

# Documented Code For glossaries v4.49

Nicola L.C. Talbot

Dickimaw Books

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

2021-11-01

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 v4.49: L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> 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

<b>1</b>	<b>Main Package Code</b>	<b>4</b>
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	168
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
<b>2</b>	<b>Prefix Support (glossaries-prefix Code)</b>	<b>273</b>
<b>3</b>	<b>Glossary Styles</b>	<b>280</b>
3.1	Glossary hyper-navigation definitions (glossary-hypernav package)	280
3.2	In-line Style (glossary-inline.sty)	282
3.3	Rollback v4.46 (glossary-list-2020-03-19.sty)	285
3.4	List Style (glossary-list.sty)	286
3.5	Glossary Styles using longtable (the glossary-long package)	290
3.6	Glossary Styles using longtable and booktabs (the glossary-longbooktabs) package	297
3.7	Glossary Styles using longtable (the glossary-longragged package)	301
3.8	Glossary Styles using multicols (glossary-mcols.sty)	307
3.9	Glossary Styles using supertabular environment (glossary-super package)	312
3.10	Glossary Styles using supertabular environment (glossary-superragged package)	319

3.11 Tree Styles (glossary-tree.sty) . . . . .	325
<b>4 Backwards Compatibility</b>	<b>335</b>
4.1 glossaries-compatible-207 . . . . .	335
4.2 glossaries-compatible-307 . . . . .	341
<b>5 Accessibility Support (glossaries-accsupp Code)</b>	<b>355</b>
5.1 Defining Replacement Text . . . . .	357
5.2 Accessing Replacement Text . . . . .	361
5.3 Displaying the Glossary . . . . .	386
5.4 Acronyms . . . . .	387
5.5 Debugging Commands . . . . .	401
<b>6 Multi-Lingual Support</b>	<b>404</b>
6.1 Polyglossia Captions . . . . .	404
6.2 Rollback v4.46 (glossaries-2020-03-19.sty) . . . . .	405
<b>Glossary</b>	<b>573</b>
<b>Change History</b>	<b>574</b>
<b>Index</b>	<b>602</b>

# 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{v4.49}{2021-11-01}
```

Declare package:

```
4 \ProvidesPackage{glossaries}[2021/11/01 v4.49 (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 \glsshowtargetsymbols\marginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}

```

`howtargetsymbols`

```

91 \newcommand*{\glsshowtargetsymbols}{\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 underscores.

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

```

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

```
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   \ifglsacronym
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   \ifglsacronym
395     \ifbool{glscompatible-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   \glsacronymtrue
406   \def\@gls@do@acronymsdef{%
407     \DeclareAcronymList{acronym}%
408     \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
409     \renewcommand*{\acronymtype}{acronym}%
410   }%
411   \translatelet{\glossarytoctitle}{Acronyms}%
412 }%
413 }%
414 }
```

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

```

#### sanitize

```

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

shortcuts   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 xyindy 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 \ifglsxindy
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.

**glsnoindexwarning** Issues a warning if xindy hasn't been specified. These warnings can be suppressed by re-defining `\glsnoindexwarning` to ignore its argument

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

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

```
1373 \newcommand*{\glsnomakeindexwarning}[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 \ifglxindy
    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 \ifglxsindy
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 \ifglxsindy
```

`\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 \ifglxindy
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 \ifglxindy
1659 \ifglxindy@glsnumbers
1660 \def\@xdylettergroups{(define-letter-group
1661 \string\glsnumbers\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 \ifglsdesscsuppressed{<label>}{<true part>}{<false part>} Does <true part> if the description is just \nopostdesc otherwise does <false part>.

```

1775 \newcommand*{\ifglsdesscsuppressed}[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 }
```

urrentfieldvalue

```
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  $\text{\LaTeX}$ , `\@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}}
```

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

```

2098 \newcommand*\@glsnodelsc{%
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@startswithexpandonce#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@gldefs@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@defglossaryentry}{\gls@defdocnewglossaryentry}%
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 }

```

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

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

```
2517 \newcommand*\glsunexpandedfieldvalue[2]{%
2518   \ifcsname glo\glsdetoklabel{#1}@#2\endcsname
2519   \expandafter\expandonce\csname glo\glsdetoklabel{#1}@#2\expandafter\endcsname
2520   \fi
2521 }
```

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

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

```
2522 \newcommand{\ifglsfieldefeq}[5]{%
2523   \glsdoifexists{#1}%
2524   {%
2525     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2526     \ifcsdef{glo@\@glo@label @#2}%
2527     {%
2528       \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
2529     }%
2530     {%
2531       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2532     }%
2533   }%
2534 }
```

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

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

```
2535 \newcommand{\ifglsfielddedefeq}[5]{%
2536   \glsdoifexists{#1}%
2537   {%
2538     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2539     \ifcsdef{glo@\@glo@label @#2}%
2540     {%
2541       \expandafter\ifdefstrequal
2542       \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
2543     }%
2544     {%
2545       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2546     }%
2547   }%
2548 }
```

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

As above but uses \ifcsstrequal instead of \ifdefstrequal

```
2549 \newcommand{\ifglsfieldcseq}[5]{%
2550   \glsdoifexists{#1}%
2551   {%
2552     \protected@edef\@glo@label{\glsdetoklabel{#1}}%
2553     \ifcsdef{glo@\@glo@label @#2}%
2554     {%
2555       \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
2556     }%
2557     {%
2558       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
2559     }%
2560   }%
2561 }
```

gls.writedefhook

```
2562 \newcommand*{\gls.writedefhook}{}%
```

gls@assign@desc

```
2563 \newcommand*{\gls@assign@desc}[1]{%
2564   \gls@assign@field{#1}{desc}{\@glo@desc}%
2565   \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
2566 }
```

ewglossaryentry

```
2567 \newcommand{\longnewglossaryentry}[3]{%
2568   \glsdoifnoexists{#1}%
2569   {%
2570     \bgroup
2571     \let\@org@newglossaryentryprehook\@newglossaryentryprehook
2572     \long\def\@newglossaryentryprehook{%
2573       \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
2574       \@org@newglossaryentryprehook
2575     }%
2576     \renewcommand*{\gls@assign@desc}[1]{%
2577       \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
2578       \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
2579     }
2580     \gls@defglossaryentry{#1}{#2}%
2581   \egroup
2582 }%
2583 }
```

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

```
2584 \@onlypreamble{\longnewglossaryentry}
```



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

```
2585 \newcommand{\longprovideglossaryentry}[3]{%
2586   \ifglstryexists{#1}{}%
2587   {\longnewglossaryentry{#1}{#2}{#3}}%
2588 }
2589 \@onlypreamble{\longprovideglossaryentry}
```

`efglossaryentry`

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

Defines a new entry without checking if it already exists.

```
2590 \newcommand{\gls@defglossaryentry}[2]{%
```

Prevent any further use of `\GlsSetQuote`:

```
2591 \let\GlsSetQuote\gls@nosetquote
```

Store label

```
2592 \protected@edef\@glo@label{\glsdetoklabel{#1}}%
```

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

```
2593 \let\glslabel\@glo@label
```

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

```
2594 \let\@glo@name\@gls@name
```

```
2595 \let\@glo@desc\@gls@desc
```

```
2596 \let\@glo@descplural\@gls@default@value
```

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

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

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

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

```
2601 \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.)

```
2602 \let\@glo@first\@gls@default@value
```

```
2603 \let\@glo@firstplural\@gls@default@value
```

Set the default sort:

```
2604 \let\@glo@sort\@gls@default@value
```

Set the default counter:

```
2605 \let\@glo@counter\@gls@default@value
```

```
2606 \def\@glo@see{}
```

```
2607 \def\@glo@parent{}
```

```
2608 \def\@glo@prefix{}
```

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

```
2609 \@gls@initnonnumberlist
```

```
2610 \def\@glo@useri{}%
2611 \def\@glo@userii{}%
2612 \def\@glo@useriii{}%
2613 \def\@glo@useriv{}%
2614 \def\@glo@userv{}%
2615 \def\@glo@uservi{}%
```

```
2616 \def\@glo@short{}%
2617 \def\@glo@shortpl{}%
2618 \def\@glo@long{}%
2619 \def\@glo@longpl{}%
```

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

```
2620 \@newglossaryentryprehook
```

Extract key-val information from third parameter:

```
2621 \setkeys{glossentry}{#2}%
```

Check there is a default glossary.

```
2622 \ifundef\glsdefaulttype
2623 {%
2624   \PackageError{glossaries}%
2625     {No default glossary type (have you used 'nomain' by mistake?)}%
2626     {If you use package option 'nomain' you must define
2627      a new glossary before you can define entries}%
2628 }%
2629 {}%
```

Assign type. This must be fully expandable

```
2630 \gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
2631 \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.

```
2632 \ifcsundef{glolist@\@glo@type}%
2633 {%
2634   \PackageError{glossaries}%
2635     {Glossary type '\@glo@type' has not been defined}%
2636     {You need to define a new glossary type, before making entries
2637      in it}%
2638 }%
2639 {}%
```

Check if it's an ignored glossary

```
2640 \ifignoredglossary\@glo@type
2641 {%
```

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

```

2642     \ifx\@glo@desc\@glsnodesc
2643         \let\@glo@desc\@empty
2644     \fi
2645 }%
2646 {%
2647 }%
2648 \protected@edef\@glo@list@{\csname glo@list@\@glo@type\endcsname}%

2649 \expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
2650     \@glo@list@{\@glo@label},}%
2651 }%
```

Initialise level to 0.

```
2652 \gls@level=0\relax
```

Has this entry been assigned a parent?

```
2653 \ifx\@glo@parent\@empty
```

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

```
2654 \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
2655 \else
```

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

```

2656 \ifdefequal\@glo@label\@glo@parent%
2657 {%
2658     \PackageError{glossaries}{Entry '@glo@label' can't be its own parent}{}%
2659     \def\@glo@parent{}%
2660     \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
2661 }%
2662 {%
```

Check the parent exists:

```

2663 \ifglstryexists{\@glo@parent}%
2664 {%
```

Parent exists. Set \glo@<label>@parent.

```

2665 \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
2666     \@glo@parent}%
```

Determine level.

```

2667 \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
2668 \advance\gls@level by 1\relax
```

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

```

2669 \ifx\@glo@name\@glsnoname
2670     \expandafter\let\expandafter\@glo@name
2671     \csname glo@\@glo@parent @name\endcsname
```

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

```

2672 \ifx\@glo@plural\@gls@default@value
2673     \expandafter\let\expandafter\@glo@plural
2674     \csname glo@\@glo@parent @plural\endcsname
```

```

2675         \fi
2676     \fi
2677 }%
2678 {%

```

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

```

2679     \PackageError{glossaries}%
2680     {%
2681         Invalid parent '\@glo@parent'
2682         for entry '\@glo@label' - parent doesn't exist%
2683     }%
2684     {%
2685         Parent entries must be defined before their children%
2686     }%
2687     \def\@glo@parent{%
2688         \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
2689     }%
2690 }%
2691 \fi

```

Set the level for this entry

```

2692 \expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%

```

Define commands associated with this entry:

```

2693 \gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
2694 \letcs\@glo@sort{glo@\@glo@label @sortvalue}%
2695 \gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
2696 \expandafter\gls@assign@field\expandafter
2697     {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
2698     {\@glo@label}{plural}{\@glo@plural}%
2699 \expandafter\gls@assign@field\expandafter
2700     {\csname glo@\@glo@label @text\endcsname}%
2701     {\@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.

```

2702 \ifx\@glo@first\@gls@default@value
2703     \expandafter\gls@assign@field\expandafter
2704         {\csname glo@\@glo@label @plural\endcsname}%
2705         {\@glo@label}{firstpl}{\@glo@firstplural}%
2706 \else
2707     \expandafter\gls@assign@field\expandafter
2708         {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
2709         {\@glo@label}{firstpl}{\@glo@firstplural}%
2710 \fi

2711 \ifcsundef{@glotype@\@glo@type @counter}%
2712 {%
2713     \def\@glo@defaultcounter{\glscounter}%
2714 }%
2715 {%

```

```

2716 \letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
2717 }%
2718 \gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
2719 \gls@assign@field{\@glo@label}{useri}{\@glo@useri}%
2720 \gls@assign@field{\@glo@label}{userii}{\@glo@userii}%
2721 \gls@assign@field{\@glo@label}{useriii}{\@glo@useriii}%
2722 \gls@assign@field{\@glo@label}{useriv}{\@glo@useriv}%
2723 \gls@assign@field{\@glo@label}{userv}{\@glo@userv}%
2724 \gls@assign@field{\@glo@label}{uservi}{\@glo@uservi}%
2725 \gls@assign@field{\@glo@label}{short}{\@glo@short}%
2726 \gls@assign@field{\@glo@label}{shortpl}{\@glo@shortpl}%
2727 \gls@assign@field{\@glo@label}{long}{\@glo@long}%
2728 \gls@assign@field{\@glo@label}{longpl}{\@glo@longpl}%
2729 \ifx\@glo@name\@gls@name
2730 \@gls@name
2731 \let\@glo@name\@gls@default@value
2732 \fi
2733 \gls@assign@field{\@glo@label}{name}{\@glo@name}%

```

Set default numberlist if not defined:

```

2734 \ifcsundef{glo@\@glo@label @numberlist}%
2735 {%
2736 \csxdef{glo@\@glo@label @numberlist}{%
2737 \noexpand\@gls@missingnumberlist{\@glo@label}}%
2738 }%
2739 {}%

```

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

```

2740 \@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.

```

2741 \def\@glo@@desc{\@glo@first}%
2742 \ifx\@glo@desc\@glo@@desc
2743 \let\@glo@desc\@glo@first
2744 \fi
2745 \ifx\@glo@desc\@gls@nodesc
2746 \@gls@nodesc
2747 \let\@glo@desc\@gls@default@value
2748 \fi
2749 \gls@assign@desc{\@glo@label}%

```

Set the sort key for this entry:

```

2750 \@gls@defsort{\@glo@type}{\@glo@label}%

2751 \def\@glo@@symbol{\@glo@text}%
2752 \ifx\@glo@symbol\@glo@@symbol
2753 \let\@glo@symbol\@glo@text
2754 \fi
2755 \gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
2756 \expandafter

```

```

2757 \gls@assign@field\expandafter
2758 {\csname glo@\@glo@label @symbol\endcsname}
2759 {\@glo@label}{symbolplural}{\@glo@symbolplural}%

```

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

```

2760 \expandafter\protected\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
2761 \noexpand\global
2762 \noexpand\let\expandafter\noexpand
2763 \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
2764 }%
2765 \expandafter\protected\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
2766 \noexpand\global
2767 \noexpand\let\expandafter\noexpand
2768 \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
2769 }%
2770 \csname glo@\@glo@label @flagfalse\endcsname

```

Sort out any cross-referencing if required.

```

2771 \@glo@autosee

```

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

```

2772 \ifignoredglossary\@glo@type
2773 {%
2774 \csdef{glo@\@glo@label @index}{}%
2775 }
2776 {%
2777 \do@glo@storeentry{\@glo@label}%
2778 }%

```

Define entry counters if enabled:

```

2779 \@newglossaryentry@defcounters

```

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

```

2780 \@newglossaryentryposthook
2781 }

```

\@glo@autosee Automatically implement \glssee.

```

2782 \newcommand*{\@glo@autosee}{%
2783 \ifdefvoid\@glo@see{%
2784 {%
2785 \protected@edef\@do@glssee{%
2786 \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
2787 \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
2788 \@do@glssee
2789 }%
2790 \@glo@autoseehook
2791 }%

```

glo@autoseehook

```

2792 \newcommand*{\@glo@autoseehook}{%

```

aryentryprehook Allow extra information to be added to glossary entries:

```
2793 \newcommand*{\@newglossaryentryprehook}{}
```

ryentryposthook Allow extra information to be added to glossary entries:

```
2794 \newcommand*{\@newglossaryentryposthook}{}
```

try@defcounters

```
2795 \newcommand*{\@newglossaryentry@defcounters}{}
```

\glsmoveentry Moves entry whose label is given by first argument to the glossary named in the second argument.

```
2796 \newcommand*{\glsmoveentry}[2]{%
2797   \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
2798   \protected@edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
2799   \def\glo@list{,}%
2800   \for\gsentries[\glo@type]{\glo@label}%
2801   {%
2802     \ifdefequal\@glo@thislabel\glo@label
2803       {\protected@eappto\glo@list{\glo@label,}}%
2804     }%
2805     \cslet\glo@list@\glo@type{\glo@list}%
2806     \csdef{glo@\@glo@thislabel @type}{#2}%
2807 }
```

ssaryentryfield Indicate what command should be used to display each entry in the glossary. (This enables the glossaries-accsupp package to use \accsuppglossaryentryfield instead.)

```
2808 \ifglxindy
2809   \newcommand*{\@glossaryentryfield}{\string\glossentry}
2810 \else
2811   \newcommand*{\@glossaryentryfield}{\string\glossentry}
2812 \fi
```

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

```
2813 \ifglxindy
2814   \newcommand*{\@glossarysubentryfield}{%
2815     \string\subglossentry}
2816 \else
2817   \newcommand*{\@glossarysubentryfield}{%
2818     \string\subglossentry}
2819 \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.)

```
2820 \newcommand{\@glo@storeentry}[1]{%
```

Escape makeindex/xindy special characters in the label:

```
2821 \protected@edef\@glo@esclabel{#1}%
```

```
2822 \@gls@checkmkidxchars\@glo@esclabel
```

Get the sort string and escape any special characters

```
2823 \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
```

```
2824 \@gls@checkmkidxchars\@glo@sort
```

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

```
2825 \@gls@checkmkidxchars\@glo@prefix
```

Get the parent, if one exists

```
2826 \protected@edef\@glo@parent{\csname glo@#1@parent\endcsname}%
```

Write the information to the glossary file.

```
2827 \ifglxindy
```

Store using xindy syntax.

```
2828 \ifx\@glo@parent\@empty
```

Entry doesn't have a parent

```
2829 \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
```

```
2830 (\string"\@glo@sort\string" %
```

```
2831 \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
```

```
2832 }%
```

```
2833 \else
```

Entry has a parent

```
2834 \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
```

```
2835 \csname glo@\@glo@parent @index\endcsname
```

```
2836 (\string"\@glo@sort\string" %
```

```
2837 \string"\@glo@prefix\@glossarysubentryfield
```

```
2838 {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
```

```
2839 }%
```

```
2840 \fi
```

```
2841 \else
```

Store using makeindex syntax.

```
2842 \ifx\@glo@parent\@empty
```

Sanitize \@glo@prefix

```
2843 \@onelevel@sanitize\@glo@prefix
```

Entry doesn't have a parent

```
2844 \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
```

```
2845 \@glo@sort\@gls@actualchar\@glo@prefix
```

```
2846 \@glossaryentryfield{\@glo@esclabel}%
```

```
2847 }%
```

```
2848 \else
```



Entry has a parent

```

2849 \expandafter\protected\edef\csname glo@#1@index\endcsname{%
2850 \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
2851 \@glo@sort\@gls@actualchar\@glo@prefix
2852 \@glossarysubentryfield
2853 {\csname glo@#1@level\endcsname}\@glo@esclabel}%
2854 }%
2855 \fi
2856 \fi
2857 }

```

## 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`

```

2858 \AtBeginDocument{%
2859 \ifpackageloaded{amsmath}%
2860 {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
2861 {}}%
2862 }
2863 \newcommand*\@gls@ifnotmeasuring[1]{%
2864 \ifmeasuring@
2865 \else
2866 #1%
2867 \fi
2868 }
2869 \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.

```

2870 \def\@gls@patchtabularx#1\hbox#2#3!!{%
2871 \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
2872 }
2873 \newcommand*\glspatchtabularx{%
2874 \ifdef\TX@trial
2875 {%
2876 \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
2877 \let\glspatchtabularx\relax
2878 }%
2879 {}}%
2880 }

```

`\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.

```

2881 \newcommand*{\glsreset}[1]{%
2882   \gls@ifnotmeasuring
2883   {%
2884     \glsdoifexists{#1}%
2885     {%
2886       \@glsreset{#1}%
2887     }%
2888   }%
2889 }

```

`\glslocalreset` As above, but with only a local effect:

```

2890 \newcommand*{\glslocalreset}[1]{%
2891   \gls@ifnotmeasuring
2892   {%
2893     \glsdoifexists{#1}%
2894     {%
2895       \@glslocalreset{#1}%
2896     }%
2897   }%
2898 }

```

`\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.

```

2899 \newcommand*{\glsunset}[1]{%
2900   \gls@ifnotmeasuring
2901   {%
2902     \glsdoifexists{#1}%
2903     {%
2904       \@glsunset{#1}%
2905     }%
2906   }%
2907 }

```

`\glslocalunset` As above, but with only a local effect:

```

2908 \newcommand*{\glslocalunset}[1]{%
2909   \gls@ifnotmeasuring
2910   {%
2911     \glsdoifexists{#1}%
2912     {%
2913       \@glslocalunset{#1}%
2914     }%
2915   }%
2916 }

```

`\@glslocalunset` Local unset. This defaults to just `\@glslocalunset` but is changed by `\glsenableentrycount`.

```

2917 \newcommand*{\@glslocalunset}{\@glslocalunset}

```

`@@glslocalunset` Local unset without checks.

```

2918 \newcommand*{\@@glslocalunset}[1]{%
2919   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
2920 }

```

`\@glsunset` Global unset. This defaults to just `\@@glsunset` but is changed by `\glsenableentrycount`.

```

2921 \newcommand*{\@glsunset}{\@@glsunset}

```

`\@@glsunset` Global unset without checks.

```

2922 \newcommand*{\@@glsunset}[1]{%
2923   \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
2924 }

```

`\@glslocalreset` Local reset. This defaults to just `\@@glslocalreset` but is changed by `\glsenableentrycount`.

```

2925 \newcommand*{\@glslocalreset}{\@@glslocalreset}

```

`@@glslocalreset` Local reset without checks.

```

2926 \newcommand*{\@@glslocalreset}[1]{%
2927   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
2928 }

```

`\@glsreset` Global reset. This defaults to just `\@@glsreset` but is changed by `\glsenableentrycount`.

```

2929 \newcommand*{\@glsreset}{\@@glsreset}

```

`\@@glsreset` Global reset without checks.

```

2930 \newcommand*{\@@glsreset}[1]{%
2931   \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
2932 }

```

Reset all entries for the named glossaries (supplied in a comma-separated list). Syntax:

```

\glsresetall[<glossary-list>]

```

```

2933 \newcommand*{\glsresetall}[1][\@glo@types]{%
2934   \forallglsentries[#1]{\@glsentry}%
2935   {%
2936     \glsreset{\@glsentry}%
2937   }%
2938 }

```

As above, but with only a local effect:

```

\lslocalresetall

```

```

2939 \newcommand*{\glslocalresetall}[1][\@glo@types]{%
2940   \forallglsentries[#1]{\@glsentry}%
2941   {%
2942     \glslocalreset{\@glsentry}%
2943   }%
2944 }

```

Unset all entries for the named glossaries (supplied in a comma-separated list). Syntax:  
`\glsunsetall[⟨glossary-list⟩]`

`\glsunsetall`

```
2945 \newcommand*{\glsunsetall}[1][\@glo@types]{%
2946   \forallglsentries[#1]{\@glsentry}%
2947   {%
2948     \glsunset{\@glsentry}%
2949   }%
2950 }
```

As above, but with only a local effect:

`\glslocalunsetall`

```
2951 \newcommand*{\glslocalunsetall}[1][\@glo@types]{%
2952   \forallglsentries[#1]{\@glsentry}%
2953   {%
2954     \glslocalunset{\@glsentry}%
2955   }%
2956 }
```

## 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  $\LaTeX$  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.

`\glsentrydefcounters` Define entry fields to keep track of how many times that entry has been marked as used.

```
2957 \newcommand*{\@newglossaryentry@defcounters}{%
2958   \csdef{glo@\@glo@label @currcount}{0}%
2959   \csdef{glo@\@glo@label @prevcount}{0}%
2960 }
```

`\glsenableentrycount` Enables tracking of how many times an entry has been marked as used.

