<code>\glsentryfullpl</code>	234, 242, 244, 390, 392, 500, 544, 550, 552
<code>\glsdohyperlink</code>	130, 131, 482, 483	<code>\glsentryitem</code> ...	283, 285–290, 292, 300, 301, 303, 311, 313, 314, 318, 319, 321, 324, 326, 327, 330, 340–347, 349–352, 535
<code>\glsdohypertarget</code>	130, 131, 482, 483	<code>\Glsentrylong</code>	104, 154, 158, 164, 236, 241, 242, 375, 378, 385, 389, 390, 464, 494, 497, 500, 545, 549, 550
<code>\glsdoifexists</code>	62–64, 85–87, 97, 98, 131–137, 149–156, 161, 164–166, 186, 187, 276–279, 376–378, 384, 437–439, 452–454, 461, 483–486, 492–496, 498, 500, 501, 516	<code>\glsentrylong</code>	103, 114, 153, 155, 158, 164, 236, 237, 239–245, 257, 375, 377, 378, 385–394, 464, 471, 494, 495, 497, 500, 545–553, 560
<code>\glsdoifexistsordo</code> .	118, 157, 438, 473, 496	<code>\glsentrylongaccess</code>	365
<code>\glsdoifexistsorwarn</code>	214, 219, 220, 438, 532, 536, 537	<code>\Glsentrylongpl</code>	105, 156, 164, 236, 237, 241, 242, 376, 385, 389, 390, 465, 496, 500, 545, 546, 549, 550
<code>\glsdoifnoexists</code>	77, 88, 438, 447, 454	<code>\glsentrylongpl</code>	104, 114, 155, 157, 164, 236, 237, 241–244, 257, 264, 376, 385, 386, 389–393, 465, 471, 495, 496, 500, 545, 546, 549–552, 560, 566
<code>\glsdonohyperlink</code>	120, 130, 474, 482	<code>\glsentrylongpluralaccess</code>	365
<code>\glsdoparenifnotempty</code>	562	<code>\Glsentryname</code> 141, 220, 380, 384, 488, 497, 536	
<code>\glsdosanitizesort</code>	14, 409	<code>\glsentryname</code> 141, 329, 380, 384, 487, 488, 497	
<code>\glsenableentrycount</code>	462, 464	<code>\glsentrynumberlist</code>	165, 186, 500, 515
<code>\glsenablehyper</code>	483	<code>\glsentryparent</code>	499
<code>\glsentryaccess</code>	363	<code>\Glsentryplural</code>	107, 110, 139, 367, 371, 379, 466, 469, 487, 498
<code>\glsentrycounter</code> ...	224, 227, 474, 538, 540	<code>\glsentryplural</code> .	106, 107, 110, 139, 367, 368, 371, 379, 465, 466, 468, 469, 487, 498
<code>\glsentrycounterfalse</code>	13, 408	<code>\glsentrypluralaccess</code>	364
<code>\glsentrycounterlabel</code>	218, 535	<code>\Glsentryprefix</code>	277
<code>\GlsEntryCounterLabelPrefix</code>	217, 218, 534, 535	<code>\glsentryprefix</code>	276, 278
<code>\glsentrycountertrue</code>	13, 408	<code>\Glsentryprefixfirst</code>	277
<code>\glsentrycurrcount</code>	100, 102, 462, 464	<code>\glsentryprefixfirst</code>	276, 279
<code>\Glsentrydesc</code> 142, 220, 380, 384, 488, 497, 536		<code>\Glsentryprefixfirstplural</code>	278
<code>\glsentrydesc</code> ...	108, 109, 141, 142, 220, 368–370, 380, 384, 467, 468, 488, 497, 536	<code>\glsentryprefixfirstplural</code>	276, 279
<code>\glsentrydescaccess</code>	364	<code>\Glsentryprefixplural</code>	278
<code>\Glsentrydescplural</code>	142, 380, 488, 497	<code>\glsentryprefixplural</code>	276, 279
<code>\glsentrydescplural</code>	106, 107, 142, 143, 367, 368, 380, 381, 465, 466, 488, 497	<code>\glsentryprevcount</code>	100–102, 462, 463
<code>\glsentrydescpluralaccess</code>	365	<code>\Glsentryshort</code>	112, 150, 158, 159, 237, 243, 244, 374–376, 386, 391, 392, 470, 492, 497, 500, 546, 551, 552
<code>\Glsentryfirst</code>	104, 108, 111, 138, 369, 372, 379, 464, 467, 469, 486, 498		
<code>\glsentryfirst</code>	103, 108, 109, 111, 138, 139, 369, 370, 372, 373, 379, 464, 467–469, 486, 487, 498		
<code>\glsentryfirstaccess</code>	364		
<code>\Glsentryfirstplural</code> ...	105, 107, 110, 140, 367, 371, 379, 465, 466, 469, 487, 498		
<code>\glsentryfirstplural</code>	104, 106, 107, 110, 140, 367, 368, 371, 372, 379, 465, 466, 468, 469, 487, 498		
<code>\glsentryfirstpluralaccess</code>	364		
<code>\glsentryfmt</code>	67, 69, 441, 442, 465		

<code>\glsentryshort</code>	112, 114, 150, 151, 158, 159, 164, 234, 236, 237, 239, 240, 242–245, 374–377, 385–388, 390– 394, 470, 471, 492, 497, 499, 500, 544–553
<code>\glsentryshortaccess</code>	365
<code>\Glsentryshorttpl</code> 112, 152, 237, 243, 244, 373, 386, 392, 393, 470, 493, 500, 546, 551, 552
<code>\glsentryshorttpl</code>	112, 114, 151, 153, 164, 236, 237, 242–244, 264, 373, 374, 376, 385, 386, 390–393, 470, 471, 493, 494, 500, 545, 546, 550–552, 566
<code>\glsentryshortpluralaccess</code>	365
<code>\glsentrysort</code>	499
<code>\Glsentrysymbol</code> 143, 220, 381, 384, 489, 498, 537
<code>\glsentrysymbol</code>	108, 109, 143, 144, 220, 250, 254, 259, 368–370, 381, 384, 467, 468, 489, 498, 537, 555, 558, 562
<code>\glsentrysymbolaccess</code>	364
<code>\Glsentrysymbolplural</code> ..	144, 381, 489, 498
<code>\glsentrysymbolplural</code> 106, 107, 144, 250, 254, 259, 367, 368, 381, 465, 466, 489, 498, 555, 558, 562
<code>\glsentrysymbolpluralaccess</code>	364
<code>\Glsentrytext</code>	108, 111, 138, 369, 372, 378, 467, 469, 486, 498
<code>\glsentrytext</code>	108, 109, 111, 137, 138, 166, 199, 368, 369, 372, 373, 378, 467–469, 486, 498, 501, 523
<code>\glsentrytextaccess</code>	364
<code>\glsentrytitlecase</code>	498, 499
<code>\glsentrytype</code>	90, 455, 499
<code>\Glsentryuseri</code>	145, 382, 489, 499
<code>\glsentryuseri</code> ..	145, 381, 382, 489, 490, 499
<code>\glsentryuseriaccess</code>	365
<code>\Glsentryuserii</code>	146, 382, 490, 499
<code>\glsentryuserii</code>	145, 146, 382, 490, 499
<code>\glsentryuseriiaccess</code>	365
<code>\Glsentryuseriii</code>	147, 382, 490, 499
<code>\glsentryuseriii</code> ...	146, 147, 382, 490, 499
<code>\glsentryuseriiiaccess</code>	366
<code>\Glsentryuseriv</code>	147, 383, 491, 499
<code>\glsentryuseriv</code> .	147, 148, 383, 490, 491, 499
<code>\glsentryuserivaccess</code>	366
<code>\Glsentryuserv</code>	148, 383, 491, 499
<code>\glsentryuserv</code>	148, 383, 491, 499
<code>\glsentryuservaccess</code>	366
<code>\Glsentryuservi</code>	149, 383, 491, 499
<code>\glsentryuservi</code>	149, 383, 491, 499
<code>\glsentryuserviaccess</code>	366
<code>\glsesclocationsfalse</code>	185, 515
<code>\glsesclocationstrue</code>	10, 407
<code>\glsexpandfields</code>	447
<code>\glsfieldaccsupp</code>	362, 363
<code>\glsfielddef</code>	453
<code>\glsfieldedef</code>	452
<code>\glsfieldfetch</code>	161, 453, 498
<code>\glsfieldgdef</code>	452
<code>\glsfieldxdef</code>	452
<code>\GLSfirst</code>	487
<code>\Glsfirst</code>	486
<code>\glsfirst</code>	486
<code>\glsfirstaccessdisplay</code> 369, 370, 372, 373, 379
<code>\GLSfirstplural</code>	487
<code>\Glsfirstplural</code>	487
<code>\glsfirstplural</code>	487
<code>\glsfirstpluralaccessdisplay</code> 367, 368, 371, 379
<code>\glsfirstpluralacesdisplay</code>	372
<code>\glsngenacfmt</code>	236, 237, 243, 385, 386, 391, 470, 545, 546, 551
<code>\glsngenentryfmt</code> .	236, 237, 242, 243, 247, 250, 252, 254, 256, 257, 259, 262, 264, 385, 386, 389, 391, 468, 545, 546, 549, 551, 553, 555, 556, 558, 560, 562, 564, 566
<code>\glsgetgrouplabel</code>	538
<code>\glsgetgrouptitle</code>	282, 286, 287, 305–310, 325–328, 332, 537
<code>\gls glossarymark</code>	47, 429
<code>\gls groupheading</code> 173, 212, 283, 285–287, 289, 290, 292, 300, 301, 303, 305–312, 314, 318, 319, 321, 324–329, 331, 332, 338, 506, 532, 537
<code>\gls groupskip</code>	172, 173, 212, 284, 286, 289, 291, 292, 295, 296, 300–303, 311, 313, 315, 318, 320, 322, 325, 326, 328, 331, 337, 505, 506, 531, 537
<code>\glshyperfirstfalse</code>	243, 391, 551
<code>\glshyperfirsttrue</code>	29, 419
<code>\glshyperlink</code>	199, 501, 523
<code>\glshypernavsep</code>	282
<code>\glshypernumber</code> ..	46, 227, 228, 428, 539–541
<code>\glsifhyper</code>	473
<code>\glsifhyperon</code>	117, 473, 474
<code>\glsIfListOfAcronyms</code>	19, 20, 412

<code>\glsifplural</code>	<code>\glslocalresetall</code>
.... 106, 109, 112, 113, 131–136, 150–	462
156, 241, 250, 254, 257, 259, 366, 370,	<code>\glslocalunset</code>
373, 374, 376–378, 388, 465, 468, 470, 100, 132–137, 461, 462, 483–486
483–485, 492–496, 549, 555, 558, 560, 562	<code>\glslocalunsetall</code>
<code>\glsifusedtranslatordict</code>	462
417	<code>\glslongaccessdisplay</code> 375, 377, 378, 385–395
<code>\glsifusetranslator</code>	<code>\glslongkey</code>
..... 27, 28, 42, 402, 417, 418, 426, 427	399, 541
<code>\glsignore</code>	<code>\glslongpluralaccessdisplay</code>
502 376, 385, 386, 389–393, 395
<code>\glsindexonlyfirstfalse</code>	<code>\glslongpluralkey</code>
29, 419	399, 541
<code>\glsinlinedescformat</code>	<code>\glslongtok</code>
283, 340	233,
<code>\glsinlinedopostchild</code>	234, 236, 237, 242, 243, 247–253, 255–
283, 340	258, 260–265, 385, 386, 390, 391, 394–
<code>\glsinlineemptydescformat</code>	399, 543–546, 550, 551, 553–561, 563–567
283, 340	<code>\glsLTpenaltycheck</code>
<code>\glsinlinenameformat</code>	298, 299
283, 340	<code>\glsmcols</code>
<code>\glsinlineparentchildseparator</code> 283, 340	305–310
<code>\glsinlinepostchild</code>	<code>\glsmoveentry</code>
283, 340	459
<code>\glsinlineseparator</code>	<code>\GLSname</code>
283, 340	488
<code>\glsinlinesubdescformat</code>	<code>\GLSname</code>
284, 340	487
<code>\glsinlinesubnameformat</code>	<code>\glsname</code>
284, 340	487
<code>\glsinlinesubseparator</code>	<code>\glsnameaccessdisplay</code>
284, 340	380, 384, 385
<code>\glsinsert</code>	<code>\glsnamefont</code>
106–113, 131–136, 150–
156, 241, 242, 250, 254, 256, 257, 259, 219, 220, 353, 354, 384, 536, 537, 539
260, 367–378, 389, 390, 465–471, 483–	<code>\glsnavhyperlink</code>
485, 492–496, 549, 555, 558, 560, 562, 563	282
<code>\glskeylisttok</code>	<code>\glsnavhyperlinkname</code>
233,	280, 281
248–251, 253, 255–258, 260–263, 265,	<code>\glsnavhypertarget</code>
394–399, 543, 554–559, 561, 563–565, 567 286, 287, 306–310, 326–328, 332
<code>\glslabel</code>	<code>\glsnavigation</code>
89, 106–113, 118–120, 286, 287, 305–310, 325, 327, 328, 332
150–156, 236, 237, 241, 243, 250, 254,	<code>\glsnextpages</code> . 10, 73, 202, 407, 444, 525, 534
256, 257, 259, 367–375, 385, 386, 388,	<code>\glsnoexpandfields</code>
389, 391, 455, 465–471, 473, 474, 492–	447
496, 545, 546, 549, 551, 555, 558, 560, 562	<code>\glsnogroupskipfalse</code>
<code>\glslabeltok</code> 233, 247–254, 256–258,	12, 408
260–265, 395–398, 543, 553–561, 563–567	<code>\glsnoidxdisplayloc</code> 187, 188, 516, 517, 532
<code>\glsletentryfield</code>	<code>\glsnoidxdisplaylocclishandler</code>
496 187, 516, 532
<code>\glslink</code>	<code>\glsnoidxloclist</code> ... 186, 212, 516, 531, 532
234, 242–244,	<code>\glsnoidxloclisthandler</code>
249, 390, 392, 473, 543, 544, 550, 551, 554	213, 532
<code>\glslink options</code>	<code>\glsnoidxnumberlistloophandler</code>
counter 187, 516, 517
116, 131, 272	<code>\glsnoidxstripaccents</code>
format	24, 414, 415
116, 131, 225	<code>\glsnomakeindexwarning</code>
hyper	175, 430, 507
116, 118, 119, 131	<code>\glsnonextpages</code>
local	72, 202, 444, 525, 534
116	<code>\glsnopostdotfalse</code>
<code>\glslinkcheckfirsthyperhook</code> ... 118, 474	12, 407
<code>\glslinkpostsetkeys</code>	<code>\glsnoxindywarning</code> ... 45, 51–53, 56–58,
119, 474	168, 169, 428, 430–433, 435, 436, 502, 503
<code>\glslinkvar</code>	<code>\glsnumberformat</code>
117, 473	165, 428, 501
<code>\glslistdottedwidth</code>	<code>\glsnumberlistloop</code>
287, 288, 341	187, 516, 517
<code>\glslistgroupheaderfmt</code>	<code>\glsnumbersgroupname</code>
286, 287
<code>\glslistnavigationitem</code> 37, 43, 223, 423, 425, 427, 538
286, 287	<code>\glsnumlistlastsep</code>
<code>\glslocalreset</code>	165, 187, 501, 516
99, 461, 462	