```
2961 \newcommand*{\glsenableentrycount}{%
```

Enable new entry fields.

```
2962 \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
```

Disable `\newglossaryentry` in the document environment.

```
2963 \renewcommand*{\gls@defdocnewglossaryentry}{%
2964   \renewcommand*{\newglossaryentry}[2]{%
2965     \PackageError{glossaries}{\string\newglossaryentry\space
2966     may only be used in the preamble when entry counting has
```

```

2967     been activated}{If you use \string\glsenableentrycount\space
2968     you must place all entry definitions in the preamble not in
2969     the document environment}%
2970 }%
2971 }%

```

Define commands `\glsentrycurrcount` and `\glsentryprevcount` to access these new fields. Default to zero if undefined.

```

2972 \newcommand*\glsentrycurrcount}[1]{%
2973   \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
2974   {0}{\@gls@entry@field{##1}{currcount}}%
2975 }%
2976 \newcommand*\glsentryprevcount}[1]{%
2977   \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
2978   {0}{\@gls@entry@field{##1}{prevcount}}%
2979 }%

```

Make the unset and reset functions also increment or reset the entry counter.

```

2980 \renewcommand*\@glsunset}[1]{%
2981   \@@glsunset{##1}%
2982   \@gls@increment@currcount{##1}%
2983 }%
2984 \renewcommand*\@glslocalunset}[1]{%
2985   \@@glslocalunset{##1}%
2986   \@gls@local@increment@currcount{##1}%
2987 }%
2988 \renewcommand*\@glsreset}[1]{%
2989   \@@glsreset{##1}%
2990   \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
2991 }%
2992 \renewcommand*\@glslocalreset}[1]{%
2993   \@@glslocalreset{##1}%
2994   \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
2995 }%

```

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.)

```

2996 \def\@cgl@s@##1##2[##3]{%
2997   \ifnum\glsentryprevcount{##2}=1\relax
2998     \cgl@sformat{##2}{##3}%
2999     \glsunset{##2}%
3000   \else
3001     \@gls@{##1}{##2}[##3]%
3002   \fi
3003 }%

```

Similarly for the analogous commands. No case change plural:

```

3004 \def\@cgl@sp@##1##2[##3]{%
3005   \ifnum\glsentryprevcount{##2}=1\relax
3006     \cgl@spformat{##2}{##3}%
3007     \glsunset{##2}%

```

```

3008 \else
3009 \cGlspl@{##1}{##2}[##3]%
3010 \fi
3011 }%

```

First letter uppercase singular:

```

3012 \def\cGls@##1##2[##3]{%
3013 \ifnum\glsentryprevcount{##2}=1\relax
3014 \cGlsformat{##2}{##3}%
3015 \glsunset{##2}%
3016 \else
3017 \cGls@{##1}{##2}[##3]%
3018 \fi
3019 }%

```

First letter uppercase plural:

```

3020 \def\cGlspl@##1##2[##3]{%
3021 \ifnum\glsentryprevcount{##2}=1\relax
3022 \cGlsplformat{##2}{##3}%
3023 \glsunset{##2}%
3024 \else
3025 \cGlspl@{##1}{##2}[##3]%
3026 \fi
3027 }%

```

Write information to aux file at the end of the document

```

3028 \AtEndDocument{\cGls@write@entrycounts}%

```

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

```

3029 \renewcommand*\cGls@entry@count[2]{%
3030 \csgdef{glo@cGlsdetoklabel{##1}@prevcount}{##2}%
3031 }%

```

\glsenableentrycount may only be used once and only in the preamble.

```

3032 \let\glsenableentrycount\relax
3033 }
3034 \@onlypreamble\glsenableentrycount

```

ement@currcount

```

3035 \newcommand*\cGls@increment@currcount[1]{%
3036 \csxdef{glo@cGlsdetoklabel{##1}@currcount}{%
3037 \number\numexpr\glsentrycurrcount{##1}+1}%
3038 }

```

ement@currcount

```

3039 \newcommand*\cGls@local@increment@currcount[1]{%
3040 \csedef{glo@cGlsdetoklabel{##1}@currcount}{%
3041 \number\numexpr\glsentrycurrcount{##1}+1}%
3042 }

```

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.)

```

3043 \newcommand*{\@gls@write@entrycounts}{%
3044   \immediate\write\@auxout
3045     {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
3046   \forallglsentries{\@glsentry}{%
3047     \ifglsused{\@glsentry}%
3048     {\immediate\write\@auxout
3049       {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
3050     }%
3051   }%
3052 }

```

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

```

3053 \newcommand*{\@gls@entry@count}[2]{}

```

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

```

3054 \newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}

```

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

```

3055 \newcommand*{\@cgl}[2][ ]{%
3056   \new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[ ]}%
3057 }

```

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

```

3058 \def\@cgl@#1#2[#3]{%
3059   \GlossariesWarning{\string\cgl\space is defaulting to
3060     \string\gls\space since you haven't enabled entry counting}%
3061   \@gls@{#1}{#2}[#3]%
3062 }

```

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

```

3063 \newcommand*{\cglformat}[2]{%
3064   \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
3065 }

```

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

```

3066 \newrobustcmd*{\cGl}{\@gls@hyp@opt\@cGl}

```

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

```

3067 \newcommand*{\@cGl}[2][ ]{%

```

```

3068 \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2} []}%
3069 }

```

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

```

3070 \def\@cGls@#1#2[#3]{%
3071 \GlossariesWarning{\string\cGls\space is defaulting to
3072 \string\Gls\space since you haven't enabled entry counting}%
3073 \@Gls@{#1}{#2}[#3]%
3074 }

```

`\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.

```

3075 \newcommand*{\cGlsformat}[2]{%
3076 \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
3077 }

```

`\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.)

```

3078 \newrobustcmd*{\cglsp1}{\@gls@hyp@opt\@cglsp1}

```

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

```

3079 \newcommand*{\@cglsp1}[2] []{%
3080 \new@ifnextchar[{\@cglsp1@{#1}{#2}}{\@cglsp1@{#1}{#2} []}%
3081 }

```

`\@cglsp1@` Read in the final optional argument. This defaults to same behaviour as `\glsp1` but issues a warning.

```

3082 \def\@cglsp1@#1#2[#3]{%
3083 \GlossariesWarning{\string\cglsp1\space is defaulting to
3084 \string\glsp1\space since you haven't enabled entry counting}%
3085 \@glsp1@{#1}{#2}[#3]%
3086 }

```

`\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.

```

3087 \newcommand*{\cglsp1format}[2]{%
3088 \ifglshaslong{#1}{\glsp1entrylongpl{#1}}{\glsp1entryfirstplural{#1}}#2%
3089 }

```

`\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.)

```

3090 \newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}

```

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

```

3091 \newcommand*{\@cGlspl}[2] []{%
3092 \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2} []}%
3093 }

```



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

```
3094 \def\cGlspl@#1#2[#3]{%
3095 \GlossariesWarning{\string\cGlspl\space is defaulting to
3096 \string\Glspl\space since you haven't enabled entry counting}%
3097 \cGlspl@{#1}{#2}[#3]%
3098 }
```

`\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.

```
3099 \newcommand*{\cGlsplformat}[2]{%
3100 \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
3101 }
```

## 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.<sup>1</sup>

`\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`

```
3102 \newcommand*{\loadglsentries}[2][\@gls@default]{%
3103 \let\@gls@default\glsdefaulttype
3104 \def\glsdefaulttype{#1}\input{#2}%
3105 \let\glsdefaulttype\@gls@default
3106 }
```

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

```
3107 \@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

---

<sup>1</sup>and any other valid  $\TeX$  code that can be used in the preamble.

`\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`

```
3108 \newcommand*{\glstextformat}[1]{#1}
```

`\glentryfmt` As from version 3.11a, the way in which an entry is displayed is now governed by `\glentryfmt`. This doesn't take any arguments. The required information is set by commands like `\gls`. To ensure backward compatibility, the default use the old `\glsdisplay` and `\glsdisplayfirst` style of commands

```
3109 \newcommand*{\glentryfmt}{%
3110   \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
3111 }
```

Format that provides backwards compatibility:

```
3112 \newcommand*{\@@gls@default@entryfmt}[2]{%
3113   \ifdefempty\glscustomtext
3114   {%
3115     \glsifplural
3116     {%
```

Plural form

```
3117     \glscapscase
3118     {%
```

Don't adjust case

```
3119     \ifglsused\glslabel
3120     {%
```

Subsequent use

```
3121         #2{\glentryplural{\glslabel}}%
3122         {\glentrydescplural{\glslabel}}%
3123         {\glentrysymbolplural{\glslabel}}{\glsinsert}%
3124     }%
3125     {%
```

First use

```
3126         #1{\glentryfirstplural{\glslabel}}%
3127         {\glentrydescplural{\glslabel}}%
3128         {\glentrysymbolplural{\glslabel}}{\glsinsert}%
3129     }%
3130     }%
3131     {%
```

Make first letter upper case

```
3132     \ifglsused\glslabel
3133     {%
```

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 `\defglentryfmt`, which avoids the issues caused by fragile commands.)

```

3134 \ifbool{glscompatible-3.07}%
3135 {%
3136   \protected@edef\@glo@etext{%
3137     #2{\glsentryplural{\glslabel}}%
3138     {\glsentrydescplural{\glslabel}}%
3139     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3140   \xmakefirstuc\@glo@etext
3141 }%
3142 {%
3143   #2{\Glsentryplural{\glslabel}}%
3144   {\glsentrydescplural{\glslabel}}%
3145   {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
3146 }%
3147 }%
3148 {%

```

#### First use

```

3149 \ifbool{glscompatible-3.07}%
3150 {%
3151   \protected@edef\@glo@etext{%
3152     #1{\glsentryfirstplural{\glslabel}}%
3153     {\glsentrydescplural{\glslabel}}%
3154     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3155   \xmakefirstuc\@glo@etext
3156 }%
3157 {%
3158   #1{\Glsentryfirstplural{\glslabel}}%
3159   {\glsentrydescplural{\glslabel}}%
3160   {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
3161 }%
3162 }%
3163 }%
3164 {%

```

#### Make all upper case

```

3165 \ifglsused\glslabel
3166 {%

```

#### Subsequent use

```

3167 \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
3168 {\glsentrydescplural{\glslabel}}%
3169 {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3170 }%
3171 {%

```

#### First use

```

3172 \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
3173 {\glsentrydescplural{\glslabel}}%
3174 {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
3175 }%
3176 }%

```

```

3177 }%
3178 {%

```

#### Singular form

```

3179 \glscapscase
3180 {%

```

#### Don't adjust case

```

3181 \ifglused\glslabel
3182 {%

```

#### Subsequent use

```

3183 #2{\glentrytext{\glslabel}}%
3184 {\glentrydesc{\glslabel}}%
3185 {\glentrysymbol{\glslabel}}{\glinsert}%
3186 }%
3187 {%

```

#### First use

```

3188 #1{\glentryfirst{\glslabel}}%
3189 {\glentrydesc{\glslabel}}%
3190 {\glentrysymbol{\glslabel}}{\glinsert}%
3191 }%
3192 }%
3193 {%

```

#### Make first letter upper case

```

3194 \ifglused\glslabel
3195 {%

```

#### Subsequent use

```

3196 \ifbool{glcompatible-3.07}%
3197 {%
3198 \protected@edef\@glo@etext{%
3199 #2{\glentrytext{\glslabel}}%
3200 {\glentrydesc{\glslabel}}%
3201 {\glentrysymbol{\glslabel}}{\glinsert}}%
3202 \xmakefirstuc\@glo@etext
3203 }%
3204 {%
3205 #2{\Glsentrytext{\glslabel}}%
3206 {\glentrydesc{\glslabel}}%
3207 {\glentrysymbol{\glslabel}}{\glinsert}%
3208 }%
3209 }%
3210 {%

```

#### First use

```

3211 \ifbool{glcompatible-3.07}%
3212 {%
3213 \protected@edef\@glo@etext{%
3214 #1{\glentryfirst{\glslabel}}%

```

```

3215          {\glsentrydesc{\glslabel}}}%
3216          {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3217      \xmakefirstuc\@glo@etext
3218  }%
3219  {%
3220      #1{\Glsentryfirst{\glslabel}}}%
3221      {\glsentrydesc{\glslabel}}}%
3222      {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3223  }%
3224  }%
3225  }%
3226  {%

    Make all upper case
3227      \ifglsused\glslabel
3228      {%

        Subsequent use
3229          \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}}%
3230          {\glsentrydesc{\glslabel}}}%
3231          {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3232      }%
3233      {%

        First use
3234          \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}}%
3235          {\glsentrydesc{\glslabel}}}%
3236          {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
3237      }%
3238      }%
3239      }%
3240      }%
3241      {%

        Custom text provided in \glsdisp
3242      \ifglsused{\glslabel}%
3243      {%

        Subsequent use
3244          #2{\glscustomtext}%
3245          {\glsentrydesc{\glslabel}}}%
3246          {\glsentrysymbol{\glslabel}}{}%
3247      }%
3248      {%

        First use
3249          #1{\glscustomtext}%
3250          {\glsentrydesc{\glslabel}}}%
3251          {\glsentrysymbol{\glslabel}}{}%
3252      }%
3253      }%
3254  }

```

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

```
3255 \newcommand*{\glsgetentryfmt}{%
3256   \ifdefempty\glscustomtext
3257   {%
3258     \glsifplural
3259     {%
```

Plural form

```
3260     \glscapscase
3261     {%
```

Don't adjust case

```
3262     \ifglsused\glslabel
3263     {%
```

Subsequent use

```
3264     \glsentryplural{\glslabel}\glsinsert
3265     }%
3266     {%
```

First use

```
3267     \glsentryfirstplural{\glslabel}\glsinsert
3268     }%
3269     }%
3270     {%
```

Make first letter upper case

```
3271     \ifglsused\glslabel
3272     {%
```

Subsequent use.

```
3273     \Glsentryplural{\glslabel}\glsinsert
3274     }%
3275     {%
```

First use

```
3276     \Glsentryfirstplural{\glslabel}\glsinsert
3277     }%
3278     }%
3279     {%
```

Make all upper case

```
3280     \ifglsused\glslabel
3281     {%
```

Subsequent use

```
3282     \mfirstucMakeUppercase
3283     {\glsentryplural{\glslabel}\glsinsert}%
3284     }%
3285     {%
```

#### First use

```
3286      \mfirstucMakeUppercase
3287      {\glentryfirstplural{\glslabel}\glsinsert}%
3288      }%
3289      }%
3290      }%
3291      {%
```

#### Singular form

```
3292      \glscapscase
3293      {%
```

#### Don't adjust case

```
3294      \ifglsused\glslabel
3295      {%
```

#### Subsequent use

```
3296      \glentrytext{\glslabel}\glsinsert
3297      }%
3298      {%
```

#### First use

```
3299      \glentryfirst{\glslabel}\glsinsert
3300      }%
3301      }%
3302      {%
```

#### Make first letter upper case

```
3303      \ifglsused\glslabel
3304      {%
```

#### Subsequent use

```
3305      \Glsentrytext{\glslabel}\glsinsert
3306      }%
3307      {%
```

#### First use

```
3308      \Glsentryfirst{\glslabel}\glsinsert
3309      }%
3310      }%
3311      {%
```

#### Make all upper case

```
3312      \ifglsused\glslabel
3313      {%
```

#### Subsequent use

```
3314      \mfirstucMakeUppercase{\glentrytext{\glslabel}\glsinsert}%
3315      }%
3316      {%
```

#### First use

```
3317      \mfirstucMakeUppercase{\glentryfirst{\glslabel}\glsinsert}%

```

```

3318     }%
3319     }%
3320     }%
3321     }%
3322     {%

```

Custom text provided in `\glsdisp`. (The insert is most likely to be empty at this point.)

```

3323     \glscustomtext\glsinsert
3324     }%
3325 }

```

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

```

3326 \newcommand*{\glsgenacfmt}{%
3327   \ifdefempty\glscustomtext
3328   {%
3329     \ifglsused\glslabel
3330     {%

```

Subsequent use:

```

3331     \glsifplural
3332     {%

```

Subsequent plural form:

```

3333     \glscapscase
3334     {%

```

Subsequent plural form, don't adjust case:

```

3335     \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
3336     }%
3337     {%

```

Subsequent plural form, make first letter upper case:

```

3338     \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
3339     }%
3340     {%

```

Subsequent plural form, all caps:

```

3341     \mfirstucMakeUppercase
3342     {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
3343     }%
3344     }%
3345     {%

```

Subsequent singular form

```

3346     \glscapscase
3347     {%

```

Subsequent singular form, don't adjust case:

```

3348     \acronymfont{\glsentryshort{\glslabel}}\glsinsert
3349     }%
3350     {%

```



Subsequent singular form, make first letter upper case:

```
3351      \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
3352      }%
3353      {%
```

Subsequent singular form, all caps:

```
3354      \mfirstucMakeUppercase
3355      {\acronymfont{\Glsentryshort{\glslabel}}\glsinsert}%
3356      }%
3357      }%
3358      }%
3359      {%
```

First use:

```
3360      \glsifplural
3361      {%
```

First use plural form:

```
3362      \glscapscase
3363      {%
```

First use plural form, don't adjust case:

```
3364      \genplacrfullformat{\glslabel}{\glsinsert}%
3365      }%
3366      {%
```

First use plural form, make first letter upper case:

```
3367      \Genplacrfullformat{\glslabel}{\glsinsert}%
3368      }%
3369      {%
```

First use plural form, all caps:

```
3370      \mfirstucMakeUppercase
3371      {\genplacrfullformat{\glslabel}{\glsinsert}}%
3372      }%
3373      }%
3374      {%
```

First use singular form

```
3375      \glscapscase
3376      {%
```

First use singular form, don't adjust case:

```
3377      \genacrfullformat{\glslabel}{\glsinsert}%
3378      }%
3379      {%
```

First use singular form, make first letter upper case:

```
3380      \Genacrfullformat{\glslabel}{\glsinsert}%
3381      }%
3382      {%
```

First use singular form, all caps:

```

3383         \mfirstucMakeUppercase
3384         {\genacrfullformat{\glslabel}{\glsinsert}}%
3385     }%
3386 }%
3387 }%
3388 }%
3389 {%

```

User supplied text.

```

3390     \glscustomtext
3391 }%
3392 }

```

enacrfullformat

```
\genacrfullformat{\label}{\insert}
```

The full format used by \glsacrfmt (singular).

```

3393 \newcommand*{\genacrfullformat}[2]{%
3394     \glsentrylong{#1}#2\space
3395     (\protect\firstacronymfont{\glsentryshort{#1}})%
3396 }

```

enacrfullformat

```
\Genacrfullformat{\label}{\insert}
```

As above but makes the first letter upper case.

```

3397 \newcommand*{\Genacrfullformat}[2]{%
3398     \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
3399     \xmakefirstuc\gls@text
3400 }

```

placrfullformat

```
\genplacrfullformat{\label}{\insert}
```

The full format used by \glsacrfmt (plural).

```

3401 \newcommand*{\genplacrfullformat}[2]{%
3402     \glsentrylongpl{#1}#2\space
3403     (\protect\firstacronymfont{\glsentryshortpl{#1}})%
3404 }

```

placrfullformat

```
\Genplacrfullformat{\label}{\insert}
```

As above but makes the first letter upper case.

```

3405 \newcommand*{\Genplacrfullformat}[2]{%
3406     \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
3407     \xmakefirstuc\gls@text
3408 }

```

glsdisplayfirst   Deprecated. Kept for backward compatibility.

```
3409 \newcommand*{\glsdisplayfirst}[4]{#1#4}
```

\glsdisplay   Deprecated. Kept for backward compatibility.

```
3410 \newcommand*{\glsdisplay}[4]{#1#4}
```

\defglsdisplay   Deprecated. Kept for backward compatibility.

```
3411 \newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
3412   \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
3413   Use \string\defglsentryfmt\space instead}%
3414   \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
3415   \edef\@gls@doentrydef{%
3416     \noexpand\defglsentryfmt[#1]{%
3417       \noexpand\ifcsdef{gls@#1@displayfirst}%
3418       {%
3419         \noexpand\@@gls@default@entryfmt
3420         {\noexpand\csuse{gls@#1@displayfirst}}}%
3421       {\noexpand\csuse{gls@#1@display}}}%
3422     }%
3423     {%
3424       \noexpand\@@gls@default@entryfmt
3425       {\noexpand\glsdisplayfirst}%
3426       {\noexpand\csuse{gls@#1@display}}}%
3427     }%
3428   }%
3429 }%
3430 \@gls@doentrydef
3431 }
```

glsdisplayfirst   Deprecated. Kept for backward compatibility.

```
3432 \newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
3433   \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
3434   Use \string\defglsentryfmt\space instead}%
3435   \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
3436   \edef\@gls@doentrydef{%
3437     \noexpand\defglsentryfmt[#1]{%
3438       \noexpand\ifcsdef{gls@#1@display}%
3439       {%
3440         \noexpand\@@gls@default@entryfmt
3441         {\noexpand\csuse{gls@#1@displayfirst}}}%
3442       {\noexpand\csuse{gls@#1@display}}}%
3443     }%
3444     {%
3445       \noexpand\@@gls@default@entryfmt
3446       {\noexpand\csuse{gls@#1@displayfirst}}}%
3447       {\noexpand\glsdisplay}%
3448     }%
3449   }%
```

```

3450 }%
3451 \@gls@doentrydef
3452 }

```

## 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.

```

3453 \define@key{glslink}{counter}{%
3454   \ifcsundef{c#1}%
3455   {%
3456     \PackageError{glossaries}%
3457     {There is no counter called ‘#1’}%
3458     {%
3459       The counter key should have the name of a valid counter
3460       as its value%
3461     }%
3462   }%
3463   {%
3464     \def\@gls@counter{#1}%
3465   }%
3466 }

```

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.

```

3467 \define@key{glslink}{format}{%
3468   \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.

```

3469 \define@boolkey{glslink}{hyper}[true]{}

```

Initialise hyper key.

```

3470 \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.

```

3471 \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.

```
3472 \newcommand*{\glslinkvar}[3]{#1}
```

`\glsifhyper` Now deprecated.

```
3473 \newcommand*{\glsifhyper}[2]{%
3474   \glslinkvar{#1}{#2}{#1}%
3475   \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
3476     you mean \string\glsifhyperon\space or \string\glslinkvar?}%
3477 }
```

`\@gls@hyp@opt` Used by the commands such as `\glslink` to determine whether to modify the hyper option.

```
3478 \newcommand*{\@gls@hyp@opt}[1]{%
3479   \let\glslinkvar\@firstofthree
3480   \let\@gls@hyp@opt@cs#1\relax
3481   \@ifstar{\s@gls@hyp@opt}%
3482   {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
3483 }
```

`\s@gls@hyp@opt` Starred version

```
3484 \newcommand*{\s@gls@hyp@opt}[1][]{%
3485   \let\glslinkvar\@secondofthree
3486   \@gls@hyp@opt@cs[hyper=false,#1]}
```

`\p@gls@hyp@opt` Plus version

```
3487 \newcommand*{\p@gls@hyp@opt}[1][]{%
3488   \let\glslinkvar\@thirdofthree
3489   \@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*[\langle options \rangle]{\langle label \rangle}{\langle text \rangle}`

which is equivalent to `\glslink[hyper=false,\langle options \rangle]{\langle label \rangle}{\langle text \rangle}`

First determine which version is being used:

`\glslink`

```
3490 \newrobustcmd*{\glslink}{%
3491   \@gls@hyp@opt\@gls@link
3492 }
```

`\@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.

```
3493 \newcommand*{\@gls@link}[3] [] {%
3494   \glsdoifexistsordo{#2}%
3495   {%
3496     \let\do@gls@link@checkfirsthyper\relax
3497     \@gls@link[#1]{#2}{#3}%
3498   }{%
```

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

```
3499   \glstextformat{#3}%
3500   }%

3501   \glspostlinkhook
3502 }
```

`glspostlinkhook`

```
3503 \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 acrf footnote setting is on.) This assumes the glossary type is stored in `\glstype` and the label is stored in `\glslabel`.

```
3504 \newcommand*{\@gls@link@checkfirsthyper}{%
3505   \ifglsused{\glslabel}%
3506   {%
3507   }%
3508   {%
3509     \gls@checkisacronymlist\glstype
3510     \ifglshyperfirst
3511       \ifglsisacronymlist
3512         \ifglsacrfootnote
3513           \KV@glslink@hyperfalse
3514         \fi
3515       \fi
3516     \else
3517       \KV@glslink@hyperfalse
3518     \fi
3519   }%
```

Allow user to hook into this

```

3520 \glslinkcheckfirsthyperhook
3521 }

```

hookfirsthyperhook Allow used to hook into the \@gls@link@checkfirsthyper macro

```

3522 \newcommand*{\glslinkcheckfirsthyperhook}{}

```

linkpostsetkeys

```

3523 \newcommand*{\glslinkpostsetkeys}{}

```

\glsifhyperon Check the value of the hyper key:

```

3524 \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}

```

disablehyperinlist Disable hyperlink if in the “nohyper” list.

```

3525 \newcommand*{\do@glssdisablehyperinlist}{%
3526 \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
3527 {\KV@glslink@hyperfalse}}{}%
3528 }

```

let@glslink@opts Hook to set default options for \@glslink.

```

3529 \newcommand*{\@gls@setdefault@glslink@opts}{}

```

\@gls@link

```

3530 \def\@gls@link[#1]#2#3{%

```

Inserting \leavevmode suggested by Donald Arseneau (avoids problem with tabularx).

```

3531 \leavevmode
3532 \protected@edef\glslabel{\glsdetoklabel{#2}}%

```

Save options in \@gls@link@opts and label in \@gls@link@label

```

3533 \def\@gls@link@opts{#1}%
3534 \let\@gls@link@label\glslabel

```

```

3535 \def\@glsnumberformat{glsnumberformat}%
3536 \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%

```

If this is in one of the “nohypertypes” glossaries, suppress the hyperlink by default

```

3537 \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%

```

Save original setting

```

3538 \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper

```

Set defaults:

```

3539 \@gls@setdefault@glslink@opts

```

Switch off hyper setting if the glossary type has been identified in nohyperlist.

```

3540 \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.

```

3541 \do@gls@link@checkfirsthyper
3542 \setkeys{glslink}{#1}%

```

Add a hook for the user to customise things after the keys have been set.

```

3543   \glslinkpostsetkeys
      Store the entry's counter in \theglsentrycounter
3544   \@gls@saveentrycounter
      Define sort key if necessary:
3545   \@gls@setsort{\glslabel}%
      (De-tok'ing done by \@do@wrglossary)
3546   \@do@wrglossary{#2}%
3547   \ifKV@glslink@hyper
3548     \@glslink{\glslinkprefix\glslabel}{\glstextformat{#3}}%
3549   \else
3550     \glsdonohyperlink{\glslinkprefix\glslabel}{\glstextformat{#3}}%
3551   \fi
      Restore original setting
3552   \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
3553 }
```

\glslinkprefix

```

3554 \newcommand*{\glslinkprefix}{glo:}
```

glsentrycounter Set default value of entry counter

```

3555 \def\glsentrycounter{\glscounter}%
```

saveentrycounter Need to check if using equation counter in align environment:

```

3556 \newcommand*{\@gls@saveentrycounter}{%
3557   \def\@gls@Hcounter{}}%
```

Are we using equation counter?

```

3558 \ifthenelse{\equal{\@gls@counter}{equation}}{%
3559   {
```

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

```

3560   \ifcsundef{xatlevel@}%
3561   {%
3562     \edef\theglsentrycounter{\expandafter\noexpand
3563       \csname the\@gls@counter\endcsname}%
3564   }%
3565   {%
3566     \ifx\xatlevel@\@empty
3567       \edef\theglsentrycounter{\expandafter\noexpand
3568         \csname the\@gls@counter\endcsname}%
3569     \else
3570       \savecounters@
3571       \advance\c@equation by 1\relax
3572       \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%

```



Check if hyperref version of this counter

```

3573     \ifcsundef{theH\@gls@counter}%
3574     {%
3575         \def\@gls@Hcounter{\theglsentrycounter}%
3576     }%
3577     {%
3578         \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
3579     }%
3580     \protected@edef\theHglentrycounter{\@gls@Hcounter}%
3581     \restorecounters@
3582 \fi
3583 }%
3584 }%
3585 {%

```

Not using equation counter so no special measures:

```

3586     \edef\theglsentrycounter{\expandafter\noexpand
3587         \csname the\@gls@counter\endcsname}%
3588 }%

```

Check if hyperref version of this counter

```

3589 \ifx\@gls@Hcounter\@empty
3590     \ifcsundef{theH\@gls@counter}%
3591     {%
3592         \def\theHglentrycounter{\theglsentrycounter}%
3593     }%
3594     {%
3595         \protected@edef\theHglentrycounter{\expandafter\noexpand
3596             \csname theH\@gls@counter\endcsname}%
3597     }%
3598 \fi
3599 }

```

`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.

```

3600 \def\@set@glo@numformat#1#2#3#4{%
3601     \expandafter\@glo@check@mkidxrangechar#3\@nil
3602     \protected@edef#1{%
3603         \@glo@prefix setentrycounter[#4]{#2}%
3604         \expandafter\string\csname\@glo@suffix\endcsname
3605     }%
3606     \@gls@checkmkidxchars#1%
3607 }

```

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.

```

3608 \def\@glo@check@mkidxrangechar#1#2\@nil{%
3609 \if#1(\relax
3610   \def\@glo@prefix{()%
3611   \if\relax#2\relax
3612     \def\@glo@suffix{glsnumberformat}%
3613   \else
3614     \def\@glo@suffix{#2}%
3615   \fi
3616 \else
3617   \if#1)\relax
3618     \def\@glo@prefix{}}}%
3619   \if\relax#2\relax
3620     \def\@glo@suffix{glsnumberformat}%
3621   \else
3622     \def\@glo@suffix{#2}%
3623   \fi
3624 \else
3625   \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
3626 \fi
3627 \fi}

```

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

```

3628 \newcommand*{\@gls@escbsdq}[1]{%
3629   \def\@gls@checkedmkidx{%
3630     \let\gls@xdystring=#1\relax
3631     \@onelevel@sanitize\gls@xdystring
3632     \edef\do@gls@xdycheckbackslash{%
3633       \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
3634       \@backslashchar\@backslashchar\noexpand\null}%
3635     \do@gls@xdycheckbackslash
3636     \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
3637     \def\@gls@checkedmkidx{%
3638       \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
3639       \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%

```

Unsanitize\gls@numberpage,\gls@alphpage,\gls@Alphpage and\gls@romanpage (thanks to David Carlisle for the suggestion.)

```

3640   \@for\@gls@tmp:=\gls@protected@pagefmts\do
3641   {%
3642     \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
3643     \@onelevel@sanitize\@gls@sanitized@tmp
3644     \edef\gls@dosubst{%
3645       \noexpand\DTLsubstituteall\noexpand\gls@xdystring
3646       {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
3647     }%
3648     \gls@dosubst
3649   }%

```

Assign to required control sequence

```

3650   \let#1=\gls@xdystring

```

3651 }

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

checkmkidxchars

```
3652 \newcommand{\@gls@checkmkidxchars}[1]{%
3653   \ifglxsindy
3654     \@gls@escbsdq{#1}%
3655   \else
3656     \def\@gls@checkedmkidx{%
3657       \expandafter\@gls@checkquote#1\@nil""\null
3658       \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3659       \def\@gls@checkedmkidx{%
3660         \expandafter\@gls@checkescquote#1\@nil\""\null
3661         \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3662         \def\@gls@checkedmkidx{%
3663           \expandafter\@gls@checkescactual#1\@nil\??\null
3664           \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3665           \def\@gls@checkedmkidx{%
3666             \expandafter\@gls@checkactual#1\@nil??\null
3667             \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3668             \def\@gls@checkedmkidx{%
3669               \expandafter\@gls@checkbar#1\@nil||\null
3670               \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3671               \def\@gls@checkedmkidx{%
3672                 \expandafter\@gls@checkescbar#1\@nil\\|\null
3673                 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3674                 \def\@gls@checkedmkidx{%
3675                   \expandafter\@gls@checklevel#1\@nil!!\null
3676                   \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3677                   \def\@gls@checkedmkidx{%
3678                     \expandafter\@gls@checkesclevel#1\@nil\!|\null
3679                     \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3680                     \def\@gls@checkedmkidx{%
3681                       \edef\@do@gls@checkesc{%
3682                         \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
3683                         \@backslashchar\@backslashchar\noexpand\null
3684                       }%
3685                       \@do@gls@checkesc
3686                       \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
3687                     \fi
3688                   }
```

Update the control sequence and strip trailing \@nil:

s@updatechecked

```
3689 \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
```

\@gls@tmpb Define temporary token

```
3690 \newtoks\@gls@tmpb
```

@gls@checkquote Replace " with "" since " is a makeindex special character.

```
3691 \def\@gls@checkquote#1"#2"#3\null{%
3692   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3693   \toks@={#1}%
3694   \ifx\null#2\null
3695     \ifx\null#3\null
3696       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3697       \def\@gls@checkquote{\relax}%
3698     \else
3699       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3700         \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
3701       \def\@gls@checkquote{\@gls@checkquote#3\null}%
3702     \fi
3703   \else
3704     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3705       \@gls@quotechar\@gls@quotechar}%
3706     \ifx\null#3\null
3707       \def\@gls@checkquote{\@gls@checkquote#2""\null}%
3708     \else
3709       \def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
3710     \fi
3711   \fi
3712   \@gls@checkquote
3713 }
```

s@checkescquote Do the same for \":

```
3714 \def\@gls@checkescquote#1\"#2\"#3\null{%
3715   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3716   \toks@={#1}%
3717   \ifx\null#2\null
3718     \ifx\null#3\null
3719       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3720       \def\@gls@checkescquote{\relax}%
3721     \else
3722       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3723         \@gls@quotechar\string\"@gls@quotechar
3724         \@gls@quotechar\string\"@gls@quotechar}%
3725       \def\@gls@checkescquote{\@gls@checkescquote#3\null}%
3726     \fi
3727   \else
3728     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3729       \@gls@quotechar\string\"@gls@quotechar}%
3730     \ifx\null#3\null
3731       \def\@gls@checkescquote{\@gls@checkescquote#2\"\" \null}%
3732     \else
3733       \def\@gls@checkescquote{\@gls@checkescquote#2\"#3\null}%
3734     \fi
3735   \fi
3736   \@gls@checkescquote
```

3737 }

@checkescactual Similarly for \? (which is replaces @ as makeindex's special character):

```
3738 \def\@gls@checkescactual#1\?#2\?#3\null{%
3739 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3740 \toks@={#1}%
3741 \ifx\null#2\null
3742 \ifx\null#3\null
3743 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3744 \def\@gls@checkescactual{\relax}%
3745 \else
3746 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3747 \@gls@quotechar\string\" \@gls@actualchar
3748 \@gls@quotechar\string\" \@gls@actualchar}%
3749 \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
3750 \fi
3751 \else
3752 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3753 \@gls@quotechar\string\" \@gls@actualchar}%
3754 \ifx\null#3\null
3755 \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
3756 \else
3757 \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
3758 \fi
3759 \fi
3760 \@gls@checkescactual
3761 }
```

gls@checkeschar Similarly for \||:

```
3762 \def\@gls@checkeschar#1\||#2\||#3\null{%
3763 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3764 \toks@={#1}%
3765 \ifx\null#2\null
3766 \ifx\null#3\null
3767 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3768 \def\@gls@checkeschar{\relax}%
3769 \else
3770 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3771 \@gls@quotechar\string\" \@gls@encapchar
3772 \@gls@quotechar\string\" \@gls@encapchar}%
3773 \def\@gls@checkeschar{\@gls@checkeschar#3\null}%
3774 \fi
3775 \else
3776 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3777 \@gls@quotechar\string\" \@gls@encapchar}%
3778 \ifx\null#3\null
3779 \def\@gls@checkeschar{\@gls@checkeschar#2\||\|\null}%
3780 \else
3781 \def\@gls@checkeschar{\@gls@checkeschar#2\||#3\null}%

```

```

3782 \fi
3783 \fi
3784 \@gls@checkescbar
3785 }

```

s@checkesclevel Similarly for \!:

```

3786 \def\@gls@checkesclevel#1\!#2\!#3\null{%
3787 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3788 \toks@={#1}%
3789 \ifx\null#2\null
3790 \ifx\null#3\null
3791 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3792 \def\@gls@checkesclevel{\relax}%
3793 \else
3794 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3795 \@gls@quotechar\string"\@gls@levelchar
3796 \@gls@quotechar\string"\@gls@levelchar}%
3797 \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
3798 \fi
3799 \else
3800 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3801 \@gls@quotechar\string"\@gls@levelchar}%
3802 \ifx\null#3\null
3803 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
3804 \else
3805 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
3806 \fi
3807 \fi
3808 \@gls@checkesclevel
3809 }

```

\@gls@checkbar and for |:

```

3810 \def\@gls@checkbar#1|#2|#3\null{%
3811 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3812 \toks@={#1}%
3813 \ifx\null#2\null
3814 \ifx\null#3\null
3815 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3816 \def\@gls@checkbar{\relax}%
3817 \else
3818 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3819 \@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
3820 \def\@gls@checkbar{\@gls@checkbar#3\null}%
3821 \fi
3822 \else
3823 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3824 \@gls@quotechar\@gls@encapchar}%
3825 \ifx\null#3\null
3826 \def\@gls@checkbar{\@gls@checkbar#2||\null}%

```

```

3827 \else
3828 \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
3829 \fi
3830 \fi
3831 \@gls@checkbar
3832 }

```

**@gls@checklevel** and for !:

```

3833 \def\@gls@checklevel#1!#2!#3\null{%
3834 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3835 \toks@={#1}%
3836 \ifx\null#2\null
3837 \ifx\null#3\null
3838 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3839 \def\@gls@checklevel{\relax}%
3840 \else
3841 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3842 \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
3843 \def\@gls@checklevel{\@gls@checklevel#3\null}%
3844 \fi
3845 \else
3846 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3847 \@gls@quotechar\@gls@levelchar}%
3848 \ifx\null#3\null
3849 \def\@gls@checklevel{\@gls@checklevel#2!!\null}%
3850 \else
3851 \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
3852 \fi
3853 \fi
3854 \@gls@checklevel
3855 }

```

**gls@checkactual** and for ?:

```

3856 \def\@gls@checkactual#1?#2?#3\null{%
3857 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3858 \toks@={#1}%
3859 \ifx\null#2\null
3860 \ifx\null#3\null
3861 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3862 \def\@gls@checkactual{\relax}%
3863 \else
3864 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3865 \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
3866 \def\@gls@checkactual{\@gls@checkactual#3\null}%
3867 \fi
3868 \else
3869 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3870 \@gls@quotechar\@gls@actualchar}%
3871 \ifx\null#3\null

```

```

3872     \def\@gls@checkactual{\@gls@checkactual#2??\null}%
3873     \else
3874     \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
3875     \fi
3876     \fi
3877     \@gls@checkactual
3878 }

```

\@gls@checkesc and for \:

```

3879 \bgroup
3880 \catcode'\|0\relax
3881 \catcode'\|12\relax
3882 |gdef|@gls@checkesc#1\#2\#3|null{%
3883   |@gls@tmpb=|expandafter{|@gls@checkedmkidx}%
3884   |toks@={#1}%
3885   |ifx|null#2|null
3886     |ifx|null#3|null
3887     |edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@}%
3888     |def|@gls@checkesc{|relax}%
3889     |else
3890     |edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
3891       |@gls@quotechar|@backslashchar|@gls@quotechar|@backslashchar}%
3892     |def|@gls@checkesc{|@gls@checkesc#3|null}%
3893     |fi
3894     |else
3895     |edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
3896       |@gls@quotechar|@backslashchar}%
3897     |ifx|null#3|null
3898     |def|@gls@checkesc{|@gls@checkesc#2\\|null}%
3899     |else
3900     |def|@gls@checkesc{|@gls@checkesc#2\#3|null}%
3901     |fi
3902     |fi
3903     |@gls@checkesc
3904 }
3905 |egroup

```

s@xdycheckquote As before but for use with xindy

```

3906 \def\@gls@xdycheckquote#1"#2"#3|null{%
3907   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
3908   \toks@={#1}%
3909   \ifx\|null#2|null
3910     \ifx\|null#3|null
3911       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
3912       \def\@gls@xdycheckquote{\relax}%
3913     \else
3914       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3915         \string\|}\string\}%
3916       \def\@gls@xdycheckquote{\@gls@xdycheckquote#3|null}%

```



```

3917     \fi
3918   \else
3919     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
3920       \string\}%
3921     \ifx\null#3\null
3922       \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
3923     \else
3924       \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
3925     \fi
3926   \fi
3927 \@@gls@xdycheckquote
3928 }

```

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

```

3929 \edef\def\@gls@xdycheckbackslash{%
3930   \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
3931     ##2\@backslashchar##3\noexpand\null{%
3932     \noexpand\@gls@tmpb=\noexpand\expandafter
3933       {\noexpand\@gls@checkedmkidx}%
3934     \noexpand\toks@={##1}%
3935     \noexpand\ifx\noexpand\null##2\noexpand\null
3936     \noexpand\ifx\noexpand\null##3\noexpand\null
3937     \noexpand\edef\noexpand\@gls@checkedmkidx{%
3938       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
3939     \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
3940     \noexpand\else
3941     \noexpand\edef\noexpand\@gls@checkedmkidx{%
3942       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
3943       \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
3944     \noexpand\def\noexpand\@gls@xdycheckbackslash{%
3945       \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
3946     \noexpand\fi
3947     \noexpand\else
3948     \noexpand\edef\noexpand\@gls@checkedmkidx{%
3949       \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
3950       \@backslashchar\@backslashchar}%
3951     \noexpand\ifx\noexpand\null##3\noexpand\null
3952     \noexpand\def\noexpand\@gls@xdycheckbackslash{%
3953       \noexpand\@gls@xdycheckbackslash##2\@backslashchar
3954       \@backslashchar\noexpand\null}%
3955     \noexpand\else
3956     \noexpand\def\noexpand\@gls@xdycheckbackslash{%
3957       \noexpand\@gls@xdycheckbackslash##2\@backslashchar
3958       ##3\noexpand\null}%
3959     \noexpand\fi
3960   \noexpand\fi
3961   \noexpand\@gls@xdycheckbackslash
3962 }%
3963 }

```

Now go ahead and define \@gls@xdycheckbackslash

```
3964 \def@gls@xdycheckbackslash
```

lsdohypertarget

```
3965 \newlength\gls@tmplen
3966 \newcommand*{\glsdohypertarget}[2]{%
3967   \@glsshowtarget{#1}%
3968   \settoheight{\gls@tmplen}{#2}%
3969   \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
3970 }
```

\glsdohyperlink

```
3971 \newcommand*{\glsdohyperlink}[2]{%
3972   \@glsshowtarget{#1}%
3973   \hyperlink{#1}{#2}%
3974 }
```

lsdonohyperlink

```
3975 \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.

```
3976 \ifcsundef{hyperlink}%
3977 {%
3978   \let\@glslink\glsdonohyperlink
3979 }%
3980 {%
3981   \let\@glslink\glsdohyperlink
3982 }
```

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

```
3983 \ifcsundef{hypertarget}%
3984 {%
3985   \let\@glstarget\@secondoftwo
3986 }%
3987 {%
3988   \let\@glstarget\glsdohypertarget
3989 }
```

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

glsdisablehyper

```
3990 \newcommand{\glsdisablehyper}{%
3991   \KV@glslink@hyperfalse
3992   \let\@glslink\glsdonohyperlink
3993   \let\@glstarget\@secondoftwo
3994 }
```

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

`\glsenablehyper`

```
3995 \newcommand{\glsenablehyper}{%
3996   \KV@glslink@hypertrue
3997   \let\@glslink\glsdohyperlink
3998   \let\@glstarget\glsdohypertarget
3999 }
```

Provide some convenience commands if not already defined:

```
4000 \providecommand{\@firstofthree}[3]{#1}
4001 \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`

```
4002 \newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
```

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

`\@gls`

```
4003 \newcommand*{\@gls}[2] []{%
4004   \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} []}%
4005 }
```

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

```
4006 \def\@gls@#1#2[#3]{%
4007   \glsdoifexists{#2}%
4008   {%
4009     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

4010     \let\glsifplural\@secondoftwo
4011     \let\glsapscase\@firstofthree
4012     \let\glscustomtext\@empty
4013     \def\glsinsert{#3}%
```

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

```
4014 \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.

```
4015 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4016 \ifKV@gls@link@local
```

```
4017 \glslocalunset{#2}%
```

```
4018 \else
```

```
4019 \glsunset{#2}%
```

```
4020 \fi
```

```
4021 }%
```

```
4022 \gls@postlinkhook
```

```
4023 }
```

\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

```
4024 \newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
```

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

```
4025 \newcommand*{\@Gls}[2] [] {%
```

```
4026 \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}%
```

```
4027 }
```

\@Gls@ Read in the final optional argument:

```
4028 \def\@Gls@#1#2[#3] {%
```

```
4029 \glsdoifexists{#2}%
```

```
4030 {%
```

```
4031 \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
4032 \let\glsifplural\@secondoftwo
```

```
4033 \let\gls@scapscase\@secondofthree
```

```
4034 \let\gls@customtext\@empty
```

```
4035 \def\glsinsert{#3}%
```

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

```
4036 \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.

```
4037 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4038 \ifKV@glslink@local
4039 \glsllocalunset{#2}%
4040 \else
4041 \glsunset{#2}%
4042 \fi
4043 }%

4044 \glspostlinkhook
4045 }
```

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

\GLS

```
4046 \newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
```

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

```
4047 \newcommand*{\@GLS}[2][\@GLS@{#1}{#2}]{\@GLS@{#1}{#2}[]}%
4048 \new@ifnextchar[\@GLS@{#1}{#2}]{\@GLS@{#1}{#2}[]}%
4049 }
```

\@GLS@ Read in the final optional argument:

```
4050 \def\@GLS@#1#2[#3]{%
4051 \glsdoidexists{#2}%
4052 {%
4053 \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

4054 \let\glsifplural\@secondoftwo
4055 \let\glscapscase\@thirdofthree
4056 \let\glscustomtext\@empty
4057 \def\glsinsert{#3}%
```

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

```
4058 \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.

```
4059 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4060 \ifKV@glslink@local
4061 \glsllocalunset{#2}%
4062 \else
4063 \glsunset{#2}%
4064 \fi
4065 }%

4066 \glspostlinkhook
4067 }
```

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

`\glspl`

```
4068 \newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
```

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

```
4069 \newcommand*{\@glspl}[2][\%  
4070 \new@ifnextchar[\@glspl@{#1}{#2}]{\@glspl@{#1}{#2}[]}%  
4071 }
```

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

```
4072 \def\@glspl@#1#2[#3]{%  
4073 \glsdoifexists{#2}%  
4074 {%  
4075 \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper  
  
4076 \let\glsifplural\@firstoftwo  
4077 \let\glsupscase\@firstofthree  
4078 \let\glscustomtext\@empty  
4079 \def\glsinsert{#3}%
```

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

```
4080 \def\@glo@text{\csname gls@\glstyp @entryfmt\endcsname}%
```

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

```
4081 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4082 \ifKV@glslink@local  
4083 \glslocalunset{#2}%  
4084 \else  
4085 \glsunset{#2}%  
4086 \fi  
4087 }%
```

```
4088 \glspostlinkhook  
4089 }
```

`\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`

```
4090 \newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
```

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

```
4091 \newcommand*{\@Glspl}[2][\%  
4092 \new@ifnextchar[\@Glspl@{#1}{#2}]{\@Glspl@{#1}{#2}[]}%  
4093 }
```

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

```
4094 \def\@Glspl@#1#2[#3]{%
4095   \glsdoifexists{#2}%
4096   {%
4097     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

4098     \let\glsifplural\@firstoftwo
4099     \let\glsapscase\@secondofthree
4100     \let\glscustomtext\@empty
4101     \def\glsinsert{#3}%
```

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

```
4102   \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

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

```
4103   \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4104   \ifKV@glslink@local
4105     \glslocalunset{#2}%
4106   \else
4107     \glsunset{#2}%
4108   \fi
4109 }%

4110 \glspostlinkhook
4111 }
```

`\GLSp1` behaves like `\glsp1` except that all the link text is converted to uppercase.