<code>\glsnumlistparser</code>	165, 183, 501, 514	<code>\glsseeformat</code>	
<code>\glsnumlistsep</code>	165, 186, 501, 516	171, 185, 187, 337, 504, 515–517, 523
<code>\glsopenbrace</code>		<code>\glsseeitem</code>	199, 523
. 55, 172, 173, 337, 338, 434, 502, 505, 506		<code>\glsseeitemformat</code>	199, 523
<code>\glsorder</code>		<code>\glsseelastsep</code>	199, 523
. 30, 180, 181, 183, 208, 419, 511–513, 529		<code>\glsseelist</code>	198, 523
<code>\glsorg@endtheglossary</code>	5, 403, 404	<code>\glsseesep</code>	199, 523
<code>\glsorg@PrintChanges</code>	5, 403, 404	<code>\glsSetAlphaCompositor</code>	428
<code>\glsorg@theglossary</code>	5, 403	<code>\glsSetCompositor</code>	428
<code>\glspagelistwidth</code> 290, 291, 293, 294, 297,		<code>\glssetexpandfield</code>	
298, 301–304, 312–314, 316, 317, 319–323		22, 25–27, 413, 414, 416, 417
<code>\glspar</code>	427	<code>\glssetnoexpandfield</code> 22, 23, 25, 26, 413–416	
<code>\glspatchLToutput</code>	295–297	<code>\GlsSetQuote</code>	89, 174, 455, 506, 507, 510
<code>\glspatchtabularx</code>	460	<code>\glsSetSuffixF</code>	428
<code>\glspenaltygroupskip</code>	295, 296	<code>\glsSetSuffixFF</code>	428
<code>\glsperscentchar</code>		<code>\glssettoctitle</code>	42, 202, 425, 427, 524
..... 80, 172, 174, 448, 449, 502, 505, 506		<code>\GlsSetWriteIstHook</code>	507
<code>\GLSpl</code>	485	<code>\GlsSetXdyCodePage</code>	436
<code>\Glspl</code>	104, 247, 465, 484, 553	<code>\GlsSetXdyFirstLetterAfterDigits</code> ..	502
<code>\glspl</code>	104, 247, 465, 484, 553	<code>\GlsSetXdyLanguage</code>	436
<code>\GLSplural</code>	487	<code>\GlsSetXdyLocationClassOrder</code>	435
<code>\Glsplural</code>	487	<code>\GlsSetXdyMinRangeLength</code>	503
<code>\glsplural</code>	487	<code>\GlsSetXdyNumberGroupOrder</code>	502, 503
<code>\glspluralaccessdisplay</code> 367, 368, 371, 379		<code>\GlsSetXdyStyles</code>	436
<code>\glspluralsuffix</code>	41, 92,	<code>\glsshortaccessdisplay</code>	
236–238, 386, 391–393, 425, 457, 545, 546		159, 374–377, 385–388, 390–395, 497
<code>\glspostdescription</code>	43, 284–	<code>\glsshortaccsupp</code>	363
287, 289, 300, 311, 318, 324–326, 328,		<code>\glsshortkey</code>	399, 541
330, 331, 340–343, 345–349, 351, 407, 427		<code>\glsshortpluralaccessdisplay</code>	
<code>\glspostinline</code>	283	373, 374, 376, 385, 386, 390–393, 395
<code>\glspostlinkhook</code>	118, 132–137,	<code>\glsshortpluralkey</code>	399, 541
150–157, 376–378, 473, 483–486, 492–496		<code>\glsshorttok</code> 233, 247–253, 255–258, 260–	
<code>\glsprefixsep</code>	276–279	265, 394–399, 543, 553–559, 561, 563–567	
<code>\glsprestandardsort</code>	14, 409	<code>\glsshowaccsupp</code>	6, 404, 405
<code>\glsquote</code>	502	<code>\glsshowtarget</code>	7, 405
<code>\glsrefentry</code>	535	<code>\glsshowtargetfont</code>	6, 405
<code>\glsreset</code>	99, 461, 462	<code>\glsshowtargetouter</code>	6, 405
<code>\glsresetall</code>	462	<code>\glsshowtargetsymbols</code>	6, 405
<code>\glsresetentrycounter</code>	217, 535	<code>\glssortnumberfmt</code>	15, 16, 409, 410
<code>\glsresetentrylist</code>		<code>\glsspace</code>	230, 248, 249, 541
..... 172, 210, 216, 337, 505, 530, 534		<code>\glsstepentry</code>	218, 535
<code>\glsresetsubentrycounter</code>		<code>\glsstepsubentry</code>	218, 535, 536
..... 218, 283, 340, 534, 535		<code>\glssubentrycounterfalse</code>	13, 408
<code>\glssanitizesortfalse</code>	26, 416	<code>\glssubentrycounterlabel</code> ...	218, 535, 536
<code>\glssanitizesorttrue</code>	26, 416	<code>\glssubentryitem</code>	
<code>\glssavenumberlistfalse</code>	10, 407	284, 285, 287–290, 292, 300,
<code>\glssavewritesfalse</code>	34, 421	301, 303, 311, 313, 314, 318, 320, 321,	
<code>\glssee</code>	523	325, 326, 328, 330, 340–347, 349–352, 535	
		<code>\GLSsymbol</code>	489

<code>\Glssymbol</code>	489	<code>\GlsUseAcrStyleDefs</code>	
<code>\glssymbol</code>	488	235, 238–240, 243–245, 387, 388, 391, 393, 394, 544, 545, 547–550, 552, 553
<code>\glssymbolaccessdisplay</code>		<code>\GLSuseri</code>	489
.....	368–370, 381, 384, 385	<code>\Glsuseri</code>	489
<code>\GLSsymbolplural</code>	489	<code>\gluseri</code>	489
<code>\Glssymbolplural</code>	489	<code>\gluseriaccessdisplay</code>	381, 382
<code>\glssymbolplural</code>	489	<code>\GLSuserii</code>	490
<code>\glssymbolpluralaccessdisplay</code>		<code>\Glsuserii</code>	490
.....	367, 368, 381	<code>\gluserii</code>	490
<code>\glssymbolsgroupname</code>		<code>\gluseriiaccessdisplay</code>	382
.....	36, 43, 223, 423, 425, 427, 538	<code>\GLSuseriii</code>	490
<code>\glstarget</code>	221, 284–290, 292, 300, 301, 303, 311, 313, 314, 318– 321, 324–328, 330, 331, 340–352, 536, 537	<code>\Glsuseriii</code>	490
<code>\GLStext</code>	486	<code>\gluseriii</code>	490
<code>\Glstext</code>	486	<code>\gluseriiiaccessdisplay</code>	382
<code>\glstext</code>	486	<code>\GLSuseriv</code>	491
<code>\glstextaccessdisplay</code>	368, 369, 372, 373, 378	<code>\Glsuseriv</code>	490
<code>\glstextformat</code>	118, 120, 465, 473, 474	<code>\gluseriv</code>	490
<code>\glstextup</code>	41, 393, 425, 541	<code>\gluserivaccessdisplay</code>	383
<code>\glstildechar</code>	51, 172, 173, 432, 502, 505, 506	<code>\GLSuserv</code>	491
<code>\glstranslatefalse</code>	28, 417, 418	<code>\Glsuserv</code>	491
<code>\glstranslatetrue</code>	28, 29, 418	<code>\gluserv</code>	491
<code>\glstreechildpredesc</code>	325, 326	<code>\gluservaccessdisplay</code>	383
<code>\glstreegroupheaderfmt</code>		<code>\GLSuservi</code>	491
.....	305–310, 325, 327, 328, 332	<code>\Glsuservi</code>	491
<code>\glstreeindent</code> ..	326, 328, 330, 331, 346–348	<code>\gluservi</code>	491
<code>\glstreeitem</code>	305, 306, 324	<code>\gluserviaccessdisplay</code>	383
<code>\glstreenamebox</code>	330, 331	<code>\glswrallowprimitivemodstrue</code> ..	193, 520
<code>\glstreenamefmt</code>	323–327, 329–331	<code>\glswrite</code>	169–175, 182, 189, 335–339, 503–507, 513, 517, 518
<code>\glstreenavigationfmt</code>		<code>\glswritedefhook</code>	80, 449, 454
.....	305–310, 325, 327, 328, 332	<code>\glswriteentry</code>	191, 519
<code>\glstreepredesc</code>	324, 326, 328	<code>\glswritefiles</code>	34, 189, 421, 517
<code>\glstreesubitem</code>	305, 324	<code>\glsxindyfalse</code>	30, 419
<code>\glstreesubsubitem</code>	305, 324	<code>\glsxindytrue</code>	31, 420
<code>\glstype</code>	118, 119, 132–136, 150– 157, 376–378, 473, 474, 483–486, 492–496		
<code>\glsucmarkfalse</code>	12, 408		
<code>\glsucmarktrue</code>	12, 408		
<code>\glunset</code>	97, 99, 101, 102, 132–137, 460–463, 483–486		
<code>\glunsetall</code>	462		
<code>\glsupacrpluralsuffix</code>	238, 244, 252, 256, 258, 261, 425, 547, 552, 556, 560, 562, 564		
<code>\GlsUseAcrEntryDisplayStyle</code>			
.....	235, 238–240, 243–245, 387, 388, 391, 393, 394, 544–550, 552, 553		