`\GLSp1`

```
4112 \newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
```

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

```
4113 \newcommand*{\@GLSp1}[2][{}]{%
4114   \new@ifnextchar[\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
4115 }
```

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

```
4116 \def\@GLSp1@#1#2[#3]{%
4117   \glsdoifexists{#2}%
4118   {%
4119     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

4120     \let\glsifplural\@firstoftwo
4121     \let\glsapscase\@thirdofthree
4122     \let\glscustomtext\@empty
4123     \def\glsinsert{#3}%
```

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

```
4124 \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.

```
4125 \@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
4126 \ifKV@gls@link@local
```

```
4127 \glslocalunset{#2}%
```

```
4128 \else
```

```
4129 \glsunset{#2}%
```

```
4130 \fi
```

```
4131 }%
```

```
4132 \gls@postlinkhook
```

```
4133 }
```

`\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:

```
4134 \newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
```

Defined the un-starred form.

`\@glsdisp`

```
4135 \newcommand*{\@glsdisp}[3] [] {%
```

```
4136 \glsdoifexists{#2}{%
```

```
4137 \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
4138 \let\glsifplural\@secondoftwo
```

```
4139 \let\gls@scapscase\@firstofthree
```

```
4140 \def\gls@customtext{#3}%
```

```
4141 \def\glsinsert{}%
```

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

```
4142 \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.

```
4143 \@gls@link[#1]{#2}{\@glo@text}%
```



Indicate that this entry has now been used

```
4144 \ifKV@glslink@local
4145 \glsllocalunset{#2}%
4146 \else
4147 \glunset{#2}%
4148 \fi
4149 }%

4150 \glspostlinkhook
4151 }
```

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.

```
4152 \newcommand*{\@gl@link@nocheckfirsthyper}{}
```

@gl@field@link

```
4153 \newcommand{\@gl@field@link}[3]{%
4154 \glsoifexists{#2}%
4155 {%
4156 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4157 \@gl@link[#1]{#2}{#3}%
4158 }%

4159 \glspostlinkhook
4160 }
```

\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

```
4161 \newrobustcmd*{\glstext}{\@gl@hyp@opt\@glstext}
```

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

```
4162 \newcommand*{\@glstext}[2][{}]{%
4163 \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4164 \def\@glstext@#1#2[#3]{%
4165 \@gl@field@link{#1}{#2}{\glstrytext{#2}#3}%
4166 }
```

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

\GLStext

```
4167 \newrobustcmd*{\GLStext}{\@gl@hyp@opt\@GLStext}
```

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

```
4168 \newcommand*{\@GLStext}[2][{}]{%
4169 \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4170 \def\@GLStext@#1#2[#3]{%
4171   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrytext{#2}#3}}%
4172 }
```

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

`\Glstext`

```
4173 \newrobustcmd*{\Glstext}{\@gls@hyp@opt\@GLstext}
```

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

```
4174 \newcommand*{\@GLstext}[2] [] {%
4175   \new@ifnextchar[{\@GLstext@{#1}{#2}}{\@GLstext@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4176 \def\@GLstext@#1#2[#3]{%
4177   \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
4178 }
```

`\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`

```
4179 \newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
```

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

```
4180 \newcommand*{\@glsfirst}[2] [] {%
4181   \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4182 \def\@glsfirst@#1#2[#3]{%
4183   \@gls@field@link{#1}{#2}{\glstentryfirst{#2}#3}%
4184 }
```

`\Glsfirst` behaves like `\glsfirst` except it displays the first letter in uppercase.

`\Glsfirst`

```
4185 \newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
```

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

```
4186 \newcommand*{\@Glsfirst}[2] [] {%
4187   \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4188 \def\@Glsfirst@#1#2[#3]{%
4189   \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
4190 }
```

`\GLSfirst` behaves like `\Glsfirst` except it displays the text in uppercase.

`\GLSfirst`

```
4191 \newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
```

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

```
4192 \newcommand*{\@GLSfirst}[2] [] {%  
4193   \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4194 \def\@GLSfirst@#1#2[#3] {%  
4195   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%  
4196 }
```

`\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`

```
4197 \newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
```

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

```
4198 \newcommand*{\@glsplural}[2] [] {%  
4199   \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4200 \def\@glsplural@#1#2[#3] {%  
4201   \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}}%  
4202 }
```

`\Glsplural` behaves like `\glsplural` except that the first letter is converted to uppercase.

`\Glsplural`

```
4203 \newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
```

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

```
4204 \newcommand*{\@Glsplural}[2] [] {%  
4205   \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4206 \def\@Glsplural@#1#2[#3] {%  
4207   \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}}%  
4208 }
```

`\GLSplural` behaves like `\glsplural` except that the text is converted to uppercase.

`\GLSplural`

```
4209 \newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
```

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

```
4210 \newcommand*{\@GLSplural}[2] [] {%  
4211   \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4212 \def\@GLSplural@#1#2[#3] {%  
4213   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%  
4214 }
```

`\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`

```
4215 \newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
```

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

```
4216 \newcommand*{\@glsfirstplural}[2] [] {%
```

```
4217   \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4218 \def\@glsfirstplural@#1#2[#3] {%
```

```
4219   \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
```

```
4220 }
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the first letter is converted to uppercase.

`\Glsfirstplural`

```
4221 \newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
```

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

```
4222 \newcommand*{\@Glsfirstplural}[2] [] {%
```

```
4223   \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4224 \def\@Glsfirstplural@#1#2[#3] {%
```

```
4225   \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
```

```
4226 }
```

`\GLSfirstplural` behaves like `\glsfirstplural` except that the link text is converted to uppercase.

`\GLSfirstplural`

```
4227 \newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
```

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

```
4228 \newcommand*{\@GLSfirstplural}[2] [] {%
```

```
4229   \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4230 \def\@GLSfirstplural@#1#2[#3] {%
```

```
4231   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
```

```
4232 }
```

`\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`

```
4233 \newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
```

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

```
4234 \newcommand*{\@glsname}[2] [] {%
```

```
4235   \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4236 \def\@glsname@#1#2[#3]{%
4237   \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
4238 }
```

\Glsname behaves like \glsname except that the first letter is converted to uppercase.

\Glsname

```
4239 \newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
```

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

```
4240 \newcommand*{\@Glsname}[2][{}]{%
4241   \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4242 \def\@Glsname@#1#2[#3]{%
4243   \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
4244 }
```

\GLSname behaves like \glsname except that the link text is converted to uppercase.

\GLSname

```
4245 \newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
```

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

```
4246 \newcommand*{\@GLSname}[2][{}]{%
4247   \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
4248 \def\@GLSname@#1#2[#3]{%
4249   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
4250 }
```

\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

```
4251 \newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
```

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

```
4252 \newcommand*{\@glsdesc}[2][{}]{%
4253   \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
4254 \def\@glsdesc@#1#2[#3]{%
4255   \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
4256 }
```

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

\Glsdesc

```
4257 \newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
```

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

```
4258 \newcommand*{\@GLSdesc}[2] [] {%  
4259   \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4260 \def\@GLSdesc@#1#2[#3] {%  
4261   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%  
4262 }
```

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

\GLSdesc

```
4263 \newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
```

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

```
4264 \newcommand*{\@GLSdesc}[2] [] {%  
4265   \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4266 \def\@GLSdesc@#1#2[#3] {%  
4267   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%  
4268 }
```

\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

```
4269 \newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
```

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

```
4270 \newcommand*{\@glsdescplural}[2] [] {%  
4271   \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4272 \def\@glsdescplural@#1#2[#3] {%  
4273   \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%  
4274 }
```

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

\Glsdescplural

```
4275 \newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
```

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

```
4276 \newcommand*{\@Glsdescplural}[2] [] {%  
4277   \new@ifnextchar [{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4278 \def\@Glsdescplural@#1#2[#3] {%  
4279   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%  
4280 }
```

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

`\GLSdescplural`

```
4281 \newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
```

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

```
4282 \newcommand*{\@GLSdescplural}[2] [] {%
```

```
4283   \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4284 \def\@GLSdescplural@#1#2[#3] {%
```

```
4285   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
```

```
4286 }
```

`\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`

```
4287 \newrobustcmd*{\glsymbol}{\@gls@hyp@opt\@glsymbol}
```

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

```
4288 \newcommand*{\@glsymbol}[2] [] {%
```

```
4289   \new@ifnextchar[{\@glsymbol@{#1}{#2}}{\@glsymbol@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4290 \def\@glsymbol@#1#2[#3] {%
```

```
4291   \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}}%
```

```
4292 }
```

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

`\Glsymbol`

```
4293 \newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
```

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

```
4294 \newcommand*{\@Glsymbol}[2] [] {%
```

```
4295   \new@ifnextchar[{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4296 \def\@Glsymbol@#1#2[#3] {%
```

```
4297   \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}}%
```

```
4298 }
```

`\GLSsymbol` behaves like `\glsymbol` except that the link text is converted to uppercase.

`\GLSsymbol`

```
4299 \newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
```

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

```
4300 \newcommand*{\@GLSsymbol}[2] [] {%
```

```
4301   \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]}
```

Read in the final optional argument:

```
4302 \def\@GLSsymbol@#1#2[#3]{%
4303   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbol{#2}#3}}%
4304 }
```

`\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`

```
4305 \newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
```

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

```
4306 \newcommand*{\@glsymbolplural}[2][\%]
4307   \new@ifnextchar[{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4308 \def\@glsymbolplural@#1#2[#3]{%
4309   \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
4310 }
```

`\Glsymbolplural` behaves like `\glsymbolplural` except that the first letter is converted to uppercase.

`Glsymbolplural`

```
4311 \newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
```

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

```
4312 \newcommand*{\@Glsymbolplural}[2][\%]
4313   \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4314 \def\@Glsymbolplural@#1#2[#3]{%
4315   \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
4316 }
```

`\GLSsymbolplural` behaves like `\glsymbolplural` except that the link text is converted to uppercase.

`GLSsymbolplural`

```
4317 \newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
```

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

```
4318 \newcommand*{\@GLSsymbolplural}[2][\%]
4319   \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4320 \def\@GLSsymbolplural@#1#2[#3]{%
4321   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}#3}}%
4322 }
```

`\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`

```
4323 \newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
```

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

```
4324 \newcommand*{\@glsuseri}[2] [] {%
```

```
4325   \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
4326 \def\@glsuseri@#1#2[#3]{%
```

```
4327   \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
```

```
4328 }
```

`\Glsuseri` behaves like `\glsuseri` except that the first letter is converted to uppercase.

`\Glsuseri`

```
4329 \newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
```

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

```
4330 \newcommand*{\@Glsuseri}[2] [] {%
```

```
4331   \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
4332 \def\@Glsuseri@#1#2[#3]{%
```

```
4333   \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
```

```
4334 }
```

`\GLSuseri` behaves like `\glsuseri` except that the link text is converted to uppercase.

`\GLSuseri`

```
4335 \newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
```

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

```
4336 \newcommand*{\@GLSuseri}[2] [] {%
```

```
4337   \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
4338 \def\@GLSuseri@#1#2[#3]{%
```

```
4339   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}%
```

```
4340 }
```

`\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`

```
4341 \newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
```

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

```
4342 \newcommand*{\@glsuserii}[2] [] {%
```

```
4343   \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}}
```

Read in the final optional argument:

```
4344 \def\@glsuserii@#1#2[#3]{%
```

```
4345   \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
```

```
4346 }
```

\Glsuserii behaves like \glsuserii except that the first letter is converted to uppercase.

\Glsuserii

```
4347 \newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
```

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

```
4348 \newcommand*{\@Glsuserii}[2] [] {%
```

```
4349   \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4350 \def\@Glsuserii@#1#2[#3] {%
```

```
4351   \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
```

```
4352 }
```

\GLSuserii behaves like \glsuserii except that the link text is converted to uppercase.

\GLSuserii

```
4353 \newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
```

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

```
4354 \newcommand*{\@GLSuserii}[2] [] {%
```

```
4355   \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4356 \def\@GLSuserii@#1#2[#3] {%
```

```
4357   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserii{#2}#3}%
```

```
4358 }
```

\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

```
4359 \newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
```

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

```
4360 \newcommand*{\@glsuseriii}[2] [] {%
```

```
4361   \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4362 \def\@glsuseriii@#1#2[#3] {%
```

```
4363   \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
```

```
4364 }
```

\Glsuseriii behaves like \glsuseriii except that the first letter is converted to uppercase.

\Glsuseriii

```
4365 \newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
```

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

```
4366 \newcommand*{\@Glsuseriii}[2] [] {%
```

```
4367   \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4368 \def\@Glsuseriii@#1#2[#3]{%
4369   \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
4370 }
```

\Glsuseriii behaves like \glsuseriii except that the link text is converted to uppercase.

\Glsuseriii

```
4371 \newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
```

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

```
4372 \newcommand*{\@Glsuseriii}[2][\@Glsuseriii@#1]{%
4373   \new@ifnextchar[\@Glsuseriii@{#1}{#2}]{\@Glsuseriii@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4374 \def\@Glsuseriii@#1#2[#3]{%
4375   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
4376 }
```

\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

```
4377 \newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
```

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

```
4378 \newcommand*{\@glsuseriv}[2][\@glsuseriv@#1]{%
4379   \new@ifnextchar[\@glsuseriv@{#1}{#2}]{\@glsuseriv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4380 \def\@glsuseriv@#1#2[#3]{%
4381   \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
4382 }
```

\Glsuseriv behaves like \glsuseriv except that the first letter is converted to uppercase.

\Glsuseriv

```
4383 \newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
```

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

```
4384 \newcommand*{\@Glsuseriv}[2][\@Glsuseriv@#1]{%
4385   \new@ifnextchar[\@Glsuseriv@{#1}{#2}]{\@Glsuseriv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
4386 \def\@Glsuseriv@#1#2[#3]{%
4387   \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
4388 }
```

\GLSuseriv behaves like \glsuseriv except that the link text is converted to uppercase.

\GLSuseriv

```
4389 \newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
```

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

```
4390 \newcommand*{\@GLSuseriv}[2] [] {%  
4391   \new@ifnextchar [{\@GLSuseriv@{#1}{#2}}]{\@GLSuseriv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4392 \def\@GLSuseriv@#1#2[#3] {%  
4393   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}%  
4394 }
```

\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

```
4395 \newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
```

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

```
4396 \newcommand*{\@glsuserv}[2] [] {%  
4397   \new@ifnextchar [{\@glsuserv@{#1}{#2}}]{\@glsuserv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4398 \def\@glsuserv@#1#2[#3] {%  
4399   \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}}%  
4400 }
```

\Glsuserv behaves like \glsuserv except that the first letter is converted to uppercase.

\Glsuserv

```
4401 \newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
```

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

```
4402 \newcommand*{\@Glsuserv}[2] [] {%  
4403   \new@ifnextchar [{\@Glsuserv@{#1}{#2}}]{\@Glsuserv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4404 \def\@Glsuserv@#1#2[#3] {%  
4405   \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}}%  
4406 }
```

\GLSuserv behaves like \glsuserv except that the link text is converted to uppercase.

\GLSuserv

```
4407 \newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
```

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

```
4408 \newcommand*{\@GLSuserv}[2] [] {%  
4409   \new@ifnextchar [{\@GLSuserv@{#1}{#2}}]{\@GLSuserv@{#1}{#2} [] }}
```

Read in the final optional argument:

```
4410 \def\@GLSuserv@#1#2[#3] {%  
4411   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%  
4412 }
```

\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

```
4413 \newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
```

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

```
4414 \newcommand*{\@glsuservi}[2][\%
```

```
4415   \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4416 \def\@glsuservi@#1#2[#3]{\%
```

```
4417   \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
```

```
4418 }
```

\Glsuservi behaves like \glsuservi except that the first letter is converted to uppercase.

\Glsuservi

```
4419 \newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
```

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

```
4420 \newcommand*{\@Glsuservi}[2][\%
```

```
4421   \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4422 \def\@Glsuservi@#1#2[#3]{\%
```

```
4423   \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
```

```
4424 }
```

\GLSuservi behaves like \glsuservi except that the link text is converted to uppercase.

\GLSuservi

```
4425 \newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
```

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

```
4426 \newcommand*{\@GLSuservi}[2][\%
```

```
4427   \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}[]}]
```

Read in the final optional argument:

```
4428 \def\@GLSuservi@#1#2[#3]{\%
```

```
4429   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}#3}}%
```

```
4430 }
```

Now deal with acronym related keys. First the short form:

\acrshort

```
4431 \newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
```

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

```
4432 \newcommand*{\@ns@acrshort}[2][\%
```

```
4433   \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2}[]}]
```

```
4434 }
```

Read in the final optional argument:

```
4435 \def\@acrshort#1#2[#3]{\%
```

```
4436   \glsdoifexists{#2}%
```

```
4437   {\%
```

```

4438 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

4439 \let\gl@sifplural\@secondoftwo
4440 \let\gl@scapscase\@firstofthree
4441 \let\gl@insert\@empty
4442 \def\glscustomtext{%
4443 \acronymfont{\gl@sentryshort{#2}}#3%
4444 }%

Call \@gl@link Note that \@gl@link sets \glstype.
4445 \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
4446 }%

4447 \glspostlinkhook
4448 }

```

#### \Acrshort

```

4449 \newrobustcmd*{\Acrshort}{\@gl@hyp@opt\ns@Acrshort}

Define the un-starred form. Need to determine if there is a final optional argument
4450 \newcommand*{\ns@Acrshort}[2][{}]{%
4451 \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2}[]}%
4452 }

Read in the final optional argument:
4453 \def\@Acrshort#1#2[#3]{%
4454 \gl@sdoifexists{#2}%
4455 {%

4456 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

4457 \def\glslabel{#2}%
4458 \let\gl@sifplural\@secondoftwo
4459 \let\gl@scapscase\@secondofthree
4460 \let\gl@insert\@empty
4461 \def\glscustomtext{%
4462 \acronymfont{\gl@sentryshort{#2}}#3%
4463 }%

Call \@gl@link Note that \@gl@link sets \glstype.
4464 \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
4465 }%

4466 \glspostlinkhook
4467 }

```

#### \ACRshort

```

4468 \newrobustcmd*{\ACRshort}{\@gl@hyp@opt\ns@ACRshort}

```

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

```
4469 \newcommand*{\ns@ACRshort}[2] [] {%
4470   \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
4471 }
```

Read in the final optional argument:

```
4472 \def\@ACRshort#1#2[#3] {%
4473   \glsdoifexists{#2}%
4474   {%
4475     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4476     \def\glslabel{#2}%
4477     \let\glsifplural\@secondoftwo
4478     \let\glscapscase\@thirdofthree
4479     \let\glsinsert\@empty
4480     \def\glscustomtext{%
4481       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
4482     }%
```

Call \@gl@link Note that \@gl@link sets \glstype.

```
4483   \@gl@link[#1]{#2}{\csname gls\@glstype @entryfmt\endcsname}%
4484   }%
4485   \glspostlinkhook
4486 }
```

Short plural:

\acrshortpl

```
4487 \newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\@ns@acrshortpl}
```

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

```
4488 \newcommand*{\ns@acrshortpl}[2] [] {%
4489   \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}]%
4490 }
```

Read in the final optional argument:

```
4491 \def\@acrshortpl#1#2[#3] {%
4492   \glsdoifexists{#2}%
4493   {%
4494     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4495     \def\glslabel{#2}%
4496     \let\glsifplural\@firstoftwo
4497     \let\glscapscase\@firstofthree
4498     \let\glsinsert\@empty
4499     \def\glscustomtext{%
4500       \acronymfont{\glsentryshortpl{#2}}#3%
4501     }%
```

Call \@gls@link Note that \@gls@link sets \glstype.

```
4502   \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4503   }%

4504   \glspostlinkhook
4505 }
```

\Acrshortpl

```
4506 \newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
```

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

```
4507 \newcommand*{\ns@Acrshortpl}[2] [] {%
4508   \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
4509 }
```

Read in the final optional argument:

```
4510 \def\@Acrshortpl#1#2[#3]{%
4511   \glsdoidexists{#2}%
4512   {%

4513     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

4514     \def\glslabel{#2}%
4515     \let\glsifplural\@firstoftwo
4516     \let\glscapscase\@secondofthree
4517     \let\glsinsert\@empty
4518     \def\glscustomtext{%
4519       \acronymfont{\Glsentryshortpl{#2}}#3%
4520     }%
```

Call \@gls@link Note that \@gls@link sets \glstype.

```
4521   \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4522   }%

4523   \glspostlinkhook
4524 }
```

\ACRshortpl

```
4525 \newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
```

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

```
4526 \newcommand*{\ns@ACRshortpl}[2] [] {%
4527   \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
4528 }
```

Read in the final optional argument:

```
4529 \def\@ACRshortpl#1#2[#3]{%
4530   \glsdoidexists{#2}%
4531   {%

4532     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```



```

4533 \def\glslabel{#2}%
4534 \let\glsifplural\@firstoftwo
4535 \let\glsifscapscase\@thirdofthree
4536 \let\glsinsert\@empty
4537 \def\glscustomtext{%
4538     \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
4539 }%

```

Call \@gls@link Note that \@gls@link sets \glsstyle.

```

4540 \@gls@link[#1]{#2}{\csname gls@\glsstyle @entryfmt\endcsname}%
4541 }%

4542 \glspostlinkhook
4543 }

```

\acrlong

```

4544 \newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}

```

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

```

4545 \newcommand*{\ns@acrlong}[2] [] {%
4546     \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
4547 }

```

Read in the final optional argument:

```

4548 \def\@acrlong#1#2[#3] {%
4549     \glsdoifexists{#2}%
4550     {%
4551         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

4552         \def\glslabel{#2}%
4553         \let\glsifplural\@secondoftwo
4554         \let\glsifscapscase\@firstofthree
4555         \let\glsinsert\@empty

```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```

4556 \def\glscustomtext{%
4557     \glsentrylong{#2}#3%
4558 }%

```

Call \@gls@link Note that \@gls@link sets \glsstyle.

```

4559 \@gls@link[#1]{#2}{\csname gls@\glsstyle @entryfmt\endcsname}%
4560 }%

4561 \glspostlinkhook
4562 }

```

\Acrlong

```

4563 \newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}

```

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

```
4564 \newcommand*{\ns@Acrlong}[2][\]{%
4565   \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
4566 }
```

Read in the final optional argument:

```
4567 \def\@Acrlong#1#2[#3]{%
4568   \glsdoifexists{#2}%
4569   {%
4570     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4571     \def\glslabel{#2}%
4572     \let\glsifplural\@secondoftwo
4573     \let\glscapscase\@secondofthree
4574     \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4575   \def\glscustomtext{%
4576     \Glsentrylong{#2}#3%
4577   }%
```

Call \@gl@link. Note that \@gl@link sets \glstype.

```
4578   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4579   }%
4580   \glspostlinkhook
4581 }
```

\ACRlong

```
4582 \newrobustcmd*{\ACRlong}{\@gl@hyp@opt\ns@ACRlong}
```

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

```
4583 \newcommand*{\ns@ACRlong}[2][\]{%
4584   \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
4585 }
```

Read in the final optional argument:

```
4586 \def\@ACRlong#1#2[#3]{%
4587   \glsdoifexists{#2}%
4588   {%
4589     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4590     \def\glslabel{#2}%
4591     \let\glsifplural\@secondoftwo
4592     \let\glscapscase\@thirdofthree
4593     \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4594 \def\glscustomtext{%
4595 \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
4596 }%
```

Call \@gls@link. Note that \@gls@link sets \glstype.

```
4597 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4598 }%

4599 \glspostlinkhook
4600 }
```

Short plural:

\acrlongpl

```
4601 \newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\@ns@acrlongpl}
```

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

```
4602 \newcommand*{\ns@acrlongpl}[2][\%
4603 \new@ifnextchar[\@acrlongpl{#1}{#2}]{\@acrlongpl{#1}{#2}[]}%
4604 }
```

Read in the final optional argument:

```
4605 \def\@acrlongpl#1#2[#3]{%
4606 \glsdoidexists{#2}%
4607 {%

4608 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

4609 \def\glslabel{#2}%
4610 \let\glsifplural\@firstoftwo
4611 \let\glscapscase\@firstofthree
4612 \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4613 \def\glscustomtext{%
4614 \glsentrylongpl{#2}#3%
4615 }%
```

Call \@gls@link. Note that \@gls@link sets \glstype.

```
4616 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4617 }%

4618 \glspostlinkhook
4619 }
```

\Acrlongpl

```
4620 \newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
```

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

```
4621 \newcommand*{\ns@Acrlongpl}[2][\%
4622   \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
4623 }
```

Read in the final optional argument:

```
4624 \def\@Acrlongpl#1#2[#3]{%
4625   \glsdoifexists{#2}%
4626   {%
4627     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4628     \def\glslabel{#2}%
4629     \let\glsifplural\@firstoftwo
4630     \let\glscapscase\@secondofthree
4631     \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
4632   \def\glscustomtext{%
4633     \Glsentrylongpl{#2}#3%
4634   }%
```

Call \@gl@link. Note that \@gl@link sets \glstype.

```
4635   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4636   }%
4637   \glspostlinkhook
4638 }
```

\ACRlongpl

```
4639 \newrobustcmd*{\ACRlongpl}{\@gl@hyp@opt\ns@ACRlongpl}
```

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

```
4640 \newcommand*{\ns@ACRlongpl}[2][\%
4641   \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
4642 }
```

Read in the final optional argument:

```
4643 \def\@ACRlongpl#1#2[#3]{%
4644   \glsdoifexists{#2}%
4645   {%
4646     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
4647     \def\glslabel{#2}%
4648     \let\glsifplural\@firstoftwo
4649     \let\glscapscase\@thirdofthree
4650     \let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
4651 \def\glscustomtext{%
4652 \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
4653 }%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
4654 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
4655 }%

4656 \glspostlinkhook
4657 }
```

## 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>}
```

```
4658 \newcommand*{\@gls@entry@field}[2]{%
4659 \csname glo@\glsdetoklabel{#1}@#2\endcsname
4660 }
```

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

```
4661 \newcommand*{\glsletentryfield}[3]{%
4662 \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
4663 }
```

`\Gls@entry@field` Generic first letter uppercase version.

```
\@Gls@entry@field{<label>}{<field>}
```

```
4664 \newcommand*{\@Gls@entry@field}[2]{%
4665 \glsdoifexistsordo{#1}%
4666 {%
4667 \letcs{@glo@text}{glo@\glsdetoklabel{#1}@#2}%
4668 \ifdef{@glo@text
4669 {%
4670 \xmakefirstuc{@glo@text}%
4671 }%
4672 {%
4673 ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
4674 entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
```

```

4675     label and the field name}%
4676   }%
4677 }%
4678 {%
4679   ??%
4680 }%
4681 }

```

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`

```

4682 \newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}

```

`\Glsentryname`

```

4683 \newrobustcmd*{\Glsentryname}[1]{%
4684   \@Gls@entryname{#1}%
4685 }

```

`\@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:

```

4686 \newcommand*{\@Gls@entryname}[1]{%
4687   \@Gls@entry@field{#1}{name}%
4688 }

```

`\ls@acentryname` Now the behaviour when `\setacronymstyle` is used:

```

4689 \newcommand*{\@Gls@acentryname}[1]{%
4690   \ifglshaslong{#1}%
4691   {%
4692     \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
4693     \@gls@getbody is defined by mfirstuc (which used to be part of glossaries).
4694     \expandafter\@gls@getbody\@glo@text{}\@nil
4695     \expandafter\ifx\@gls@body\glsentrylong\relax
4696       \expandafter\Glsentrylong\@gls@rest
4697     \else
4698       \expandafter\ifx\@gls@body\glsentryshort\relax
4699         \expandafter\Glsentryshort\@gls@rest
4700       \else
4701         \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.

```

4701     {%
4702       \let\glsentryshort\Glsentryshort
4703       \@glo@text

```

```

4704      }%
4705      \else

4706      \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
4707      {%
4708      \let\glsentryshort\Glsentryshort
4709      \@glo@text
4710      }%
4711      \else
4712      \xmakefirstuc{\@glo@text}%
4713      \fi
4714      \fi
4715      \fi
4716      \fi
4717      }%
4718      {%

```

Not an acronym

```

4719      \@Gls@entry@field{#1}{name}%
4720      }%
4721      }

```

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`

```

4722 \newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}

```

`\Glsentrydesc`

```

4723 \newrobustcmd*{\Glsentrydesc}[1]{%
4724   \@Gls@entry@field{#1}{desc}%
4725 }

```

Plural form:

`entrydescplural`

```

4726 \newcommand*{\glsentrydescplural}[1]{%
4727   \@gls@entry@field{#1}{descplural}%
4728 }

```

`entrydescplural`

```

4729 \newrobustcmd*{\Glsentrydescplural}[1]{%
4730   \@Gls@entry@field{#1}{descplural}%
4731 }

```

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`

```
4732 \newcommand*{\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
```

`\Glsentrytext`

```
4733 \newrobustcmd*{\Glsentrytext}[1]{%
4734   \@Gls@entry@field{#1}{text}%
4735 }
```

Get the plural form:

`\glsentryplural`

```
4736 \newcommand*{\glsentryplural}[1]{%
4737   \@gls@entry@field{#1}{plural}%
4738 }
```

`\Glsentryplural`

```
4739 \newrobustcmd*{\Glsentryplural}[1]{%
4740   \@Gls@entry@field{#1}{plural}%
4741 }
```

Get the symbol associated with this entry. The argument is the label associated with the entry.

`\glsentrysymbol`

```
4742 \newcommand*{\glsentrysymbol}[1]{%
4743   \@gls@entry@field{#1}{symbol}%
4744 }
```

`\Glsentrysymbol`

```
4745 \newrobustcmd*{\Glsentrysymbol}[1]{%
4746   \@Gls@entry@field{#1}{symbol}%
4747 }
```

Plural form:

`\glsentrysymbolplural`

```
4748 \newcommand*{\glsentrysymbolplural}[1]{%
4749   \@gls@entry@field{#1}{symbolplural}%
4750 }
```

`\Glsentrysymbolplural`

```
4751 \newrobustcmd*{\Glsentrysymbolplural}[1]{%
4752   \@Gls@entry@field{#1}{symbolplural}%
4753 }
```

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`

```
4754 \newcommand*{\glsentryfirst}[1]{%
4755   \@gls@entry@field{#1}{first}%
4756 }
```

`\Glsentryfirst`

```
4757 \newrobustcmd*{\Glsentryfirst}[1]{%
4758   \@Gls@entry@field{#1}{first}%
4759 }
```

Get the plural form (as specified by the `firstplural` key when the entry was defined).

`\glsentryfirstplural`

```
4760 \newcommand*{\glsentryfirstplural}[1]{%
4761   \@gls@entry@field{#1}{firstpl}%
4762 }
```

`\Glsentryfirstplural`

```
4763 \newrobustcmd*{\Glsentryfirstplural}[1]{%
4764   \@Gls@entry@field{#1}{firstpl}%
4765 }
```

`\capitalisewords` Make it easier to switch from `\capitalisewords` to `\capitalisefmtwords`.

```
4766 \newcommand{\glscapitalisewords}[1]{%
4767   \capitalisewords{#1}%
4768 }
```

`\glsentrytitlecase`

```
4769 \newrobustcmd*{\@glsentrytitlecase}[2]{%
4770   \glsdoifexists{#1}%
4771   {%
4772     \glsfieldfetch{#1}{#2}{\@gls@value}%
4773     \expandafter\glscapitalisewords\expandafter{\@gls@value}%
4774   }%
4775 }
4776 \ifdef\texorpdfstring
4777 {
4778   \newcommand*{\glsentrytitlecase}[2]{%
4779     \texorpdfstring
4780       {\@glsentrytitlecase{#1}{#2}}%
4781       {\@gls@entry@field{#1}{#2}}%
4782   }
4783 }
4784 {
4785   \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
4786 }
```

Display the glossary type with which this entry is associated (as specified by the `type` key used when the entry was defined)

`\glentrytype`

```
4787 \newcommand*{\glentrytype}[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.

`\glentrysort`

```
4788 \newcommand*{\glentrysort}[1]{%
4789   \@gls@entry@field{#1}{sort}%
4790 }
```

`\glentryparent` Expands to the label of the entry's parent.

```
4791 \newcommand*{\glentryparent}[1]{%
4792   \@gls@entry@field{#1}{parent}%
4793 }
```

`\glentryuseri` Get the first user key (as specified by the user1 when the entry was defined). The argument is the label associated with the entry.

```
4794 \newcommand*{\glentryuseri}[1]{%
4795   \@gls@entry@field{#1}{useri}%
4796 }
```

`\Glsentryuseri`

```
4797 \newrobustcmd*{\Glsentryuseri}[1]{%
4798   \@Gls@entry@field{#1}{useri}%
4799 }
```

`\glentryuserii` Get the second user key (as specified by the user2 when the entry was defined). The argument is the label associated with the entry.

```
4800 \newcommand*{\glentryuserii}[1]{%
4801   \@gls@entry@field{#1}{userii}%
4802 }
```

`\Glsentryuserii`

```
4803 \newrobustcmd*{\Glsentryuserii}[1]{%
4804   \@Gls@entry@field{#1}{userii}%
4805 }
```

`\glentryuseriii` Get the third user key (as specified by the user3 when the entry was defined). The argument is the label associated with the entry.

```
4806 \newcommand*{\glentryuseriii}[1]{%
4807   \@gls@entry@field{#1}{useriii}%
4808 }
```

`\Glsentryuseriii`

```
4809 \newrobustcmd*{\Glsentryuseriii}[1]{%
4810   \@Gls@entry@field{#1}{useriii}%
4811 }
```

\glentryuseriv Get the fourth user key (as specified by the user4 when the entry was defined). The argument is the label associated with the entry.

```
4812 \newcommand*{\glentryuseriv}[1]{%
4813   \@gls@entry@field{#1}{useriv}%
4814 }
```

\Glsentryuseriv

```
4815 \newrobustcmd*{\Glsentryuseriv}[1]{%
4816   \@Gls@entry@field{#1}{useriv}%
4817 }
```

\glentryuserv Get the fifth user key (as specified by the user5 when the entry was defined). The argument is the label associated with the entry.

```
4818 \newcommand*{\glentryuserv}[1]{%
4819   \@gls@entry@field{#1}{userv}%
4820 }
```

\Glsentryuserv

```
4821 \newrobustcmd*{\Glsentryuserv}[1]{%
4822   \@Gls@entry@field{#1}{userv}%
4823 }
```

\glentryuservi Get the sixth user key (as specified by the user6 when the entry was defined). The argument is the label associated with the entry.

```
4824 \newcommand*{\glentryuservi}[1]{%
4825   \@gls@entry@field{#1}{uservi}%
4826 }
```

\Glsentryuservi

```
4827 \newrobustcmd*{\Glsentryuservi}[1]{%
4828   \@Gls@entry@field{#1}{uservi}%
4829 }
```

\glentryshort Get the short key (as specified by the short the entry was defined). The argument is the label associated with the entry.

```
4830 \newcommand*{\glentryshort}[1]{\@gls@entry@field{#1}{short}}
```

\Glsentryshort

```
4831 \newrobustcmd*{\Glsentryshort}[1]{%
4832   \@Gls@entry@field{#1}{short}%
4833 }
```

glentryshortpl Get the short plural key (as specified by the shortplural the entry was defined). The argument is the label associated with the entry.

```
4834 \newcommand*{\glentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
```

Glsentryshortpl

```
4835 \newrobustcmd*{\Glsentryshortpl}[1]{%
4836   \@Gls@entry@field{#1}{shortpl}%
4837 }
```

`\glsentrylong` Get the long key (as specified by the long the entry was defined). The argument is the label associated with the entry.

```
4838 \newcommand*{\glsentrylong}[1]{\@Gls@entry@field{#1}{long}}
```

`\Glsentrylong`

```
4839 \newrobustcmd*{\Glsentrylong}[1]{%
4840   \@Gls@entry@field{#1}{long}%
4841 }
```

`\glsentrylongpl` Get the long plural key (as specified by the longplural the entry was defined). The argument is the label associated with the entry.

```
4842 \newcommand*{\glsentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}
```

`\Glsentrylongpl`

```
4843 \newrobustcmd*{\Glsentrylongpl}[1]{%
4844   \@Gls@entry@field{#1}{longpl}%
4845 }
```

Short cut macros to access full form:

`\glsentryfull`

```
4846 \newcommand*{\glsentryfull}[1]{%
4847   \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
4848 }
```

`\Glsentryfull`

```
4849 \newrobustcmd*{\Glsentryfull}[1]{%
4850   \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
4851 }
```

`\glsentryfullpl`

```
4852 \newcommand*{\glsentryfullpl}[1]{%
4853   \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
4854 }
```

`\Glsentryfullpl`

```
4855 \newrobustcmd*{\Glsentryfullpl}[1]{%
4856   \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
4857 }
```

entrynumberlist Displays the number list as is.

```

4858 \newcommand*\glentrynumberlist}[1]{%
4859   \glsdoifexists{#1}%
4860   {%
4861     \@gls@entry@field{#1}{numberlist}%
4862   }%
4863 }

```

splaynumberlist Formats the number list for the given entry label. Doesn't work with hyperref.

```

4864 \@ifpackageloaded{hyperref} {%
4865   \newcommand*\glsdisplaynumberlist}[1]{%
4866     \GlossariesWarning
4867     {%
4868       \string\glsdisplaynumberlist\space
4869       doesn't work with hyperref.^^JUsing
4870       \string\glentrynumberlist\space instead%
4871     }%
4872     \glentrynumberlist{#1}%
4873   }%
4874 }%
4875 {%
4876   \newcommand*\glsdisplaynumberlist}[1]{%
4877     \glsdoifexists{#1}%
4878     {%
4879       \bgroup

4880         \protected@edef\@glo@label{\glsdetoklabel{#1}}%
4881         \let\@org@glsnumberformat\glsnumberformat
4882         \def\glsnumberformat##1{##1}%
4883         \protected@edef\the@numberlist{%
4884           \csname glo@\@glo@label @numberlist\endcsname}%
4885         \def\@gls@numlist@sep{%
4886         \def\@gls@numlist@nextsep{%
4887         \def\@gls@numlist@lastsep{%
4888         \def\@gls@thislist{%
4889         \def\@gls@donext@def{%
4890         \renewcommand\do[1]{%
4891           \protected@edef\@gls@thislist{%
4892             \@gls@thislist
4893             \noexpand\@gls@numlist@sep
4894             ##1%
4895           }%
4896           \let\@gls@numlist@sep\@gls@numlist@nextsep
4897           \def\@gls@numlist@nextsep{\glsnumlistsep}%
4898           \@gls@donext@def
4899           \def\@gls@donext@def{%
4900             \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
4901           }%
4902         }%

```

```

4903         \expandafter \glsnumlistparser \expandafter{\the@numberlist}%
4904         \let\@gls@numlist@sep\@gls@numlist@lastsep
4905         \@gls@thislist
4906     \egroup
4907 }%
4908 }
4909 }

```

`\glsnumlistsep`

```

4910 \newcommand*{\glsnumlistsep}{, }

```

`\glsnumlistlastsep`

```

4911 \newcommand*{\glsnumlistlastsep}{ \& }

```

`\gls hyperlink`

Provide a hyperlink to a glossary entry without adding information to the glossary file. The entry needs to be added using a command like `\gls link` or `\gls add` to ensure that the target is defined. The first (optional) argument specifies the link text. The entry name is used by default. The second argument is the entry label.

```

4912 \newcommand*{\gls hyperlink}[2][\gls entrytext{\@glo@label}]{%
4913   \def\@glo@label{#2}%
4914   \@gls link{\glo link prefix\gls detok label{#2}}{#1}}

```

## 1.12 Adding an entry to the glossary without generating text

The following keys are provided for `\gls add` and `\gls add all`:

```

4915 \define@key{gloss add}{counter}{\def\@gls@counter{#1}}
4916 \define@key{gloss add}{format}{\def\@gls number format{#1}}

```

This key is only used by `\gls add all`:

```

4917 \define@key{gloss add}{types}{\def\@glo@type{#1}}

```

`\gls add[<options>]{<label>}`

Add a term to the glossary without generating any link text. The optional argument indicates which counter to use, and how to format it (using a key-value list) the second argument is the entry label. Note that *<options>* only has two keys: counter and format (the types key will be ignored).

`\gls add`

```

4918 \newrobustcmd*{\gls add}[2][ ]{%

```

Need to move to horizontal mode if not already in it, but only if not in preamble.

```

4919   \@gls@adjustmode
4920   \gls do if exists{#2}%
4921   {%
4922     \def\@gls number format{\gls number format}%

```

```

4923 \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
4924 \setkeys{glossadd}{#1}%

```

Store the entry's counter in \theglsentrycounter

```

4925 \@gls@saveentrycounter

```

Define sort key if necessary:

```

4926 \@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.

```

4927 \@do@wrglossary{#2}%
4928 }%
4929 }

```

@gls@adjustmode

```

4930 \newcommand*{\@gls@adjustmode}{}
4931 \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

```

4932 \newrobustcmd*{\glsaddall}[1][]{%
4933 \protected@edef\@glo@type{\@glo@types}%
4934 \setkeys{glossadd}{#1}%
4935 \forallglsentries[\@glo@type]{\@glo@entry}{%
4936 \glsadd[#1]{\@glo@entry}%
4937 }%
4938 }

```

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.

```

4939 \newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
4940 \forallglsentries[#1]{\@glo@entry}%
4941 {%
4942 \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
4943 }%
4944 }

```

\glsignore

```

4945 \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 `\glsgetgrouptitle` 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.
4946 \edef\glsopenbrace{\expandafter\@gobble\string\{}}

\glsclosebrace Define \glsclosebrace to make it easier to write an opening brace to a file.
4947 \edef\glsclosebrace{\expandafter\@gobble\string\}}

\glsbackslash  Define \glsbackslash to make it easier to write a backslash to a file.
4948 \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.
4949 \edef\glsquote#1{\string"#1\string"}

\glpercentchar Define \glpercentchar to make it easier to write a percent character to a file.
4950 \edef\glpercentchar{\expandafter\@gobble\string\%}

\glstildechar  Define \glstildechar to make it easier to write a tilde character to a file.
4951 \edef\glstildechar{\string~}

@glsfirstletter Define the first letter to come after the digits 0,...,9. Only required for xindy.
4952 \ifglsxindy
4953   \newcommand*{\@glsfirstletter}{A}
4954 \fi

tterAfterDigits Sets the first letter to come after the digits 0,...,9. The starred version sanitizes.
4955 \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{\%
4956   \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
4957 \ifglsxindy
4958   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{\%
```



```

4959 \renewcommand*{\@glsfirstletter}{#1}}
4960 \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
4961 \renewcommand*{\@glsfirstletter}{#1}%
4962 \@onelevel@sanitize\@glsfirstletter
4963 }
4964 \else
4965 \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
4966 \glsnoindywarning\GlsSetXdyFirstLetterAfterDigits}
4967 \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
4968 \@GlsSetXdyFirstLetterAfterDigits
4969 }
4970 \fi

```

**numbergrouporder** Specifies the order of the number group.

```

4971 \ifglxindy
4972 \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
4973 \fi

```

**numberGroupOrder** Sets the relative location of the number group. The starred version sanitizes.

```

4974 \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
4975 \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
4976 }
4977 \ifglxindy
4978 \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
4979 \renewcommand*{\@xdynumbergrouporder}{#1}%
4980 }
4981 \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
4982 \renewcommand*{\@xdynumbergrouporder}{#1}%
4983 \@onelevel@sanitize\@xdynumbergrouporder
4984 }
4985 \else
4986 \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
4987 \glsnoindywarning\GlsSetXdyNumberGroupOrder}
4988 \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
4989 \@GlsSetXdyNumberGroupOrder}
4990 \fi

```

**\@glsminrange** Define the minimum number of successive location references to merge into a range.

```

4991 \newcommand*{\@glsminrange}{2}

```

**yMinRangeLength** 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.

```

4992 \ifglxindy
4993 \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
4994 \renewcommand*{\@glsminrange}{#1}}
4995 \else
4996 \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
4997 \glsnoindywarning\GlsSetXdyMinRangeLength}
4998 \fi

```

```

\writeist
4999 \ifglxindy
    Code to use if xindy is required.
5000 \def\writeist{%
    Define write register if not already defined
5001 \ifundef{\glswrite}{\newwrite\glswrite}{}%
    Update attributes list
5002 \@gls@addpredefinedattributes
    Open the file.
5003 \openout\glswrite=\istfilename
    Write header comment at the start of the file
5004 \write\glswrite{;; xindy style file created by the glossaries
5005     package}%
5006 \write\glswrite{;; for document '\jobname' on
5007     \the\year-\the\month-\the\day}%
    Specify the required styles
5008 \write\glswrite{^^J; required styles^^J}
5009 \@for\@xdystyle:=\@xdyrequiredstyles\do{%
5010     \ifx\@xdystyle\@empty
5011     \else
5012     \protected@write\glswrite{{(require
5013         \string"\@xdystyle.xdy\string")}}%
5014     \fi
5015 }%
    List the allowed attributes (possible values used by the format key)
5016 \write\glswrite{^^J%
5017     ; list of allowed attributes (number formats)^^J}%
5018 \write\glswrite{(define-attributes ((\@xdyattributes)))}%
    Define any additional alphabets
5019 \write\glswrite{^^J; user defined alphabets^^J}%
5020 \write\glswrite{\@xdyuseralphabets}%
    Define location classes.
5021 \write\glswrite{^^J; location class definitions^^J}%
    As from version 3.0, locations are now specified as {\<Hprefix>}{\<number>}, so need to add all
    possible combinations of location types.
5022 \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
    Case where \<Hprefix> is empty:
5023 \protected@write\glswrite{{(define-location-class
5024     \string"\@gls@classI\string"^^J\space\space\space
5025     (
5026         :sep "{ }"
5027         \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space

```

```

5028         :sep "}"
5029     )
5030     ^^J\space\space\space
5031     :min-range-length \@glsminrange^^J%
5032 )
5033 }%

```

Nested iteration over all classes:

```

5034 {%
5035     \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
5036         \protected@write\glswrite{}\{(define-location-class
5037             \string"\@gls@classII-\@gls@classI\string"
5038             ^^J\space\space\space
5039             (
5040                 :sep "{"
5041                 \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
5042                 :sep "{{"
5043                 \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
5044                 :sep "}"
5045             )
5046             ^^J\space\space\space
5047             :min-range-length \@glsminrange^^J%
5048         )
5049     }%
5050 }%
5051 }%
5052 }%

```

User defined location classes (needs checking for new location format).

```

5053 \write\glswrite{^^J; user defined location classes}%
5054 \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.)