H

<code>\H</code>	24, 415
<code>\hangindent</code>	
.....	309, 310, 323, 326–328, 330–332, 345–348
<code>\hbox</code>	97, 287, 288, 341, 460
<code>\hfill</code>	287, 288, 341
<code>\hline</code> ...	289–291, 293, 300–302, 304, 311–323
<code>\hspace</code>	288, 299, 310, 317
<code>\hspace</code>	324
<code>\hss</code>	287, 288, 341
<code>\ht</code>	298
<code>\hyperbf</code>	540
<code>\hyperdef</code>	39, 425

<code>\hyperemph</code>	541	370, 373, 388, 412, 421, 429, 430, 437,
<code>\hyperit</code>	541	439, 440, 442, 465, 468, 470, 513, 514,
<code>\hyperlink</code>	116, 130, 227, 473, 482, 540	530, 531, 543, 544, 549, 555, 558, 560, 562
<code>\hypermd</code>	540	<code>\ifdefequal</code>
<code>hyperref</code> package	197, 200, 225, 226, 272	31,
<code>\hyperrm</code>	540	32, 62–65, 75–77, 81, 91, 95, 212, 363,
<code>\hypersc</code>	541	420, 438–440, 446, 447, 449, 456, 459, 531
<code>\hypersf</code>	540	<code>\ifdefstrequal</code>
<code>\hypersl</code>	541	87, 454
<code>\hypertarget</code>	130, 482	<code>\ifdefstring</code>
<code>\hypertt</code>	540	11, 42, 66, 180,
<code>\hyperup</code>	541	181, 184, 207–209, 211, 213, 354, 407,
		426, 441, 511, 512, 514, 528, 529, 531, 532
		<code>\ifdefvoid</code>
		... 23, 24, 94, 211, 212, 414, 459, 531, 532
		<code>\ifdim</code>
		237, 298, 329, 546
		<code>\iffalse</code>
		93, 99, 458, 461
		<code>\IfFileExists</code>
		11,
		27, 28, 181, 203, 407, 417, 418, 512, 525
		<code>\ifglossaryexists</code>
		46, 57, 62, 179–181, 428, 436–438, 510–512
		<code>\ifgl@sanitize@description</code>
		25, 415
		<code>\ifgl@sanitize@name</code>
		25, 416
		<code>\ifgl@sanitize@symbol</code>
		25, 416
		<code>\ifgl@xindy@gl@numbers</code>
		58, 436
		<code>\ifgl@acr@description</code>
		264, 566
		<code>\ifgl@acr@dua</code>
		... 252, 259, 262, 264, 556, 562, 564, 566
		<code>\ifgl@acr@footnote</code>
		118, 264, 473, 566
		<code>\ifgl@acr@synonym</code>
		18, 411
		<code>\ifgl@acr@shortcuts</code>
		38, 247, 424, 553
		<code>\ifgl@acr@smallcaps</code>
		251–253, 255,
		256, 258, 261, 262, 556, 557, 560, 562, 564
		<code>\ifgl@acr@smaller</code>
		251–253, 255, 256, 258, 556, 557, 560, 562
		<code>\ifgl@automake</code>
		33, 184, 421, 514
		<code>\ifgl@desc@suppressed</code>
		283, 439
		<code>\ifgl@sentry@counter</code>
		... 12, 13, 38, 217, 218, 408, 424, 535
		<code>\ifgl@sentry@exists</code>
		. 61, 78, 80, 88, 91, 437, 438, 448, 454, 456
		<code>\ifgl@esc@locations</code>
		193, 520
		<code>\ifgl@field@dcseq</code>
		454
		<code>\ifgl@field@defeq</code>
		453
		<code>\ifgl@field@deq</code>
		453
		<code>\ifgl@has@children</code>
		283, 340, 438
		<code>\ifgl@has@desc</code>
		283, 439
		<code>\ifgl@has@field</code>
		439
		<code>\ifgl@has@long</code>
		103–105, 158, 236,
		237, 241, 243, 257, 385, 386, 388, 391,
		439, 464, 465, 497, 545, 546, 549, 551, 560
		<code>\ifgl@has@parent</code> .
		206, 211, 329, 438, 527, 531

I

<code>\IeC</code>	24, 414
<code>\if</code>	122, 196, 334, 475, 476, 521
<code>\if@endfor</code>	281
<code>\if@gl@debug</code>	5, 21, 190, 404, 413, 518
<code>\if@gl@docloaded</code> ...	5, 17, 35, 403, 410, 422
<code>\if@gl@is@acronymlist</code>	118, 412, 473
<code>\if@openright</code>	49, 430
<code>\ifbool</code>	18, 29,
	34, 61, 106–108, 411, 419, 422, 437, 466, 467
<code>\ifbool@expr</code>	42, 66, 223, 426, 441, 538
<code>\ifcase</code>	5, 8, 9, 28, 36, 72,
	215, 325, 345, 404–406, 418, 422, 444, 533
<code>\ifcsdef</code>	27, 43, 49, 75,
	76, 83–87, 115, 191, 205, 208–210, 224,
	235, 362, 417, 427, 430, 446, 451–454,
	471, 472, 518, 526, 527, 529, 530, 539, 544
<code>\ifcsempy</code>	63, 275, 439
<code>\ifcsequal</code>	63, 439
<code>\ifcsstrequal</code>	87, 454
<code>\ifcsstring</code>	87, 453
<code>\ifcsundef</code>	8, 15, 31, 39, 42, 45, 47, 49,
	60, 67, 69, 71, 90, 92, 93, 100, 101, 116,
	120, 121, 130, 180, 181, 189, 200, 203,
	206, 214, 218, 223–226, 228, 234, 282,
	339, 363–366, 406, 409, 419, 425, 426,
	428–430, 437, 441–443, 455, 457, 458,
	462, 472, 474, 475, 482, 511, 512, 518,
	523, 525, 527, 533, 536, 538, 539, 541, 544
<code>\ifdef</code>	64,
	65, 73, 79, 80, 97, 116, 157, 161, 186,
	187, 229, 275, 323, 324, 362, 439, 440,
	444, 448, 460, 473, 496, 498, 515, 516, 541
<code>\ifdefempty</code>	20, 32, 48, 59, 63–
	65, 69, 106, 109, 112, 182, 184, 211, 212,
	233, 234, 241, 250, 254, 256, 259, 366,