```

5055 \write\glswrite{^^J; define cross-reference class^^J}%
5056 \write\glswrite{(define-crossref-class \string"see\string"
5057     :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.)

```

5058 \write\glswrite{(markup-crossref-list
5059     :class \string"see\string"^^J\space\space\space
5060     :open \string"\string\glsseeformat\string"
5061     :close \string"{}\string")}%

```

Provide hook to write extra material here (used by `glossaries-extra` to define a `seealso` class).

```

5062 \@xdycrossrefhook

```

List the order to sort the classes.

```
5063 \write\glswrite{^^J; define the order of the location classes}%
5064 \write\glswrite{(define-location-class-order
5065 (\@xdylocationclassorder))}%
```

Specify what to write to the start and end of the glossary file.

```
5066 \write\glswrite{^^J; define the glossary markup^^J}%

5067 \write\glswrite{(markup-index^^J\space\space\space
5068 :open \string"\string
5069 \glossarysection[\string\glossarytoctitle]{\string
5070 \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)

```
5071 \@for\@this@ctr:=\@xdycounters\do{%
5072   {%
5073     \@for\@this@attr:=\@xdyattributelist\do{%
5074       \protected\write\glswrite{}\string\providecommand*%
5075       \expandafter\string
5076       \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
5077       {%
5078         \string\setentrycounter
5079         [\expandafter\@gobble\string\#1]{\@this@ctr}%
5080         \expandafter\string
5081         \csname\@this@attr\endcsname
5082         {\expandafter\@gobble\string\#2}%
5083       }%
5084     }%
5085   }%
5086 }%
5087 }%
```

Add the end part of the open tag and the rest of the markup-index information:

```
5088 \write\glswrite{%
5089 \string\begin
5090 {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
5091 \space\space:close \string"\glspercentchar\glstildechar n\string
5092 \end{theglossary}\string\glossarypostamble
5093 \glstildechar n\string" ^^J\space\space\space
5094 :tree)}}%
```

Specify what to put between letter groups

```
5095 \write\glswrite{(markup-letter-group-list
5096 :sep \string"\string\glsgroupskip\glstildechar n\string")}%
```

Specify what to put between entries

```
5097 \write\glswrite{(markup-indexentry
5098 :open \string"\string\relax \string\glsresetentrylist
5099 \glstildechar n\string")}%
```

Specify how to format entries

```
5100 \write\glswrite{(markup-locclass-list :open
5101 \string\glsopenbrace\string\glossaryentrynumbers
5102 \glsopenbrace\string\relax\space \string^^J\space\space\space
5103 :sep \string", \string"
5104 :close \string\glsclosebrace\glsclosebrace\string)}}%
```

Specify how to separate location numbers

```
5105 \write\glswrite{(markup-locref-list
5106 :sep \string\string\delimN\space\string)}}%
```

Specify how to indicate location ranges

```
5107 \write\glswrite{(markup-range
5108 :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.

```
5109 \@onelevel@sanitize\gls@suffixF
5110 \@onelevel@sanitize\gls@suffixFF
5111 \ifx\gls@suffixF\@empty
5112 \else
5113 \write\glswrite{(markup-range
5114 :close "\gls@suffixF" :length 1 :ignore-end)}}%
5115 \fi
5116 \ifx\gls@suffixFF\@empty
5117 \else
5118 \write\glswrite{(markup-range
5119 :close "\gls@suffixFF" :length 2 :ignore-end)}}%
5120 \fi
```

Specify how to format locations.

```
5121 \write\glswrite{^^J; define format to use for locations^^J}%
5122 \write\glswrite{\@xdylocref}%
```

Specify how to separate letter groups.

```
5123 \write\glswrite{^^J; define letter group list format^^J}%
5124 \write\glswrite{(markup-letter-group-list
5125 :sep \string\string\glsgroupskip\glstildechar n\string)}}%
```

Define letter group headings.

```
5126 \write\glswrite{^^J; letter group headings^^J}%
5127 \write\glswrite{(markup-letter-group
5128 :open-head \string\string\glsgroupheading
5129 \glsopenbrace\string^^J\space\space\space
5130 :close-head \string\glsclosebrace\string)}}%
```

Define additional letter groups.

```
5131 \write\glswrite{^^J; additional letter groups^^J}%
5132 \write\glswrite{\@xdylettergroups}%
```

Define additional sort rules

```
5133 \write\glswrite{^^J; additional sort rules^^J}
5134 \write\glswrite{\@xdysortrules}%
```

Hook for any additional information:

```
5135 \@gls@writeisthook
```

Close the style file

```
5136 \closeout\glswrite
```

Suppress any further calls.

```
5137 \let\writeist\relax
5138 }
5139 \else
```

Code to use if makeindex is required.

```
5140 \edef\@gls@actualchar{\string?}
5141 \edef\@gls@encapchar{\string|}
5142 \edef\@gls@levelchar{\string!}
5143 \edef\@gls@quotechar{\string"}%
5144 \let\GlsSetQuote\gls@nosetquote
5145 \def\writeist{\relax
5146 \ifundefined\glswrite{\newwrite\glswrite}{}\relax
5147 \openout\glswrite=\istfilename
5148 \write\glswrite{\glspersentchar\space makeindex style file
5149 created by the glossaries package}
5150 \write\glswrite{\glspersentchar\space for document
5151 '\jobname' on \the\year-\the\month-\the\day}
5152 \write\glswrite{actual '@gls@actualchar'}
5153 \write\glswrite{encap '@gls@encapchar'}
5154 \write\glswrite{level '@gls@levelchar'}
5155 \write\glswrite{quote '@gls@quotechar'}
5156 \write\glswrite{keyword \string"\string\glossaryentry\string"}
5157 \write\glswrite{preamble \string"\string\glossarysection[\string
5158 \glossarytoctitle]{\string\glossarytitle}\string
5159 \glossarypreamble\string\n\string\begin{theglossary}\string
5160 \glossaryheader\string\n\string"}
5161 \write\glswrite{postamble \string"\string%\string\n\string
5162 \end{theglossary}\string\glossarypostamble\string\n
5163 \string"}
5164 \write\glswrite{group_skip \string"\string\glsgroupskip\string\n
5165 \string"}
5166 \write\glswrite{item_0 \string"\string%\string\n\string"}
5167 \write\glswrite{item_1 \string"\string%\string\n\string"}
5168 \write\glswrite{item_2 \string"\string%\string\n\string"}
5169 \write\glswrite{item_01 \string"\string%\string\n\string"}
5170 \write\glswrite{item_x1
5171 \string"\string\relax \string\glsresetentrylist\string\n
5172 \string"}
5173 \write\glswrite{item_12 \string"\string%\string\n\string"}
5174 \write\glswrite{item_x2
```

```

5175     \string"\string\relax \string\glsresetentrylist\string\n
5176     \string"}

5177     \write\glswrite{delim_0 \string"\string\{\string
5178         \glossaryentrynumbers\string\{\string\relax \string"}
5179     \write\glswrite{delim_1 \string"\string\{\string
5180         \glossaryentrynumbers\string\{\string\relax \string"}
5181     \write\glswrite{delim_2 \string"\string\{\string
5182         \glossaryentrynumbers\string\{\string\relax \string"}
5183     \write\glswrite{delim_t \string"\string\}\string\}\string"}
5184     \write\glswrite{delim_n \string"\string\delimN \string"}
5185     \write\glswrite{delim_r \string"\string\delimR \string"}
5186     \write\glswrite{headings_flag 1}
5187     \write\glswrite{heading_prefix
5188         \string"\string\glsgroupheading\string\{\string"}
5189     \write\glswrite{heading_suffix
5190         \string"\string\}\string\relax
5191         \string\glsresetentrylist \string"}
5192     \write\glswrite{symhead_positive \string"glssymbols\string"}
5193     \write\glswrite{numhead_positive \string"glsglnumbers\string"}
5194     \write\glswrite{page_compositor \string"glsglcompositor\string"}
5195     \@gls@escbsdq\gls@suffixF
5196     \@gls@escbsdq\gls@suffixFF
5197     \ifx\gls@suffixF\@empty
5198     \else
5199         \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
5200     \fi
5201     \ifx\gls@suffixFF\@empty
5202     \else
5203         \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
5204     \fi

```

Hook for any additional information:

```

5205     \@gls@writeisthook

    Close the file and disable \writeist.

5206     \closeout\glswrite
5207     \let\writeist\relax
5208 }
5209 \fi

```

**SetWriteIstHook** Allow user to append information to the style file.

```

5210 \newcommand*{\GlsSetWriteIstHook}[1]{\renewcommand*{\@gls@writeisthook}{#1}}
5211 \@onlypremakeg\GlsSetWriteIstHook

```

**ls@writeisthook**

```

5212 \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.

```

5213 \ifglxsindy
5214 \newcommand*{\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
5215 \newcommand*{\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
5216 \else
5217 \newcommand*{\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}}%

    If German is in use, set the extra makeindex option so makeglossaries can pick it up.

5218 \@ifpackageloaded{tracklang}%
5219 {%
5220 \IfTrackedLanguage{german}%
5221 {%
5222 \def\@gls@extramakeindexopts{-g}%
5223 }%
5224 }%
5225 }%
5226 {}%

    Need to redefine \@gls@checkquote

5227 \edef\@gls@docheckquotedef{%
5228 \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
5229 \noexpand\@gls@tmpb=\noexpand\expandafter\noexpand\@gls@checkedmkidx{%
5230 \noexpand\toks@={####1}%
5231 \noexpand\ifx\noexpand\null####2\noexpand\null
5232 \noexpand\ifx\noexpand\null####3\noexpand\null
5233 \noexpand\edef\noexpand\@gls@checkedmkidx{%
5234 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
5235 \noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
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\@gls@quotechar\noexpand\@gls@quotechar}%
5241 \noexpand\def\noexpand\@gls@checkquote{%
5242 \noexpand\@gls@checkquote####3\noexpand\null}%
5243 \noexpand\fi
5244 \noexpand\else
5245 \noexpand\edef\noexpand\@gls@checkedmkidx{%
5246 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
5247 \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
5248 \noexpand\ifx\noexpand\null####3\noexpand\null
5249 \noexpand\def\noexpand\@gls@checkquote{%
5250 \noexpand\@gls@checkquote####2#1#1\noexpand\null}%
5251 \noexpand\else
5252 \noexpand\def\noexpand\@gls@checkquote{%
5253 \noexpand\@gls@checkquote####2#1####3\noexpand\null}%
5254 \noexpand\fi
5255 \noexpand\fi
5256 \noexpand\@gls@checkquote
5257 }%
5258 }%

```



```

5259 \@gls@docheckquotedef
5260 \edef\@gls@docheckquotedef{%
5261   \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
5262     \noexpand\def\noexpand\@gls@checkedmkidx{%
5263       \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
5264       #1#1\noexpand\null
5265       \noexpand\expandafter\noexpand\@gls@updatechecked
5266       \noexpand\@gls@checkedmkidx{####1}%
5267       \noexpand\def\noexpand\@gls@checkedmkidx{%
5268         \noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
5269         \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
5270         \noexpand\null
5271         \noexpand\expandafter\noexpand\@gls@updatechecked
5272         \noexpand\@gls@checkedmkidx{####1}%
5273         \noexpand\def\noexpand\@gls@checkedmkidx{%
5274           \noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
5275           \noexpand\?\noexpand\?\noexpand\null
5276           \noexpand\expandafter\noexpand\@gls@updatechecked
5277           \noexpand\@gls@checkedmkidx{####1}%
5278           \noexpand\def\noexpand\@gls@checkedmkidx{%
5279             \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
5280             \noexpand?\noexpand?\noexpand\null
5281             \noexpand\expandafter\noexpand\@gls@updatechecked
5282             \noexpand\@gls@checkedmkidx{####1}%
5283             \noexpand\def\noexpand\@gls@checkedmkidx{%
5284               \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
5285               \noexpand|\noexpand|\noexpand\null
5286               \noexpand\expandafter\noexpand\@gls@updatechecked
5287               \noexpand\@gls@checkedmkidx{####1}%
5288               \noexpand\def\noexpand\@gls@checkedmkidx{%
5289                 \noexpand\expandafter\noexpand\@gls@checkescbar####1\noexpand\@nil
5290                 \noexpand||\noexpand||\noexpand\null
5291                 \noexpand\expandafter\noexpand\@gls@updatechecked
5292                 \noexpand\@gls@checkedmkidx{####1}%
5293                 \noexpand\def\noexpand\@gls@checkedmkidx{%
5294                   \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
5295                   \noexpand!\noexpand!\noexpand\null
5296                   \noexpand\expandafter\noexpand\@gls@updatechecked
5297                   \noexpand\@gls@checkedmkidx{####1}%
5298
5299                   \noexpand\def\noexpand\@gls@checkedmkidx{%
5300                     \noexpand\expandafter\noexpand\@gls@checkesclevel####1\noexpand\@nil
5301                     \noexpand\!\noexpand\!\noexpand\null
5302                     \noexpand\expandafter\noexpand\@gls@updatechecked
5303                     \noexpand\@gls@checkedmkidx{####1}%
5304
5305                     \noexpand\def\noexpand\@gls@checkedmkidx{%
5306                       \noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
5307                       \backslashchar\backslashchar\noexpand\null
5308                       \noexpand\expandafter\noexpand\@gls@updatechecked

```

```

5307         \noexpand\@gls@checkedmkidx{####1}%
5308     }%
5309 }%
5310 \@gls@docheckquotedef
5311 \edef\@gls@docheckquotedef{%
5312     \noexpand\def\noexpand\@gls@checkescquote####1%
5313         \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
5314         ####3\noexpand\null{%
5315             \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
5316             \noexpand\toks@={####1}%
5317             \noexpand\ifx\noexpand\null####2\noexpand\null
5318             \noexpand\ifx\noexpand\null####3\noexpand\null
5319             \noexpand\edef\noexpand\@gls@checkedmkidx{%
5320                 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
5321             \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
5322             \noexpand\else
5323             \noexpand\edef\noexpand\@gls@checkedmkidx{%
5324                 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
5325                 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
5326                     \csname#1\endcsname}\noexpand\@gls@quotechar
5327                 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
5328                     \csname#1\endcsname}\noexpand\@gls@quotechar}%
5329             \noexpand\def\noexpand\@gls@checkescquote{%
5330                 \noexpand\@gls@checkescquote####3\noexpand\null}%
5331         \noexpand\fi
5332     \noexpand\else
5333     \noexpand\edef\noexpand\@gls@checkedmkidx{%
5334         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
5335         \noexpand\@gls@quotechar\noexpand\string
5336         \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
5337     \noexpand\ifx\noexpand\null####3\noexpand\null
5338     \noexpand\def\noexpand\@gls@checkescquote{%
5339         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
5340         \expandonce{\csname#1\endcsname}\noexpand\null}%
5341     \noexpand\else
5342     \noexpand\def\noexpand\@gls@checkescquote{%
5343         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
5344         ####3\noexpand\null}%
5345     \noexpand\fi
5346     \noexpand\fi
5347     \noexpand\@gls@checkescquote
5348 }%
5349 }%
5350 \@gls@docheckquotedef
5351 }
5352 \newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
5353     {\string\GlsSetQuote\space not permitted here}%
5354     {\Move \string\GlsSetQuote\space earlier in the preamble, as
5355     soon as possible after glossaries.sty has been loaded}}

```

```
5356 \fi
```

```
ramakeindexopts
```

```
5357 \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
```

```
5358 \newcommand{\noist}{%
```

Update attributes list

```
5359 \@gls@addpredefinedattributes
```

```
5360 \let\writeist\relax
```

```
5361 }
```

`\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.

```
5362 \newcommand*{\@makeglossary}[1]{%
```

```
5363 \ifglossaryexists{#1}%
```

```
5364 {%
```

Only create a new write if `savewrites=false` otherwise create a token to collect the information.

```
5365 \ifglssavewrites
```

```
5366 \expandafter\newtoks\csname glo@#1@filetok\endcsname
```

```
5367 \else
```

```
5368 \expandafter\newwrite\csname glo@#1@file\endcsname
```

```
5369 \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
```

```
5370 \fi
```

```
5371 \@gls@renewglossary
```

```
5372 \writeist
```

```
5373 }%
```

```
5374 {%
```

```
5375 \PackageError{glossaries}%
```

```
5376 {Glossary type ‘#1’ not defined}%
```

```
5377 {New glossaries must be defined before using \string\makeglossaries}%
```

```
5378 }%
```

```
5379 }
```

`\@glsopenfile` Open write file associated with the given glossary.

```
5380 \newcommand*{\@glsopenfile}[2]{%
5381   \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
5382   \PackageInfo{glossaries}{Writing glossary file
5383     \jobname.\csname @glotype@#2@out\endcsname}%
5384 }
```

`\@closegls`

```
5385 \newcommand*{\@closegls}[1]{%
5386   \closeout\csname glo@#1@file\endcsname
5387 }
```

`\@gls@automake` Unstarred form of `\ifglossaryexists` is used as `\@gls@automake` can't be used with an ignored glossary.

```
5388 \ifglxindy
5389 \newcommand*{\@gls@automake}[1]{%
5390   \ifglossaryexists{#1}
5391   {%
5392     \@closegls{#1}%
5393     \ifdefstring{\glsorder}{letter}%
5394       {\def\@gls@order{-M ord/letorder }}%
5395       {\let\@gls@order\@empty}%
5396     \ifcsundef{@xdy@#1@language}%
5397       {\let\@gls@langmod\@xdy@main@language}%
5398       {\letcs\@gls@langmod{@xdy@#1@language}}%
5399     \edef\@gls@dothiswrite{\noexpand\write18{xindy
5400       -I xindy
5401       \@gls@order
5402       -L \@gls@langmod\space
5403       -M \gls@istfilebase\space
5404       -C \gls@codepage\space
5405       -t \jobname.\csuse{@glotype@#1@log}
5406       -o \jobname.\csuse{@glotype@#1@in}
5407       \jobname.\csuse{@glotype@#1@out}}}%
5408     }%
5409     \@gls@dothiswrite
5410   }%
5411   {%
5412     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5413   }%
5414 }
5415 \else
5416 \newcommand*{\@gls@automake}[1]{%
5417   \ifglossaryexists{#1}
5418   {%
5419     \@closegls{#1}%
5420     \ifdefstring{\glsorder}{letter}%
5421       {\def\@gls@order{-l }}%
5422       {\let\@gls@order\@empty}%

```

```

5423 \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
5424 -s \listfilename\space
5425 -t \jobname.\csuse{@glotype@#1@log}
5426 -o \jobname.\csuse{@glotype@#1@in}
5427 \jobname.\csuse{@glotype@#1@out}}}%
5428 }%
5429 \@gls@dothiswrite
5430 }%
5431 {%
5432 \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5433 }%
5434 }
5435 \fi

```

omake@immediate Unstarred form of \ifglossaryexists is used as \@gls@automake@immediate can't be used with an ignored glossary.

```

5436 \ifglxsindy
5437 \newcommand*{\@gls@automake@immediate}[1]{%
5438 \ifglossaryexists{#1}
5439 {%
5440 \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
5441 {%
5442 \ifdefstring{\glsorder}{letter}%
5443 {\def\@gls@order{-M ord/letorder }}%
5444 {\let\@gls@order\@empty}%
5445 \ifcsundef{@xdy@#1@language}%
5446 {\let\@gls@langmod\@xdy@main@language}%
5447 {\letcs\@gls@langmod{ @xdy@#1@language}}}%
5448 \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
5449 -I xindy
5450 \@gls@order
5451 -L \@gls@langmod\space
5452 -M \gls@listfilebase\space
5453 -C \gls@codepage\space
5454 -t \jobname.\csuse{@glotype@#1@log}
5455 -o \jobname.\csuse{@glotype@#1@in}
5456 \jobname.\csuse{@glotype@#1@out}}}%
5457 }%
5458 \@gls@dothiswrite
5459 }%
5460 {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
5461 doesn't exist. Rerun may be required}}}%
5462 }%
5463 {%
5464 \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5465 }%
5466 }
5467 \else
5468 \newcommand*{\@gls@automake@immediate}[1]{%

```

```

5469 \ifglossaryexists{#1}
5470 {%
5471   \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
5472     {%
5473       \ifdefstring{\glsorder}{letter}%
5474       {\def\@gls@order{-1 }}%
5475       {\let\@gls@order\empty}%
5476       \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
5477         -s \istfilename\space
5478         -t \jobname.\csuse{@glotype@#1@log}
5479         -o \jobname.\csuse{@glotype@#1@in}
5480         \jobname.\csuse{@glotype@#1@out}}}%
5481     }%
5482     \@gls@dothiswrite
5483   }%
5484   {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
5485     doesn't exist. Rerun may be required}}%
5486 }%
5487 {%
5488   \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
5489 }%
5490 }
5491 \fi

```

omakeglossaries Issue warning that \makeglossaries hasn't been used.

```

5492 \newcommand*{\@warn@nomakeglossaries}{}

```

Only use this if warning if \printglossary has been used without \makeglossaries

```

5493 \newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}

```

omake@immediate

```

5494 \newcommand{\@gls@@automake@immediate}{%
5495   \ifnum\gls@automake@nr=2\relax
5496   \@for\@gls@type:=\@glo@types\do{%
5497     \ifdefempty{\@gls@type}{}%
5498     {\@gls@automake@immediate{\@gls@type}}%
5499   }%
5500   \glsautomakefalse
5501   \renewcommand*{\@gls@doautomake}{}%
5502 \fi
5503 }

```

\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

```

5504 \newcommand*{\makeglossaries}{%
5505   \@domakeglossaries
5506   {%

```

If automake=immediate setting is on, use the shell escape now.

```
5507 \@@gls@@automake@immediate
```

Define the write used for style file also used for all other output files if savewrites=true.

```
5508 \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.

```
5509 \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
```

```
5510 \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}
```

If \@@gls@extramakeindexopts has been defined, write it:

```
5511 \ifundef\@@gls@extramakeindexopts
```

```
5512 {}%
```

```
5513 {%
```

```
5514 \protected@write\@auxout{}{\string\providecommand
```

```
5515 \string\@gls@extramakeindexopts[1]{}}
```

```
5516 \protected@write\@auxout{}{\string\@gls@extramakeindexopts
```

```
5517 {\@@gls@extramakeindexopts}}%
```

```
5518 }%
```

Write the name of the style file to the aux file (needed by makeglossaries)

```
5519 \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
```

```
5520 \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
```

Iterate through each glossary type and activate it.

```
5521 \@for\@glo@type:=\@glo@types\do{%
```

```
5522 \ifthenelse{equal{\@glo@type}{}}{}}{%
```

```
5523 \@makeglossary{\@glo@type}}%
```

```
5524 }%
```

New glossaries must be created before \makeglossaries so disable \newglossary.

```
5525 \renewcommand*\newglossary[4][]{%
```

```
5526 \PackageError{glossaries}{New glossaries
```

```
5527 must be created before \string\makeglossaries}{You need
```

```
5528 to move \string\makeglossaries\space after all your
```

```
5529 \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).

```
5530 \let\@makeglossary\@gobble
```

```
5531 \let\makeglossaries\relax
```

Disable all commands that have no effect after \makeglossaries

```
5532 \@disable@onlypremakeg
```

Allow see key:

```
5533 \let\gls@checkseeallowed\relax
```

Suppress warning about no \makeglossaries

```
5534 \let\warn@nomakeglossaries\relax
```

Activate warning about missing \printglossary

```

5535 \let\warn@noprintglossary\@makeglossaries@warn@noprintglossary

Declare list parser for \glsdisplaynumberlist
5536 \ifglssavenumberlist
5537 \edef\@gls@dodolistparser{\noexpand\DeclareListParser
5538 {\noexpand\glsnumlistparser}{\delimN}}}%
5539 \@gls@dodolistparser
5540 \fi

Prevent user from also using \makenoidxglossaries
5541 \let\makenoidxglossaries\@no@makeglossaries

Prohibit sort key in printgloss family:
5542 \renewcommand*{\@printgloss@setsort}{\%
5543 \let\@glo@assign@sortkey\@glo@no@assign@sortkey
5544 }%

Check the automake setting:
5545 \ifglssautomake
5546 \renewcommand*{\@gls@doautomake}{\%
5547 \@for\@gls@type:=\@glo@types\do{%
5548 \ifdefempty{\@gls@type}{\}%
5549 {\@gls@automake{\@gls@type}}}%
5550 }%
5551 }%
5552 \fi

Check the sort setting:
5553 \@glo@check@sortallowed\makeglossaries
5554 }%
5555 }

Must occur in the preamble:
5556 \onlypreamble{\makeglossaries}

```

noprintglossary

```

5557 \newcommand\@makeglossaries@warn@noprintglossary{%
5558 \ifdefstring{\@glo@types}{,}%
5559 {%
5560 \GlossariesWarningNoLine{No glossaries have been defined}%
5561 }%
5562 {%
5563 \GlossariesWarningNoLine{No \string\printglossary\space
5564 or \string\printglossaries\space
5565 found. ^^J(Remove \string\makeglossaries\space if you
5566 don't want any glossaries.) ^^JThis document will not
5567 have a glossary}%
5568 }%
5569 }%

```



`\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.