<code>\ifglshasprefix</code>	276–278	<code>\IfTrackedLanguageFileExists</code>	
<code>\ifglshasprefixfirst</code>	276, 277, 279	42, 402, 403, 426
<code>\ifglshasprefixfirstplural</code> .	276, 278, 279	<code>\iftrue</code>	94, 98, 458, 461
<code>\ifglshasprefixplural</code>	276, 278, 279	<code>\ifundef</code>	12, 13, 67, 78, 90, 169, 174, 182, 183, 408, 441, 448, 455, 503, 506, 513
<code>\ifglshasshort</code>	439	<code>\ifvmode</code>	167, 502
<code>\ifglshassymbol</code>	250, 254, 259, 324–326, 328, 330, 331, 439, 555, 558, 562	<code>\ifvoid</code>	298
<code>\ifglshyperfirst</code>	118, 473	<code>\ifx</code>	12, 14, 16, 19, 32, 37, 39, 50, 52, 54, 55, 58, 59, 90–93, 96, 120, 121, 124–129, 158, 159, 170, 173, 175– 178, 189, 192, 193, 195, 197, 198, 201, 202, 223, 224, 226, 227, 248, 249, 251, 253, 256, 258, 259, 261, 263, 265, 266, 335–337, 339, 340, 345–348, 353, 363, 408–412, 423–425, 431–434, 436, 437, 456–460, 474, 475, 477–482, 497, 503, 505, 507–510, 518, 520–524, 538–540, 554, 556, 558, 559, 561, 562, 564, 565, 567
<code>\ifglsindeonlyfirst</code>	191, 519	<code>\immediate</code> 32, 33, 78, 80, 102, 179, 181, 189, 204, 421, 448, 449, 464, 511, 512, 518, 525, 526
<code>\ifglsnogroupskip</code>	286, 289, 291, 292, 295, 296, 300, 301, 303, 311, 313, 315, 318, 320, 322, 325, 326, 328, 331	<code>\in@</code>	37, 424
<code>\ifglsnonumberlist</code>	216, 534	<code>\indexname</code>	37, 423
<code>\ifglsnopostdot</code>	12, 407	<code>\indexspace</code> .	286, 305–310, 325–328, 331, 332
<code>\ifglslnumberline</code>	49, 430	<code>\input</code>	41, 105, 426, 465
<code>\ifglssanitizesort</code>	23, 26, 414, 416	inputenc package	210
<code>\ifglssavenumberlist</code>		<code>\inputencodingname</code>	31, 419
.....	75, 183, 199, 446, 514, 523	<code>\InputIfFileExists</code>	79, 448
<code>\ifglssavewrites</code> .	34, 179, 191, 421, 510, 518	<code>\istfile</code>	517
<code>\ifglssubentrycounter</code>		<code>\istfilename</code>	43, 170, 174, 180, 181, 183, 335, 338, 427, 503, 506, 511, 513
.....	13, 38, 217, 218, 408, 424, 534, 535	<code>\item</code>	285–288, 305, 306, 324, 325, 340, 341, 345
<code>\ifglstoc</code>	49, 430		
<code>\ifglstranslate</code>	41, 417, 426		
<code>\ifglsucmark</code>	47, 48, 429		
<code>\ifglused</code>	102, 106–112, 118, 167, 191, 250, 254, 256, 259, 276–279, 367–373, 437, 464–470, 473, 502, 519, 555, 558, 560, 562		
<code>\ifglswrallowprimitivemods</code>	195, 520		
<code>\ifglsxindy</code>	43– 45, 50–53, 55–58, 68, 95, 96, 123, 168, 169, 175, 180, 181, 196, 197, 203, 333– 335, 419, 427, 428, 431–436, 442, 459, 476, 502, 503, 507, 511, 512, 521, 522, 525		
<code>\ifignoredglossary</code>			
..	90, 94, 191, 200, 442, 456, 458, 519, 523		
<code>\ifin@</code>	37, 424		
<code>\ifinlistcs</code>	209, 214, 530, 533		
<code>\ifinner</code>	6, 405		
<code>\ifKV@glslink@hyper</code>	119, 120, 474		
<code>\ifKV@glslink@local</code>	132–137, 483–486		
<code>\ifmeasuring@</code>	97, 460		
<code>\ifmmode</code>	6, 405		
<code>\ifnum</code> .	15, 33, 101, 102, 182, 211, 298, 326, 328, 330, 346, 347, 409, 421, 463, 513, 531		
<code>\ifstrempty</code>	7, 340, 405		
<code>\ifstrequal</code>	223, 538		
<code>\ifthenelse</code>	26, 39, 49, 120, 183, 223, 264, 281, 417, 425, 430, 474, 513, 538, 566		
<code>\IfTrackedLanguage</code>	175, 507		

J

<code>\jobname</code>	32, 44, 78–80, 170, 174, 179–182, 203, 335, 338, 421, 427, 448, 503, 506, 511–513, 525
-----------------------------	--

K

<code>\key@ifundefined</code>	82, 83, 450
<code>\KV@glslink@hyperfalse</code> 116, 118, 119, 130, 473, 474, 482
<code>\KV@glslink@hypertrue</code> ..	116, 131, 473, 483

L

<code>\L</code>	25, 415
<code>\l</code>	25, 415
<code>\label</code>	9, 215, 217, 406, 533, 535
<code>\language</code>	30, 419
<code>\leaders</code>	287, 288, 341

464, 465, 468, 470–474, 476, 482–503, 507, 510–514, 517–532, 534–546, 553–569	symbolpluralaccess 360, 364
\newcount 15, 16, 75, 409, 410, 446	text 70, 71, 131, 137, 159, 251, 255, 355
\newcounter 12, 13, 408	textaccess 359, 363
\newenvironment 218, 536	type 17, 71, 105, 161
\newglossary 17, 18, 36, 37, 68, 183, 410, 411, 423, 441, 442, 513, 514	user1 144, 162, 356
\newglossaryentry . . 7, 37, 74, 75, 78, 100, 233, 247, 250, 252, 254, 257, 260, 262, 265, 395–398, 405, 423, 445, 447, 462, 543, 553, 555, 557, 559, 560, 563, 564, 567	user1access 361, 365
\newglossaryentry options	user2 145, 162, 356
access 357, 359	user2access 361, 365
counter 71	user3 146, 162, 356
description	user3access 361, 365
. 29, 70, 75, 77, 89, 141, 159, 229, 258, 356	user4 147, 162, 357
descriptionaccess 360, 364	user4access 361, 366
descriptionplural 142, 356	user5 148, 163, 357
descriptionpluralaccess 360, 364	user5access 361, 366
entrycounter 215	user6 148, 163, 357
first 71, 92, 131, 138, 160, 255, 261, 355	user6access 361, 366
firstaccess 360, 364	\newglossarystyle 282, 285–297, 299–322, 324–329, 331, 332, 539
firstplural 71, 139, 161, 355	\newif 4, 5, 19, 27, 30, 33, 193, 403, 404, 412, 417, 419, 421, 520
firstpluralaccess 360, 364	\newignoredglossary 442
format 170	\newlength . . 130, 288, 299, 310, 317, 327, 482
long 112, 163, 356	\newrobustcmd 7, 62–64, 77, 78, 84, 103, 104, 118, 131–156, 158–164, 166, 167, 198, 199, 229–232, 274–279, 329, 363–366, 405, 438, 439, 447, 448, 451, 452, 464, 465, 473, 483–502, 523, 541, 542
longaccess 361, 365	\newterm 37, 423
longplural 164, 356	\newtoks 123, 179, 232, 233, 477, 510, 543
longpluralaccess 361, 365	\newwrite 32, 78, 169, 174, 179, 182, 421, 448, 503, 506, 510, 513
name . . 70, 74, 77, 89, 140, 158, 199, 355, 357	\nfss@text 6, 405
nonumberlist 72, 73	ngerman package 175
parent 72, 77	\noalign 298
plural 70, 92, 139, 355	\nobreak 286, 299, 341
pluralaccess 360, 364	\noexpand 19, 39, 51, 52, 79, 93, 94, 115, 120–123, 129, 165, 176–178, 180, 181, 183, 192, 193, 196, 199, 203, 204, 207, 219, 220, 228, 233, 247, 248, 250, 252, 254, 255, 257, 260, 262, 264, 265, 333, 353, 354, 395–399, 412, 425, 431, 432, 448, 458, 459, 471, 472, 474–476, 481, 482, 501, 508–512, 514, 519–521, 523, 525–528, 536, 537, 541, 543, 553–555, 557, 559–561, 563–567
prefix 273	\nohyperpage 226, 539
prefixfirst 273	\noindent . . . 221, 306–308, 310, 327, 328, 537
prefixfirstplural 274	\noist 338, 339, 510
prefixplural 274	
see 7, 10, 72, 77, 183, 185	
short 112, 163, 356, 362	
shortaccess 360, 365, 394	
shortpl 362	
shortplural 163, 356	
shortpluralaccess 360, 365	
sort 70, 161, 190, 221, 222	
symbol 70, 71, 143, 251–253, 255, 261, 292, 314, 355, 357, 362	
symbolaccess 360, 364	
symbolplural 144, 355	

`\nopostdesc`
 ... [37](#), [43](#), [88](#), [202](#), [340](#), [423](#), [427](#), [454](#), [525](#)
`\normalbaselineskip` [298](#)
`\ns@ACRfull` [231](#), [542](#)
`\ns@Acrfull` [230](#), [541](#)
`\ns@acrfull` [229](#), [541](#)
`\ns@ACRfullpl` [232](#), [542](#)
`\ns@Acrfullpl` [231](#), [542](#)
`\ns@acrfullpl` [231](#), [542](#)
`\ns@ACRlong` [154](#), [494](#), [495](#)
`\ns@Acrlong` [153](#), [154](#), [494](#)
`\ns@acrlong` [153](#), [494](#)
`\ns@ACRlongpl` [156](#), [496](#)
`\ns@Acrlongpl` [155](#), [156](#), [495](#)
`\ns@acrlongpl` [155](#), [495](#)
`\ns@ACRshort` [150](#), [151](#), [492](#)
`\ns@Acrshort` [150](#), [492](#)
`\ns@acrshort` [149](#), [491](#)
`\ns@ACRshortpl` [152](#), [493](#)
`\ns@Acrshortpl` [152](#), [493](#)
`\ns@acrshortpl` [151](#), [493](#)
`\ns@newglossary` [67](#), [441](#)
`\null` [122](#)–
 [129](#), [176](#)–[178](#), [203](#), [476](#)–[482](#), [508](#)–[510](#), [525](#)
`\number` [15](#), [79](#), [92](#), [102](#), [192](#), [195](#),
 [220](#), [354](#), [409](#), [448](#), [457](#), [464](#), [519](#), [520](#), [537](#)
`\numberline` [49](#), [430](#)
`\numexpr` [102](#), [464](#)

O

`\O` [25](#), [415](#)
`\o` [25](#), [415](#)
`\OE` [24](#), [415](#)
`\oe` [25](#), [415](#)
`\oldacronym` [541](#)
`\openout` [32](#), [78](#), [170](#),
 [174](#), [179](#), [335](#), [338](#), [421](#), [448](#), [503](#), [506](#), [511](#)
`\OR` [264](#), [566](#)
`\or` [5](#), [6](#), [8](#), [9](#),
 [28](#), [36](#), [215](#), [325](#), [345](#), [404](#), [406](#), [418](#), [423](#), [533](#)
`\org@glossaryentrynumbers`
 ... [202](#), [216](#), [525](#), [534](#)
`\org@glossarytitle` [201](#), [202](#), [524](#)
`\org@glspostdescription` [43](#), [427](#)
`\org@ifKV@glslink@hyper` [119](#), [120](#), [474](#)
`\outputpenalty` [298](#)

P

`\p@` [285](#), [305](#), [323](#), [324](#)
`\p@gl@hyp@opt` [117](#), [473](#)

package options:

`acronym` [17](#), [18](#), [40](#), [201](#), [229](#)
 `true` [18](#)
`counter` [20](#)
`debug`
 `showaccsupp` [7](#)
 `showtargets` [6](#), [7](#)
`description` [255](#), [256](#)
`disablemakegloss` [32](#)
`dua` [254](#)–[256](#)
`entrycounter` [12](#), [215](#), [217](#)
 `true` [13](#)
`esclocations` [594](#)
 `false` [10](#)
`footnote` [132](#)–[136](#), [252](#), [254](#), [255](#), [258](#)
`hyperfirst`
 `false` [132](#)–[136](#)
`index` [37](#)
`indexonlyfirst` [578](#)
`kernelglossredefs`
 `nowarn` [35](#)
`makeindex` [172](#), [272](#)
`nogroupskip` [289](#), [291](#), [292](#), [295](#), [296](#),
 [300](#), [301](#), [303](#), [311](#), [313](#), [315](#), [318](#), [320](#), [322](#)
`nolist` [266](#)
`nolong` [266](#), [288](#)
`nomain` [17](#)
`nonumberlist` [10](#)
`nosuper` [266](#)
`notree` [266](#)
`nowarn` [5](#)
`numberline` [8](#)
`record` [275](#)
`sanitize` [25](#), [70](#), [158](#), [159](#)
`sanitizesort` [22](#)
`savewrites` [34](#), [575](#)
 `false` [179](#)
 `true` [182](#), [189](#)
`section` [8](#), [48](#)
`sort`
 `def` [14](#), [15](#)
 `none` [14](#)
 `standard` [14](#)
 `use` [14](#), [15](#), [594](#)
`style` [9](#), [266](#)
`subentrycounter` [13](#), [215](#), [217](#)
`toc` [8](#)
 `true` [8](#)