```
5570 \AtEndDocument{%
5571   \warn@nomakeglossaries
5572   \warn@noprintglossary
5573 }
```

`noidxglossaries` Analogous to `\makeglossaries` this activates the commands needed for `\printnoidxglossary`

```
5574 \newcommand*{\makenoidxglossaries}{%
5575   \@domakeglossaries
5576   {%
```

Redefine empty glossary warning:

```
5577   \renewcommand{\@gls@noref@warn}[1]{%
5578     \GlossariesWarning{Empty glossary for
5579       \string\printnoidxglossary[type={##1}].
5580     Rerun may be required (or you may have forgotten to use
5581     commands like \string\gls)}}%
5582   }%
```

Don't escape makeindex/xindy characters:

```
5583   \let\@gls@checkmkidxchars\@gobble
```

Don't escape locations:

```
5584   \glscsclocationsfalse
```

Write glossary information to aux instead of glossary files

```
5585   \let\@do@@@wrglossary\gls@noidxglossary
```

Switch on group headings that use the character code:

```
5586   \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
```

Allow see key:

```
5587   \let\@gls@checkseeallowed\relax
```

Redefine cross-referencing macro:

```
5588   \renewcommand{\@do@seeglossary}[2]{%
5589     \protected@edef\@gls@label{\glsdetoklabel{##1}}%
5590     \protected@write\@auxout{}{%
5591       \string\@gls@reference
5592       {\csname glo@\@gls@label @type\endcsname}%
5593       {\@gls@label}%
5594       {%
5595         \string\glsseeformat##2}%
5596       }%
5597     }%
5598   }%
```

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.

```

5599 \AtBeginDocument
5600 {%
5601   \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
5602 }%

Change warning about no glossaries
5603 \def\warn@noprintglossary{%
5604   \GlossariesWarningNoLine{No \string\printnoidxglossary\space
5605     or \string\printnoidxglossaries ^^J
5606     found. (Remove \string\makenoidxglossaries\space if you
5607     don't want any glossaries.)^^JThis document will not have a glossary}%
5608 }%

Suppress warning about no \makeglossaries
5609 \let\warn@nomakeglossaries\relax

Prevent user from also using \makeglossaries
5610 \let\makeglossaries\@no@makeglossaries

Allow sort key in printgloss family:
5611 \renewcommand*{\@printgloss@setsort}{%
5612   \let\@glo@assign@sortkey\@glo@assign@sortkey

Initialise default sort order:
5613   \def\@glo@sorttype{\@glo@default@sorttype}%
5614 }%

All entries must be defined in the preamble:
5615 \renewcommand*\new@glossaryentry[2]{%
5616   \PackageError{glossaries}{Glossary entries must be
5617     defined in the preamble^^Jwhen you use
5618     \string\makenoidxglossaries}%
5619   {Either move your definitions to the preamble or use
5620     \string\makeglossaries}%
5621 }%

Redefine \glsentrynumberlist
5622 \renewcommand*\glsentrynumberlist[1]{%
5623   \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
5624   \ifdef\@gls@loclist
5625   {%
5626     \glsnoidxloclist{\@gls@loclist}%
5627   }%
5628   {%
5629     ??\glsdoifexists{##1}%
5630     {%
5631       \GlossariesWarning{Missing location list for '##1'. Either
5632         a rerun is required or you haven't referenced the entry}%
5633     }%
5634   }%

```

```

5635 }%
    Redefine \glsdisplaynumberlist
5636 \renewcommand*{\glsdisplaynumberlist}[1]{%
5637   \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
5638   \ifdef\@gls@loclist
5639   {%
5640     \def\@gls@noidxloclist@sep{%
5641       \def\@gls@noidxloclist@sep{%
5642         \def\@gls@noidxloclist@sep{%
5643           \glsnumlistsep
5644         }%
5645       \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
5646     }%
5647   }%
5648   \def\@gls@noidxloclist@finalsep{}%
5649   \def\@gls@noidxloclist@prev{}%
5650   \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
5651   \@gls@noidxloclist@finalsep
5652   \@gls@noidxloclist@prev
5653 }%
5654 {%
5655   ??\glsdoifexists{##1}%
5656   {%
5657     \GlossariesWarning{Missing location list for ‘##1’. Either
5658       a rerun is required or you haven’t referenced the entry}%
5659   }%
5660 }%
5661 }%

```

Provide a generic way of iterating through the number list:

```

5662 \renewcommand*{\glsnumberlistloop}[3]{%
5663   \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
5664   \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
5665   \let\@gls@org@glsseeformat\glsseeformat
5666   \let\glsnoidxdisplayloc##2\relax
5667   \let\glsseeformat##3\relax
5668   \ifdef\@gls@loclist
5669   {%
5670     \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
5671   }%
5672   {%
5673     ??\glsdoifexists{##1}%
5674     {%
5675       \GlossariesWarning{Missing location list for ‘##1’. Either
5676         a rerun is required or you haven’t referenced the entry}%
5677     }%
5678   }%
5679   \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
5680   \let\glsseeformat\@gls@org@glsseeformat

```

5681 }%

Modify sanitize sort function

```
5682 \let\@gls@sanitizesort\@gls@noidx@sanitizesort
5683 \let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
5684 \@gls@noidx@setsanitizesort
```

Check sort option allowed.

```
5685 \@gls@check@sortallowed\makenoidxglossaries
5686 }%
5687 }
```

Preamble-only command:

```
5688 \@onlypreamble{\makenoidxglossaries}
```

snumberlistloop `\glsnumberlistloop{<label>}{<handler>}`

```
5689 \newcommand*{\glsnumberlistloop}[2]{%
5690   \PackageError{glossaries}{\string\glsnumberlistloop\space
5691     only works with \string\makenoidxglossaries}{}%
5692 }
```

listloophandler Handler macro for `\glsnumberlistloop`. (The argument should be in the form `\glsnoidxdisplayloc {<prefix>}{<counter>}{<format>}{<n>}`)

```
5693 \newcommand*{\glsnoidxnumberlistloophandler}[1]{%
5694   #1%
5695 }
```

@makeglossaries Can't use both `\makeglossaries` and `\makenoidxglossaries`

```
5696 \newcommand*{\@no@makeglossaries}{%
5697   \PackageError{glossaries}{You can't use both
5698     \string\makeglossaries\space and \string\makenoidxglossaries}%
5699   {Either use one or other (or none) of those commands but not both
5700     together.}%
5701 }
```

@gls@noref@warn Warning when no instances of `\@gls@reference` found.

```
5702 \newcommand{\@gls@noref@warn}[1]{%
5703   \GlossariesWarning{\string\makenoidxglossaries\space
5704     is required to make \string\printnoidxglossary[type={#1}] work}%
5705 }
```

## 1.14 Writing information to associated files

s@noidxglossary Write the glossary information to the aux file (for the 'noidx' method):

```
5706 \newcommand*{\gls@noidxglossary}{%
5707   \protected@write\@auxout{}{%
```

```

5708   \string\@gls@reference
5709     {\csname glo@\@gls@label @type\endcsname}%
5710     {\@gls@label}%
5711     {\string\glsnoidxdisplayloc
5712       {\@glo@counterprefix}%
5713       {\@gls@counter}%
5714       {\@glsnumberformat}%
5715       {\@glslocref}%
5716     }%
5717   }%
5718 }

```

`\istfile` Deprecated.

```

5719 \providecommand\istfile{\glswrite}

```

At the end of the document, the files should be created if `savewrites=true`.

```

5720 \AtEndDocument{%
5721   \glswritefiles
5722 }

```

`\@glswritefiles` Only write the files if `savewrites=true`.

```

5723 \newcommand*{\@glswritefiles}{%

```

Iterate through all the glossaries.

```

5724   \forallglossaries{\@glo@type}{%

```

Check for empty glossaries (patch provided by Patrick Häcker)

```

5725     \ifcsundef{glo@\@glo@type @filetok}%
5726     {%
5727       \def\gls@tmp{}%
5728     }%
5729     {%
5730       \edef\gls@tmp{\expandafter\the
5731         \csname glo@\@glo@type @filetok\endcsname}%
5732     }%
5733     \ifx\gls@tmp\@empty
5734       \ifx\@glo@type\glsdefaulttype
5735         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
5736           entries.^^JRemember to use package option 'nomain' if
5737 you
5738           don't want to^^Juse the main glossary}%
5739       \else
5740         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
5741           entries}%
5742       \fi
5743     \else
5744       \@glsopenfile{\glswrite}{\@glo@type}%
5745       \immediate\write\glswrite{%
5746         \expandafter\the
5747         \csname glo@\@glo@type @filetok\endcsname}%

```

```

5748         \immediate\closeout\glswrite
5749     \fi
5750 }%
5751 }

```

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`

```

5752 \newcommand*{\gls@glossary}[1]{%
5753   \@gls@glossary{#1}%
5754 }

```

`\@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 `\glspersentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```

5755 \newcommand*{\@gls@glossary}[2]{%
5756   \if@gldebug
5757     \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
5758   \fi
5759 }

```

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

```
5760 \newcommand{\@gls@renewglossary}{%
5761   \gdef\@gls@glossary##1{\@bsphack\beginngroup\gls@wrglossary{##1}}%
5762   \let\@gls@renewglossary\@empty
5763 }
```

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

```
5764 \newcommand*{\gls@wrglossary}[2]{%
5765   \ifglssavewrites
5766     \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
5767     \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
5768       \expandafter{\@gls@tmp~J}%
5769   \else
5770     \ifcsdef{glo@#1@file}%
5771     {%
5772       \expandafter\protected@write\csname glo@#1@file\endcsname{%
5773         \gls@disablepagerefexpansion}{#2}%
5774     }%
5775     {%
5776       \ifignoredglossary{#1}{}%
5777       {%
5778         \GlossariesWarning{No file defined for glossary ‘#1’}%
5779       }%
5780     }%
5781   \fi
5782   \endgroup\@esphack
5783 }
```

\@do@wrglossary

```
5784 \newcommand*{\@do@wrglossary}[1]{%
5785   \glswriteentry{#1}{\@do@wrglossary{#1}}%
5786 }
```

\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.

```
5787 \newcommand*{\glswriteentry}[2]{%
5788   \ifglsexindexonlyfirst
5789     \ifglssused{#1}{}{#2}%
5790   \else
5791     #2%
5792   \fi
5793 }
```

protected@pagefmts List of page formats to be protected against expansion.

```
5794 \newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%  
5795 \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
```

agerefexpansion

```
5796 \newcommand*{\gls@disablepagerefexpansion}{%  
5797 \@for\@gls@this:=\gls@protected@pagefmts\do  
5798 {%  
5799 \expandafter\let\@gls@this\relax  
5800 }%  
5801 }
```

\gls@alphpage

```
5802 \newcommand*{\gls@alphpage}{\@alph\c@page}
```

\gls@Alphpage

```
5803 \newcommand*{\gls@Alphpage}{\@Alph\c@page}
```

\gls@numberpage

```
5804 \newcommand*{\gls@numberpage}{\number\c@page}
```

\gls@arabicpage

```
5805 \newcommand*{\gls@arabicpage}{\@arabic\c@page}
```

\gls@romanpage

```
5806 \newcommand*{\gls@romanpage}{\romannumeral\c@page}
```

\gls@Romanpage

```
5807 \newcommand*{\gls@Romanpage}{\@Roman\c@page}
```

protectedpagefmt

`\glsaddprotectedpagefmt{<cs name>}`

Added a page format to the list of protected page formats. The argument should be the name (without a backslash) of the command that takes a  $\TeX$  register as the argument (`\<csname>\c@page` must be valid).

```
5808 \newcommand*{\glsaddprotectedpagefmt}[1]{%  
5809 \protected@eappto\gls@protected@pagefmts{,\expandonce{\csname gls#1page\endcsname}}%  
5810 \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%  
5811 \protected@eappto\@wrglossarynumberhook{%  
5812 \noexpand\let\expandonce{\csname org@gls#1\endcsname}%  
5813 \expandonce{\csname#1\endcsname}%  
5814 \noexpand\def\expandonce{\csname#1\endcsname}{%  
5815 \noexpand\@wrglossary@pageformat  
5816 \expandonce{\csname gls#1page\endcsname}%  
5817 \expandonce{\csname org@gls#1\endcsname}%  
5818 }%  
5819 }%  
5820 }
```



ssarynumberhook Hook used by \@@do@wrglossary  
5821 \newcommand\*\@wrglossarynumberhook{}

sary@pageformat  
5822 \newcommand{\@wrglossary@pageformat}[3]{%  
5823 \ifx#3\c@page #1\else #2#3\fi  
5824 }

@@do@wrglossary Write the glossary entry in the appropriate format.  
5825 \newcommand\*\@@do@wrglossary}[1]{%  
5826 \ifglsecllocations  
5827 \@@do@esc@wrglossary{#1}%  
5828 \else  
5829 \@@do@noesc@wrglossary{#1}%  
5830 \fi  
5831 }

noesc@wrglossary Write the glossary entry in the appropriate format. The locations don't need to be pre-processed before writing the information to the glossary file, but the prefix still needs to be found.

5832 \newcommand\*\@@do@noesc@wrglossary}[1]{%

Don't fully expand yet.

5833 \expandafter\def\expandafter\@glsloc\ref\expandafter{\the\glsentrycounter}%  
5834 \expandafter\def\expandafter\@glsHloc\ref\expandafter{\the\Hglentrycounter}%

Find the prefix if \@glsHloc\ref and \@glsloc\ref aren't the same.

5835 \ifx\@glsHloc\ref\@glsloc\ref  
5836 \def\@glo@counterprefix{}%  
5837 \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 \the\glsentrycounter and \the\Hglentrycounter at this point, even if it hasn't been updated yet. The page number is not expected to occur in the prefix.)

5838 \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix  
5839 {\@glsloc\ref}\@glsHloc\ref}%  
5840 }%  
5841 \@do@gls@getcounterprefix  
5842 \fi

De-tok label if required.

5843 \protected@edef\@gls@label{\@glsdetoklabel{#1}}%

Write the information to file:

5844 \@@do@wrglossary  
5845 }

owprimitivemods Conditional to determine whether or not \@@do@esc@wrglossary should be allowed to temporarily redefine \the and \number.

5846 \newif\ifglswrallowprimitivemods

5847 \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).

```
5848 \newcommand*{\@@do@esc@wrglossary}[1]{% please read documented code!
5849   \begingroup
```

First a bit of hackery to prevent premature expansion of `\c@page`. Store original definitions (scoped):

```
5850   \let\gls@orgthe\the
5851   \let\gls@orgnumber\number
5852   \let\gls@orgarabic\@arabic
5853   \let\gls@orgromannumeral\romannumeral
5854   \let\gls@orgalph\@alph
5855   \let\gls@orgAlph\@Alph
5856   \let\gls@orgRoman\@Roman
```

Redefine:

```
5857   \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.

```
5858   \def\gls@the##1{%
5859     \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
5860   \def\the{\expandafter\gls@the}%
5861   \def\gls@number##1{%
5862     \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
5863   \def\number{\expandafter\gls@number}%
5864   \fi
5865   \def\@arabic##1{%
5866     \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
5867   \def\romannumeral##1{%
5868     \ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
5869   \def\@Roman##1{%
5870     \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
5871   \def\@alph##1{%
5872     \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
5873   \def\@Alph##1{%
5874     \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
```

Add hook to allow for other number formats:

```
5875   \@wrglossarynumberhook
```

Prevent expansion:

```
5876   \gls@disablepagerefexpansion
```

Now store location in `\@glslocref`:

```

5877 \protected@xdef\@glslocref{\theglsentrycounter}%
5878 \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.

```

5879 \@gls@checkmkidxchars\@glslocref

```

Check if the hyper-location is the same as the location and set the hyper prefix.

```

5880 \expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
5881 \def\@glo@counterprefix{}%
5882 \else
5883 \protected@edef\@glsHlocref{\theglsentrycounter}%
5884 \@gls@checkmkidxchars\@glsHlocref
5885 \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
5886   {\@glslocref}{\@glsHlocref}}%
5887 }%
5888 \@do@gls@getcounterprefix
5889 \fi

```

De-tok label if required

```

5890 \protected@edef\@gls@label{\glsdetoklabel{#1}}%

```

Write the information to file:

```

5891 \@do@wrglossary
5892 }

```

@do@wrglossary

```

5893 \newcommand*{\@do@wrglossary}{%

```

Determine whether to use `xindy` or `makeindex` syntax

```

5894 \ifglsxindy

```

Need to determine if the formatting information starts with a ( or ) indicating a range.

```

5895 \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
5896 \def\@glo@range{}%
5897 \expandafter\if\@glo@prefix(\relax
5898 \def\@glo@range{:open-range}%
5899 \else
5900 \expandafter\if\@glo@prefix)\relax
5901 \def\@glo@range{:close-range}%
5902 \fi
5903 \fi

```

Write to the glossary file using `xindy` syntax.

```

5904 \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
5905 (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)

5906   :locref \string"\@glo@counterprefix}{\@glslocref}\string" %
5907   :attr \string"\@gls@counter\@glo@suffix\string"
5908   \@glo@range
5909 )
5910 }%
5911 \else

```

Convert the format information into the format required for makeindex

```
5912 \set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
5913 {\@glo@counterprefix}%
```

Write to the glossary file using makeindex syntax.

```
5914 \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
5915 \string\glossaryentry{\csname glo@\@gls@label @index\endcsname
5916 \@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
5917 \fi
5918 }
```

`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.)

```
5919 \newcommand*\@gls@getcounterprefix[2]{%
5920 \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
5921 \ifx\@gls@thisloc\@gls@thisHloc
5922 \def\@glo@counterprefix{}%
5923 \else
5924 \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
5925 \def\@glo@tmp{##2}%
5926 \ifx\@glo@tmp\@empty
5927 \def\@glo@counterprefix{}%
5928 \else
5929 \def\@glo@counterprefix{##1}%
5930 \fi
5931 }%
5932 \@gls@get@counterprefix#2.#1\end@getprefix
```

Warn if no prefix can be formed.

```
5933 \ifx\@glo@counterprefix\@empty
5934 \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
5935 prefixing^^Jlocation ‘#1’. You need to modify the
5936 definition of \string\theH\@gls@counter^^Jotherwise you
5937 will get the warning: “name{\@gls@counter.#1}’ has been^^J
5938 referenced but does not exist”}%
5939 \fi
5940 \fi
5941 }
```

## 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.

```
5942 \newcommand{\@do@seeglossary}[2]{%
```

```

5943 \def\@gls@xref{#2}%
5944 \@onelevel@sanitize\@gls@xref
5945 \@gls@checkmkidxchars\@gls@xref
5946 \ifglsxindy
5947   \gls@glossary{\csname glo@#1@type\endcsname}{%
5948     (indexentry
5949       :tkey (\csname glo@#1@index\endcsname)
5950       :xref (\string"\@gls@xref\string")
5951       :attr \string"see\string"
5952     )
5953   }%
5954 \else
5955   \gls@glossary{\csname glo@#1@type\endcsname}{%
5956     \string\glossaryentry{\csname glo@#1@index\endcsname
5957     \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
5958 \fi
5959 }

```

`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```

5960 \def\@gls@fixbraces#1#2#3\@nil{%
5961   \ifx#2[\relax
5962     \@gls@fixbraces#1#2#3\@end@fixbraces
5963   \else
5964     \def#1{{#2#3}}%
5965   \fi
5966 }

```

`@@gls@fixbraces`

```

5967 \def\@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
5968   \def#1{[#2]{#3}}%
5969 }

```

```

\glssee   \glssee{<label>}{<cross-ref list>}
5970 \newrobustcmd*{\glssee}[3][\seename]{%
5971   \@do@seeglossary{#2}{[#1]{#3}}}
5972 \newcommand*{\@glssee}[3][\seename]{%
5973   \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.

```

5974 \newrobustcmd*{\glsseeformat}[3][\seename]{%
5975   \emph{#1} \glsseelist{#2}}

```

`\glsseelist` `\glsseelist{<list>}` formats list of entry labels.

```

5976 \newrobustcmd*{\glsseelist}[1]{%

```

If there is only one item in the list, set the last separator to do nothing.

```

5977   \let\@gls@dolast\relax

```

Don't display separator on the first iteration of the loop

```
5978 \let\@gls@donext\relax
```

Iterate through the labels

```
5979 \@for\@gls@thislabel:=#1\do{%
```

Check if on last iteration of loop

```
5980 \ifx\@xfor@nextelement\@nnil
5981 \@gls@dolast
5982 \else
5983 \@gls@donext
5984 \fi
```

Display the entry for this label. (Expanding label as it's a temporary control sequence that's used elsewhere.)

```
5985 \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
```

Update separators

```
5986 \let\@gls@dolast\glsseelastsep
5987 \let\@gls@donext\glsseesep
5988 }%
5989 }
```

`\glsseelastsep` Separator to use between penultimate and ultimate entries in a cross-referencing list.

```
5990 \newcommand*{\glsseelastsep}{\space\andname\space}
```

`\glsseesep` Separator to use between entries in a cross-referencing list.

```
5991 \newcommand*{\glsseesep}{, }
```

`\glsseeitem` `\glsseeitem{<label>}` formats individual entry in a cross-referencing list.

```
5992 \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.)

```
5993 \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.

```
5994 \newcommand*{\gls@save@numberlist}[1]{%
5995 \ifglssavenumberlist
5996 \toks@{#1}%
```

```

5997 \protected@edef\@do@writeaux@info{%
5998 \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
5999 }%
6000 \@onelevel@sanitize\@do@writeaux@info
6001 \protected@write\@auxout{}\@do@writeaux@info}%
6002 \fi
6003 }

```

`\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.)

```
6004 \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.

```

6005 \ifcsundef{printglossary}{}%
6006 {%

```

If `\printglossary` is already defined, issue a warning and undefine it.

```

6007 \@gls@warnonglossdefined
6008 \undef\printglossary
6009 }

```

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`

```

6010 \newcommand*{\@printgloss@checkexists@noignored}[2]{%
6011 \ifglossaryexists{#1}%
6012 {#2}%
6013 {%
6014 \ifignoredglossary{#1}%
6015 {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
6016 {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
6017 }%
6018 }

```

`\exists@allowignored` For use with `\printunsrtglossary`.

```

6019 \newcommand*{\@printgloss@checkexists@allowignored}[2]{%
6020 \s@ifglossaryexists{#1}%
6021 {#2}%
6022 {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
6023 }

```

`\exists@checkexists`

```
6024 \let\@printgloss@checkexists\@printgloss@checkexists@noignored
```



`\printglossary` has an optional argument. The default value is to set the glossary type to the main glossary.

```
6025 \newcommand*{\printglossary}[1] [type=\glsdefaulttype]{%
6026   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
6027   \@printglossary{#1}{\@print@glossary}%
6028 }
```

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`

```
6029 \newcommand*{\printglossaries}{%
6030   \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
6031 }
```

`\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.

```
6032 \newcommand*{\printnoidxglossary}[1] [type=\glsdefaulttype]{%
6033   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
6034   \@printglossary{#1}{\@print@noidx@glossary}%
6035 }
```

`\printnoidxglossaries` Analogous to `\printglossaries`

```
6036 \newcommand*{\printnoidxglossaries}{%
6037   \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
6038 }
```

`\printgloss@setsort` Initialise to do nothing.