translate	28	\printnoidxglossary options	
false	27	sort	216
translator	27	\printnumbers	37, 423
xindy	30, 31, 172, 272	\printsymbols	36, 423
\PackageError	7, 8, 17,	\ProcessOptions	273, 353
33, 40, 51, 57, 61, 62, 66, 71, 74, 75, 82–		\ProcessOptionsX	38, 424
88, 90, 91, 100, 116, 157, 178, 179, 183,		\protect	49, 114, 236, 237, 243,
186, 188, 208–210, 214, 216, 224, 225,		250, 254, 257, 375, 376, 385, 386, 391,	
234, 235, 252, 253, 259, 262, 339, 366,		392, 430, 471, 545, 546, 551, 555, 558, 560	
405, 406, 410, 421, 425, 431, 436–438,		\protected@csedef	85, 452
441, 443, 445, 446, 450–456, 462, 472,		\protected@csxdef	85, 452
496, 510, 514, 515, 517, 528–530, 533,		\protected@eappto	32, 69, 95, 192
534, 538, 539, 544, 545, 556, 557, 562, 564		\protected@edef	7, 9,
\PackageInfo		16, 19, 32, 52, 54, 56, 58, 59, 62, 67, 69,	
..... 5, 6, 32, 179, 190, 404, 420, 511, 518		85–87, 89, 90, 94–96, 106–108, 114, 119,	
\PackageWarning	5, 6, 21, 404, 413	121, 165–167, 185, 191, 193, 196, 199,	
\PackageWarningNoLine	5, 6, 21, 402–404, 413	205, 206, 215, 217, 219, 220, 224, 233,	
\pagegoal	298	259, 265, 274, 280, 334, 335, 353, 354,	
\pagelistname	43, 291, 293, 295,	362, 363, 394, 405, 406, 432, 433, 435,	
296, 302, 304, 313–317, 320–323, 425, 427		437, 456, 459, 466, 467, 471, 475, 501,	
\par	43, 221, 285–287, 305, 307–310,	518, 520, 521, 533, 536–539, 543, 562, 567	
323, 324, 326–332, 341, 346–348, 427, 537		\protected@write	66, 67, 170, 172,
\parindent		182, 183, 185, 188, 191, 199, 281, 335,	
305–310, 324, 326–328, 330–332, 346–348		440, 441, 503, 504, 513, 515, 517, 519, 523	
\parskip	305–308, 324, 326, 327	\protected@xdef	14–
\PassOptionsToPackage	273, 353	16, 19, 24, 76, 90, 91, 93, 96, 195, 203,	
\penalty	298	358, 359, 409–411, 414, 446, 459, 460, 521	
\phantomsection	48, 430	\providecommand	18, 31,
polyglossia package	27, 41	40, 41, 48, 66, 102, 131, 172, 182, 183,	
\printacronyms	411	185, 189, 204, 219, 220, 275, 285, 305,	
\PrintChanges	403	323, 354, 411, 420, 425, 429, 440, 464,	
\printglossaries	184, 514, 524	483, 504, 513, 515, 517, 525, 526, 536, 537	
\printglossary	18, 21, 36, 37, 184,	\provideglossaryentry	447, 448
201, 216, 411, 413, 423, 514, 523, 524, 534		\ProvidesFile	41, 426
\printglossary options		\ProvidesGlossariesLang	426
entrycounter	215	\ProvidesPackage	
nogroupskip	215	.. 4, 273, 280, 282, 285, 288, 294, 299,	
nonumberlist	216	304, 310, 317, 323, 333, 339, 353, 402, 403	
nopostdot	215		
numberedsection	214		
style	214		
subentrycounter	215		
title	214		
toctitle	214		
type	17, 199, 214		
\printindex	37, 423		
\printnoidxglossaries	185, 515, 524		
\printnoidxglossary	185, 188, 201,		
208, 209, 216, 515, 517, 524, 528, 529, 534			

R

\r	24, 415
\raggedright	297–304, 318–323
\raisebox	130, 482
\ref	218, 535
\refstepcounter	217, 535
\relax	5, 9,
11, 15–18, 28, 32, 33, 36–38, 53, 66, 71,	
72, 76, 79, 91, 93, 97, 101, 102, 117, 118,	
120, 122, 124–129, 158, 159, 172–176,	
178, 182, 183, 185–187, 191, 195, 196,	

198, 201, 203, 211, 214, 215, 223, 224, 266, 281, 285, 298, 305, 309, 310, 323, 325–332, 334, 337–339, 345–348, 357, 358, 376–378, 395–398, 404, 406, 407, 409–411, 417, 418, 421–424, 433, 440, 443, 444, 447, 448, 456, 458, 460, 463, 464, 473, 475–481, 497, 505–510, 513– 516, 519, 521–525, 531, 533, 538, 539, 567	\setcounter 217, 534, 535
\renewacronymstyle 385–388, 391, 393, 394, 544	\SetCustomDisplayStyle 265, 566, 567
\renewcommand 4–6, 8–11, 13, 14, 17, 18, 20–22, 26, 28–31, 33, 34, 36, 38, 42–45, 58, 69, 72, 73, 88, 100–102, 165, 167–169, 175, 176, 182–187, 203, 204, 215, 233, 234, 236–240, 242–245, 249, 251–253, 256, 258, 261, 263, 265, 283–293, 295, 296, 298–315, 318–322, 324–334, 339–347, 349–352, 354, 357, 358, 366, 370, 373, 375, 376, 384–388, 390–394, 396–398, 403–413, 416, 418–421, 423, 424, 426– 428, 436, 442, 444, 454, 462–464, 501– 503, 507, 508, 513–516, 525, 526, 533, 543–554, 556, 557, 559–562, 564, 565, 567	\SetCustomStyle 567
\renewenvironment 219, 283, 285, 289–294, 297–324, 326, 327, 329, 536	\SetDefaultAcronymDisplayStyle 249, 553, 554
\renewglossarystyle 539	\SetDefaultAcronymStyle 264, 554, 566
\RequireGlossariesLang 42, 402, 403, 425, 426	\SetDescriptionAcronymDisplayStyle 256, 558–560
\RequirePackage 4, 10, 11, 27, 28, 38, 41, 266, 272, 273, 288, 294, 295, 299, 305, 310, 317, 354, 402, 403, 407, 417, 418, 424, 426, 567, 569	\SetDescriptionAcronymStyle 264, 559, 566
\restorecounters@ 121, 475	\SetDescriptionDUAAcronymDisplayStyle 253, 254, 556, 558
\romannumeral 192, 195, 329–331, 347, 519, 520	\SetDescriptionDUAAcronymStyle 264, 557, 566
	\SetDescriptionFootnoteAcronymDisplayStyle 251, 252, 555, 556
	\SetDescriptionFootnoteAcronymStyle 264, 556, 566
	\SetDUADisplayStyle 263, 564–566
	\SetDUASStyle 264, 565, 566
	\setentrycounter 52, 172, 213, 333, 432, 505, 532, 538
	\SetFootnoteAcronymDisplayStyle 258, 560, 561
	\SetFootnoteAcronymStyle ... 264, 561, 566
	\SetGenericNewAcronym 235, 543, 544
	\setglossarypreamble 428
	\setglossarysection 429
	\setglossarystyle 202, 224, 266, 286–298, 300– 322, 325–328, 331, 332, 524, 538, 539, 567
	\setglossentrycompatibility 214, 224, 533, 537, 539
	\setkeys 27, 31, 38, 48, 90, 119, 166, 167, 202, 233, 249, 251, 253, 256, 258, 261, 263, 265, 417, 420, 424, 429, 455, 474, 501, 502, 524, 543, 554, 556, 558, 559, 561, 564, 565, 567
	\setlength 288, 299, 305– 308, 310, 317, 324, 326, 327, 331, 347, 348
	\SetSmallAcronymDisplayStyle 261, 562, 564
	\SetSmallAcronymStyle 264, 564, 566
	\setstylefile 427
	\settoheight 130, 482
	\settowidth 237, 329–331, 347, 546
	\setupglossaries 424
	\sfcode 12, 407
	\show 266–271, 400, 401, 567–569
	\showacronymlists 569
	\showglocounter 568
S	
\s@glshyp@opt 117, 473	
\s@GlsSetXdyFirstLetterAfterDigits 168, 502	
\s@GlsSetXdyNumberGroupOrder 169, 502, 503	
\s@ifglossaryexists 60, 200, 437, 524	
\s@newglossary 67, 441	
\savecounters@ 120, 475	
\seename 198, 425, 523	
\setabbreviationstyle 263	
\SetAcronymLists 412	
\SetAcronymStyle 29, 30, 419, 566	
\setacronymstyle 263, 544	
\setbool 26, 416	
\setbox 298, 299	

<code>\showglodesc</code>	568	72, 74, 75, 80, 83, 84, 95, 96, 100, 102–
<code>\showglodescplural</code>	568	104, 115, 117, 121, 122, 124–126, 128,
<code>\showglofield</code>	569	129, 165, 168–175, 178, 179, 182–186,
<code>\showglofirst</code>	568	188, 196–198, 202, 204, 208, 209, 216,
<code>\showglofirstpl</code>	568	221, 224, 263, 281, 333–339, 405, 406,
<code>\showgloflag</code>	569	410, 413, 420–422, 425, 430–437, 440,
<code>\showgloindex</code>	569	441, 444, 445, 448, 449, 451, 459, 460,
<code>\showglolevel</code>	568	462, 464, 465, 471–473, 475–479, 481,
<code>\showgloloclist</code>	569	500, 502–507, 509, 510, 513–515, 517,
<code>\showglolong</code>	569	521, 522, 524–526, 528, 529, 534, 537, 539
<code>\showgloname</code>	568	<code>\strut</code>
<code>\showgloparent</code>	567	221, 286–290,
<code>\showgloplural</code>	568	292, 300, 301, 303, 311, 313, 314, 318,
<code>\showgloshort</code>	569	320, 321, 328, 340–344, 346, 349–352, 537
<code>\showglosort</code>	568	<code>\subglossentry</code>
<code>\showglossaries</code>	569	95,
<code>\showglossarycounter</code>	569	203, 212, 221, 284, 285, 287–290, 292,
<code>\showglossaryentries</code>	569	300, 301, 303, 311, 313, 314, 318, 320,
<code>\showglossaryin</code>	569	321, 325, 326, 328, 330, 459, 525, 531, 537
<code>\showglossaryout</code>	569	<code>\subitem</code>
<code>\showglossarytitle</code>	569	305, 324, 325, 345
<code>\showglosymbol</code>	569	<code>\subsubitem</code>
<code>\showglosymbolplural</code>	569	305, 324, 325, 345
<code>\showglotext</code>	568	supertabular package
<code>\showglotype</code>	568	11, 266, 310, 317
<code>\showglouseri</code>	568	<code>\symbolname</code>
<code>\showglouserii</code>	568	43, 292,
<code>\showglouseriii</code>	568	293, 296, 304, 315–317, 322, 323, 425, 427
<code>\showglouseriv</code>	568	
<code>\showglouserv</code>	568	
<code>\showglouservi</code>	568	
<code>\SmallNewAcronymDef</code>	261, 563, 564	
<code>\space</code> ...	7, 8, 31–36, 40, 51, 55, 56, 58, 72,	
	74, 75, 100, 103, 104, 114, 115, 117, 165,	
	170–174, 178, 180, 181, 183–185, 188,	
	199, 202, 204, 218, 224, 230, 236, 237,	
	239, 240, 242–245, 254, 259, 282, 284–	
	287, 289, 300, 311, 318, 324–326, 328,	
	330, 331, 333, 334, 336–338, 340–343,	
	345–349, 351, 375, 376, 385–388, 390,	
	392–395, 405, 406, 420–422, 425, 431,	
	434–437, 444, 445, 462, 464, 465, 471–	
	473, 500, 503–506, 510–515, 517, 523,	
	524, 526, 535, 539, 541, 545–553, 558, 562	
<code>\spacefactor</code>	12, 407	
<code>\SS</code>	25, 415	
<code>\ss</code>	25, 415	
<code>\string</code>	7, 8, 17,	
	21, 31–36, 40, 50, 51, 54–56, 58, 66, 67,	
		T
		<code>\t</code>
		24, 415
		<code>\tablehead</code>
		311–323
		<code>\tabletail</code>
		311–323
		<code>\tabularnewline</code>
		289–293, 295,
		296, 300–304, 311–323, 343, 344, 349, 350
		<code>\texorpdfstring</code>
		161, 498
		<code>\textbar</code>
		282
		<code>\textbf</code>
		221, 227, 323, 345–348, 537, 540
		textcase package
		4
		<code>\textit</code>
		227, 541
		<code>\textmd</code>
		227, 540
		<code>\textrm</code>
		227, 540
		<code>\textsc</code>
		228, 238, 244, 252, 256, 258,
		261, 393, 541, 547, 552, 556, 560, 562, 564
		<code>\textsf</code>
		227, 540
		<code>\textsl</code>
		228, 541
		<code>\textsmaller</code>
		238, 244, 252, 256,
		258, 261, 393, 547, 552, 556, 560, 562, 564
		<code>\texttt</code>
		227, 540
		<code>\textulc</code>
		229, 541
		<code>\textup</code>
		228, 229, 541
		<code>\TH</code>
		25, 415
		<code>\th</code>
		25, 415
		<code>\the</code>
		39, 42,
		51, 56, 58, 67, 124–129, 170, 174, 176–
		178, 189, 191, 195, 199, 211, 219, 221,

227, 233, 234, 236, 237, 242, 243, 247, 248, 250, 252–255, 257, 260, 262, 264, 265, 333, 335, 338, 354, 385, 386, 390, 391, 394–399, 425, 426, 432, 435, 437, 441, 477–482, 503, 506, 508–510, 518, 520, 523, 531, 536, 537, 540, 543–546, 550, 551, 553–555, 557, 559–561, 563–567	\usepackage 208, 209, 528, 529
\the@numberlist 165, 501	\UTFviii@two@octets 25, 415
\theglossary 5, 403	\UTFviii@two@octets@combine 25, 415
\theglossaryentry 12, 218, 408, 535	
\theglossarysubentry 13, 218, 408, 535	V
\theglentrycounter 120, 121, 193, 195, 334, 335, 474, 475, 520, 521	\v 24, 415
\theH 197, 522	\vbox 298, 299
\theHglossaryentry 12, 408	\vsize 298, 299
\theHglossarysubentry 13, 408	\vskip 285, 298, 305, 323
\theHglentrycounter 121, 193, 195, 196, 475, 520, 521	\vss 298, 299
\thesection 39, 425	W
\this@dialect 42, 402, 403, 426	\warn@nomakeglossaries 183, 184, 186, 513–515
\tiny 6, 405	\warn@noprintglossary 183–185, 203, 514, 515, 523, 525
\toks@ 39, 42, 51, 56, 58, 67, 124–129, 176–178, 199, 219–221, 227, 333, 353, 354, 425, 426, 432, 435, 437, 441, 477–482, 508–510, 523, 536, 537, 540	\write 33, 80, 102, 170–175, 180, 181, 185, 189, 204, 335–339, 421, 448, 449, 464, 503–507, 511, 512, 515, 518, 525, 526
\toprule 295, 296	\writeist ... 178, 179, 339, 503, 506, 507, 510
tracklang package 41, 402	X
\trans@languages 42, 426	\x 540
\translate 42, 43, 426, 427	\xatlevel@ 120, 474
\translatelet ... 17, 18, 36, 37, 410, 411, 423	\xcapitalisewords 161, 498
translator package . 17, 18, 27, 36, 37, 41, 42, 200	\xdef 92, 281, 456–458, 525
\triangleright 6, 405	\xglaccsupp 363
\ttfamily 7, 405	\xifinlist 78, 79
\TX@trial 97, 460	\xifinlistcs 205, 206, 209, 526, 527, 529
\typeout 21, 413	xindy 571
U	xindy 10, 14, 24, 30, 31, 33, 44, 45, 50, 53, 55, 57, 58, 96, 128, 129, 168, 169, 171, 189, 193, 194, 196, 200, 203, 221, 222, 272, 334
\u 24, 415	\xmakefirstuc 106–108, 114, 157, 159, 274, 466, 467, 471, 496, 497
\uccode 211, 531	\xspace 228, 541
\undef 73, 79, 200, 444, 448, 523	xspace package 4, 228
\unskip 88, 287, 288, 341, 454	Y
\unvbox 298, 299	\year 170, 174, 335, 338, 503, 506
\usedictionary 42, 426	Z
	\z@ 298