```
6039 \newcommand*{\@printgloss@setsort}{}%
```

`\preglossaryhook`

```
6040 \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.

```
6041 \newcommand{\@printglossary}[2]{%
6042   \def\@glo@type{\glsdefaulttype}%
6043   \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
6044   \printglossary{#1}{\@print@glossary}%
6045 }
```

Set up defaults.

```
6042 \def\@glo@type{\glsdefaulttype}%
6043 \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
6044 \printglossary{#1}{\@print@glossary}%
6045 }
```

```

6044 \def\glossarytoctitle{\glossarytitle}%
6045 \let\org@glossarytitle\glossarytitle

6046 \def\@glossarystyle{%
6047   \ifx\@glossary@default@style\relax
6048     \GlossariesWarning{No default glossary style provided \MessageBreak
6049       for the glossary '@glo@type'. \MessageBreak
6050       Using deprecated fallback. \MessageBreak
6051       To fix this set the style with \MessageBreak
6052       \string\setglossarystyle\space or use the \MessageBreak
6053       style key=value option}%
6054   \fi
6055 }%
6056 \def\gls@dotoc@title{\gls@settoc@title{\@glo@type}}%

  Store current value of \glossaryentrynumbers. (This may be changed via the optional ar-
  gument)
6057 \let\@org@glossaryentrynumbers\glossaryentrynumbers

  Localise the effects of the optional argument
6058 \bgroup

  Activate or deactivate sort key:
6059 \@printgloss@setsort

  Determine settings specified in the optional argument.
6060 \setkeys{printgloss}{#1}%

  Does the glossary exist?
6061 \@printgloss@checkexists{\@glo@type}%
6062 {%

  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)
6063 \ifx\glossarytitle\org@glossarytitle
6064 \else
6065   \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
6066     \glossarytitle
6067 \fi

  Allow a high-level user command to indicate the current glossary
6068 \let\currentglossary\@glo@type

  Enable individual number lists to be suppressed.
6069 \let\org@glossaryentrynumbers\glossaryentrynumbers
6070 \let\glsnonextpages\@glsnonextpages

  Enable individual number list to be activated:
6071 \let\glsnextpages\@glsnextpages

  Enable suppression of description terminators.
6072 \let\nopostdesc\@nopostdesc

```

Set up the entry for the TOC

```
6073 \gls@dotocitle
```

Set the glossary style

```
6074 \@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):

```
6075 \let\gls@org@glossaryentryfield\glossentry
6076 \let\gls@org@glossarysubentryfield\subglossentry
6077 \renewcommand{\glossentry}[1]{%
6078   \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
6079   \gls@org@glossaryentryfield{##1}%
6080 }%
6081 \renewcommand{\subglossentry}[2]{%
6082   \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
6083   \gls@org@glossarysubentryfield{##1}{##2}%
6084 }%
6085 \@gls@preglossaryhook
```

Now do the handler macro that deals with the actual glossary:

```
6086 #2%
6087 }%
```

End the current scope

```
6088 \egroup
```

Reset `\glossaryentrynumbers`

```
6089 \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
```

Suppress warning about no `\printglossary`

```
6090 \global\let\warn@noprintglossary\relax
6091 }
```

`@print@glossary` Internal workings of `\printglossary` dealing with reading the external file.

```
6092 \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.)

```
6093 \makeatletter
```

Input the glossary file, if it exists.

```
6094 \@input@{\jobname.\csname @glo@type\@glo@type @in\endcsname}%
```

If the glossary file doesn't exist, do `\null`. (This ensures that the page is shipped out and all write commands are done.) This might produce an empty page, but at this point the document isn't complete, so it shouldn't matter.

```
6095 \IfFileExists{\jobname.\csname @glo@type\@glo@type @in\endcsname}%
6096 {}%
6097 {\null}%
```

If xindy is being used, need to write the language dependent information to the .aux file for makeglossaries.

```
6098 \ifglxindy
6099 \ifcsundef{@xdy@\@glo@type @language}%
6100 {%
6101 \edef\@do@auxoutstuff{%
6102 \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.

```
6103 \noexpand\immediate\noexpand\write\@auxout{%
6104 \string\providecommand\string\@xdylanguage[2]{}}%
6105 \noexpand\immediate\noexpand\write\@auxout{%
6106 \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
6107 }%
6108 }%
6109 }%
6110 {%
6111 \edef\@do@auxoutstuff{%
6112 \noexpand\AtEndDocument{%
6113 \noexpand\immediate\noexpand\write\@auxout{%
6114 \string\providecommand\string\@xdylanguage[2]{}}%
6115 \noexpand\immediate\noexpand\write\@auxout{%
6116 \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
6117 @language\endcsname}}%
6118 }%
6119 }%
6120 }%
6121 \@do@auxoutstuff
6122 \edef\@do@auxoutstuff{%
6123 \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.

```
6124 \noexpand\immediate\noexpand\write\@auxout{%
6125 \string\providecommand\string\@gls@codepage[2]{}}%
6126 \noexpand\immediate\noexpand\write\@auxout{%
6127 \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
6128 }%
6129 }%
6130 \@do@auxoutstuff
6131 \fi
```

Activate warning if \makeglossaries hasn't been used.

```
6132 \renewcommand*{\@warn@nomakeglossaries}{%
6133 \GlossariesWarningNoLine{\string\makeglossaries\space
6134 hasn't been used,^^Jthe glossaries will not be updated}%
6135 }%
6136 }
```

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`

```

6137 \newcommand*{\@glo@sortentries}[2]{%
6138   \glosortentrieswarning
6139   \def\@glo@sortinglist{}%
6140   \def\@glo@sortinghandler{#1}%

6141   \protected@edef\@glo@type{#2}%
6142   \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
6143   \csdef{\@glsref@#2}{}%
6144   \@for\@this@label:=\@glo@sortinglist\do{%

```

Has this entry already been added?

```

6145   \xifinlistcs{\@this@label}{\@glsref@#2}%
6146   {}%
6147   {%
6148     \listcsxadd{\@glsref@#2}{\@this@label}%
6149   }%
6150   \ifcsdef{\@glo@sortingchildren@{\@this@label}}%
6151   {%
6152     \@glo@addchildren{#2}{\@this@label}%
6153   }%
6154   {}%
6155 }%
6156 }

```

`glo@addchildren`

`\@glo@addchildren{<type>}{<parent>}`

```

6157 \newcommand*{\@glo@addchildren}[2]{%

```

Scope to allow nesting.

```

6158   \bgroup
6159   \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
6160   \@for\@this@childlabel:=\@glo@childlist\do
6161   {%

```

Check this label hasn't already been added.

```

6162     \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
6163     {}%
6164     {%
6165       \listcsxadd{\@glsref@#1}{\@this@childlabel}%
6166     }%

```

Does this child have children?

```

6167      \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
6168      {%
6169      \@glo@addchildren{#1}{\@this@childlabel}%
6170      }%
6171      {%
6172      }%
6173      }%
6174 \egroup
6175 }

```

@do@sortentries

```

6176 \newcommand*{\@glo@do@sortentries}[1]{%
6177   \ifglshasparent{#1}%
6178   {%

```

This entry has a parent, so add it to the child list

```

6179   \protected@edef\@glo@parent{\csuse{glo@glsdetoklabel{#1}@parent}}%
6180   \ifcsundef{@glo@sortingchildren@\@glo@parent}%
6181   {%
6182   \csdef{@glo@sortingchildren@\@glo@parent}{}%
6183   }%
6184   {}%
6185   \expandafter\@glo@sortedinsert
6186   \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%

```

Has the parent been added?

```

6187   \xifinlistcs{\@glo@parent}{@gleref@\@glo@type}%
6188   {%

```

Yes, it has so do nothing.

```

6189   }%
6190   {%

```

No, it hasn't so add it now.

```

6191   \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
6192   }%
6193   }%
6194   {%
6195   \@glo@sortedinsert{\@glo@sortinglist}{#1}%
6196   }%
6197 }

```

lo@sortedinsert

```
\@glo@sortedinsert{<list>}{<entry label>}
```

Insert into list.

```

6198 \newcommand*{\@glo@sortedinsert}[2]{%
6199   \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
6200 }%

```

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

```
6201 \newcommand*{\@glo@sorthandler@word}[2]{%
6202   \letcs\@gls@sort@A{glo\glsdetoklabel{\#1}@sort}%
6203   \letcs\@gls@sort@B{glo\glsdetoklabel{\#2}@sort}%
6204   \edef\glo@do@compare{%
6205     \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
6206     {\expandonce\@gls@sort@B}%
6207     {\expandonce\@gls@sort@A}%
6208   }%
6209   \glo@do@compare
6210 }
```

thandler@letter

```
6211 \newcommand*{\@glo@sorthandler@letter}[2]{%
6212   \letcs\@gls@sort@A{glo\glsdetoklabel{\#1}@sort}%
6213   \letcs\@gls@sort@B{glo\glsdetoklabel{\#2}@sort}%
6214   \edef\glo@do@compare{%
6215     \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
6216     {\expandonce\@gls@sort@B}%
6217     {\expandonce\@gls@sort@A}%
6218   }%
6219   \glo@do@compare
6220 }
```

orthandler@case Case-sensitive sort.

```
6221 \newcommand*{\@glo@sorthandler@case}[2]{%
6222   \letcs\@gls@sort@A{glo\glsdetoklabel{\#1}@sort}%
6223   \letcs\@gls@sort@B{glo\glsdetoklabel{\#2}@sort}%
6224   \edef\glo@do@compare{%
6225     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
6226     {\expandonce\@gls@sort@B}%
6227     {\expandonce\@gls@sort@A}%
6228   }%
6229   \glo@do@compare
6230 }
```

thandler@nocase Case-insensitive sort.

```
6231 \newcommand*{\@glo@sorthandler@nocase}[2]{%
6232   \letcs\@gls@sort@A{glo\glsdetoklabel{\#1}@sort}%
6233   \letcs\@gls@sort@B{glo\glsdetoklabel{\#2}@sort}%
6234   \edef\glo@do@compare{%
6235     \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
6236     {\expandonce\@gls@sort@B}%
6237     {\expandonce\@gls@sort@A}%
6238   }%
```

```

6239 \glo@do@compare
6240 }

```

@sortmacro@word Sort macro for ‘word’

```

6241 \newcommand*{\@glo@sortmacro@word}[1]{%
6242 \ifdefstring{\@glo@default@sorttype}{standard}%
6243 {%
6244 \@glo@sortentries{\@glo@sorthandler@word}{#1}%
6245 }%
6246 {%
6247 \PackageError{glossaries}{Conflicting sort options:^^J
6248 \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
6249 \string\printnoidxglossary[sort=word]}{}%
6250 }%
6251 }

```

ortmacro@letter Sort macro for ‘letter’

```

6252 \newcommand*{\@glo@sortmacro@letter}[1]{%
6253 \ifdefstring{\@glo@default@sorttype}{standard}%
6254 {%
6255 \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
6256 }%
6257 {%
6258 \PackageError{glossaries}{Conflicting sort options:^^J
6259 \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
6260 \string\printnoidxglossary[sort=letter]}{}%
6261 }%
6262 }

```

tmacro@standard Sort macro for ‘standard’. (Use either ‘word’ or ‘letter’ order.)

```

6263 \newcommand*{\@glo@sortmacro@standard}[1]{%
6264 \ifdefstring{\@glo@default@sorttype}{standard}%
6265 {%
6266 \ifcsdef{\@glo@sorthandler@\glsorder}%
6267 {%
6268 \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
6269 }%
6270 {%
6271 \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}%
6272 }%
6273 }%
6274 {%
6275 \PackageError{glossaries}{Conflicting sort options:^^J
6276 \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
6277 \string\printnoidxglossary[sort=standard]}{}%
6278 }%
6279 }

```

@sortmacro@case Sort macro for ‘case’



```

6280 \newcommand*{\@glo@sortmacro@case}[1]{%
6281   \ifdefstring{\@glo@default@sorttype}{standard}%
6282   {%
6283     \@glo@sortentries{\@glo@sorthandler@case}{#1}%
6284   }%
6285   {%
6286     \PackageError{glossaries}{Conflicting sort options:^^J
6287       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
6288       \string\printnoidxglossary[sort=case]}{}}%
6289   }%
6290 }

```

ortmacro@nocase Sort macro for ‘nocase’

```

6291 \newcommand*{\@glo@sortmacro@nocase}[1]{%
6292   \ifdefstring{\@glo@default@sorttype}{standard}%
6293   {%
6294     \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
6295   }%
6296   {%
6297     \PackageError{glossaries}{Conflicting sort options:^^J
6298       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
6299       \string\printnoidxglossary[sort=nocase]}{}}%
6300   }%
6301 }

```

o@sortmacro@def Sort macro for ‘def’. The order of definition is given in \glolist@*type*.

```

6302 \newcommand*{\@glo@sortmacro@def}[1]{%
6303   \def\@glo@sortinglist{%
6304     \forglseentries[#1]{\@gls@thislabel}%
6305     {%
6306       \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
6307       {%
6308         \listead{\@glo@sortinglist}{\@gls@thislabel}%
6309       }%
6310     }%

```

Hasn't been referenced.

```

6311   }%
6312 }%
6313 \cslet{\@glsref@#1}{\@glo@sortinglist}%
6314 }

```

ortmacro@def@do This won't include parent entries that haven't been referenced.

```

6315 \newcommand*{\@glo@sortmacro@def@do}[1]{%
6316   \ifinlistcs{#1}{\@glsref@\@glo@type}%
6317   {}%
6318   {%
6319     \listcsadd{\@glsref@\@glo@type}{#1}%
6320   }%

```

```

6321 \ifcsdef{@glo@sortingchildren@#1}%
6322 {%
6323   \@glo@addchildren{\@glo@type}{#1}%
6324 }%
6325 {}%
6326 }

```

`@sortmacro@use` Sort macro for ‘use’. (No sorting is required, as the entries are already in order of use, so do nothing.)

```

6327 \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.

```

6328 \newcommand*{\@print@noidx@glossary}{%
6329   \ifcsdef{@gls@ref@{\@glo@type}}%
6330   {%

```

Sort the entries:

```

6331   \ifcsdef{@glo@sortmacro@{\@glo@sorttype}}%
6332   {%
6333     \csuse{@glo@sortmacro@{\@glo@sorttype}}{\@glo@type}%
6334   }%
6335   {%
6336     \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
6337   }%

```

Do the glossary heading and preamble

```

6338   \glossarysection[\glossarytoctitle]{\glossarytitle}%
6339   \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.

```

6340   \def\@gls@currentlettergroup{}%
6341   \begin{theglossary}%
6342   \glossaryheader
6343   \glsresetentrylist

```

Iterate through the entries.

```

6344   \forlistcsloop{\@gls@noidx@do}{\@gls@ref@{\@glo@type}}%

```

Finally end the glossary and do the postamble:

```

6345   \end{theglossary}%
6346   \glossarypostamble
6347 }%
6348 {%

```

```

6349 \@gls@noref@warn{\@glo@type}%
6350 }%
6351 }

```

\glo@grabfirst This won't work for UTF-8 and inputenc.

```

6352 \def\glo@grabfirst#1#2\@nil{%
6353 \def\@gls@firsttok{#1}%
6354 \ifdefempty\@gls@firsttok
6355 {%
6356 \def\@glo@thislettergrp{0}%
6357 }%
6358 {%

```

Sanitize it:

```

6359 \@onelevel@sanitize\@gls@firsttok

```

Fetch the first letter:

```

6360 \expandafter\@glo@grabfirst\@gls@firsttok{}\{}\@nil
6361 }%
6362 }

```

\@glo@grabfirst

```

6363 \def\@glo@grabfirst#1#2\@nil{%
6364 \ifdefempty\@glo@thislettergrp
6365 {%
6366 \def\@glo@thislettergrp{glssymbols}%
6367 }%
6368 {%
6369 \count@=\uccode'#1\relax
6370 \ifnum\count@=0\relax
6371 \def\@glo@thislettergrp{glssymbols}%
6372 \else
6373 \ifdefstring\@glo@sorttype{case}%
6374 {%
6375 \count@='#1\relax
6376 }%
6377 {%
6378 }%
6379 \edef\@glo@thislettergrp{\the\count@}%
6380 \fi
6381 }%
6382 }

```

\@gls@noidx@do Handler for list iteration used by \@print@noidx@glossary. The argument is the entry label.  
This only allows one sublevel.

```

6383 \newcommand{\@gls@noidx@do}[1]{%

```

Get this entry's location list

```

6384 \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%

```

Does this entry have a parent?

```
6385 \ifglshasparent{#1}%  
6386 {%
```

Has a parent.

```
6387 \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax  
6388 \ifdefvoid{\@gls@loclist}  
6389 {%  
6390 \subglossentry{\gls@level}{#1}{}}%  
6391 }%  
6392 {%  
6393 \subglossentry{\gls@level}{#1}%  
6394 {%  
6395 \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%  
6396 }%  
6397 }%  
6398 }%  
6399 {%
```

Doesn't have a parent Get this entry's sort key

```
6400 \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
```

Fetch the first letter:

```
6401 \expandafter\glo@grabfirst\@gls@sort{ }\@nil  
6402 \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%  
6403 {}%  
6404 {%
```

Do the group header:

```
6405 \ifdefempty{\@gls@currentlettergroup}{}%  
6406 {%
```

The group skip may start a new scope, so make a global assignment.

```
6407 \global\let\@glo@thislettergrp\@glo@thislettergrp  
6408 \glsgroupskip  
6409 }%  
6410 \glsgroupheading{\@glo@thislettergrp}%  
6411 }%  
  
6412 \global\let\@gls@currentlettergroup\@glo@thislettergrp
```

Do this entry:

```
6413 \ifdefvoid{\@gls@loclist}  
6414 {%  
6415 \glossentry{#1}{}}%  
6416 }%  
6417 {%  
6418 \glossentry{#1}%  
6419 {%  
6420 \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%  
6421 }%  
6422 }%
```

```
6423 }%
6424 }
```

glsnoidxloclist

```
\glsnoidxloclist{<list cs>}
```

Display location list.

```
6425 \newcommand*{\glsnoidxloclist}[1]{%
6426   \def\@gls@noidxloclist@sep{}%
6427   \def\@gls@noidxloclist@prev{}%
6428   \forlistloop{\glsnoidxloclisthandler}{#1}%
6429 }
```

xloclisthandler

Handler for location list iterator.

```
6430 \newcommand*{\glsnoidxloclisthandler}[1]{%
6431   \ifdefstring{\@gls@noidxloclist@prev}{#1}%
6432   {%
6433     }%
6434     {%
6435       \@gls@noidxloclist@sep
6436       #1%
6437       \def\@gls@noidxloclist@sep{\delimN}%
6438       \def\@gls@noidxloclist@prev{#1}%
6439     }%
6440 }
```

Same as previous location so skip.

yloclisthandler

Handler for location list iterator when used with \glsdisplaynumberlist.

```
6441 \newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
6442   \ifdefstring{\@gls@noidxloclist@prev}{#1}%
6443   {%
6444     }%
6445     {%
6446       \@gls@noidxloclist@sep
6447       \@gls@noidxloclist@prev
6448       \def\@gls@noidxloclist@prev{#1}%
6449     }%
6450 }
```

Same as previous location so skip.

noidxdisplayloc

```
\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<location>}
```

Display a location in the location list.

```
6451 \newcommand*{\glsnoidxdisplayloc}[4]{%
6452   \setentrycounter[#1]{#2}%
6453   \csuse{#3}{#4}%
6454 }
```

`\@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.

```
6455 \newcommand*{\@gls@reference}[3]{%
```

Add to label list

```
6456 \glsdoifexistsorwarn{#2}%
6457 {%
6458 \ifcsundef{@glsref@#1}{\csgdef{@glsref@#1}{}}{}%
6459 \ifinlistcs{#2}{@glsref@#1}%
6460 {}%
6461 {\listcsgadd{@glsref@#1}{#2}}}%
```

Add to location list

```
6462 \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
6463 {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
6464 {}%
6465 \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
6466 }%
6467 }
```

The keys that can be used in the optional argument to `\printglossary` or `\printnoidxglossary` are as follows: The type key sets the glossary type.

```
6468 \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`.

```
6469 \define@key{printgloss}{title}{%
6470 \def\glossarytitle{#1}%
6471 \let\gls@dotoc@title\relax
6472 }
```

The toctitle sets the text used for the relevant entry in the table of contents.

```
6473 \define@key{printgloss}{toctitle}{%
6474 \def\glossarytoctitle{#1}%
6475 \let\gls@dotoc@title\relax
6476 }
```

The style key sets the glossary style (but only for the given glossary).

```
6477 \define@key{printgloss}{style}{%
6478 \ifcsundef{@glsstyle@#1}%
6479 {%
6480 \PackageError{glossaries}%
6481 {Glossary style ‘#1’ undefined}{}%
6482 }%
6483 {%
6484 \def\@glossarystyle{\setglossentrycompatibility
6485 \csname @glsstyle@#1\endcsname}%
6486 }%
6487 }
```

The `numberedsection` key determines if this glossary should be in a numbered section.

```

6488 \define@choicekey{printgloss}{numberedsection}%
6489 [\gls@numberedsection@val\gls@numberedsection@nr]%
6490 {false,nolabel,autolabel,nameref}[nolabel]%
6491 {%
6492   \ifcase\gls@numberedsection@nr\relax
6493     \renewcommand*{\@@glossarysecstar}{*}%
6494     \renewcommand*{\@@glossaryseclabel}{}%
6495   \or
6496     \renewcommand*{\@@glossarysecstar}{}%
6497     \renewcommand*{\@@glossaryseclabel}{}%
6498   \or
6499     \renewcommand*{\@@glossarysecstar}{}%
6500     \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
6501   \or
6502     \renewcommand*{\@@glossarysecstar}{*}%
6503     \renewcommand*{\@@glossaryseclabel}{%
6504       \protected@edef\@currentlabelname{\glossarytoctitle}%
6505       \label{\glsautoprefix\@glo@type}}%
6506   \fi
6507 }

```

The `nogroupskip` key determines whether or not there should be a vertical gap between glossary groups.

```

6508 \define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
6509   \csuse{glsnogroupskip#1}%
6510 }

```

The `nopostdot` key has the same effect as the package option of the same name.

```

6511 \define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
6512   \csuse{glsnopostdot#1}%
6513 }

```

**CounterLabelPrefix** Make it easier to redefine the label prefix.

```

6514 \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.

```

6515 \define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
6516   \csuse{glsentrycounter#1}%
6517   \@gls@define@glossaryentrycounter
6518 }

```

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.

```

6519 \define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
6520   \csuse{glssubentrycounter#1}%
6521   \@gls@define@glossarysubentrycounter
6522 }

```

The `nonumberlist` key determines if this glossary should have a number list.

```

6523 \define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
6524   \ifglsnonumberlist
6525     \def\glossaryentrynumbers##1{%
6526   \else
6527     \def\glossaryentrynumbers##1{##1}%
6528 \fi}

```

The `sort` key sets the glossary sort handler (`\printnoidxglossary` only).

```

6529 \define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}

```

`@assign@sortkey` Issue error if used with `\printglossary`

```

6530 \newcommand*{\@glo@no@assign@sortkey}[1]{%
6531   \PackageError{glossaries}{'sort' key not permitted with
6532   \string\printglossary}%
6533   {The 'sort' key may only be used with \string\printnoidxglossary}%
6534 }

```

`@assign@sortkey` For use with `\printnoidxglossary`

```

6535 \newcommand*{\@glo@assign@sortkey}[1]{%
6536   \def\@glo@sorttype{#1}%
6537 }

```

`@glsnonextpages` Suppresses the next number list only. Global assignments required as it may not occur in the same level of grouping as the next numberlist. (For example, if `\glsnonextpages` is placed 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.

```

6538 \newcommand*{\@glsnonextpages}{%
6539   \gdef\glossaryentrynumbers##1{%
6540     \glsresetentrylist
6541   }%
6542 }

```

`\@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 placed 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.

```

6543 \newcommand*{\@glsnextpages}{%
6544   \gdef\glossaryentrynumbers##1{%
6545     ##1\glsresetentrylist}}

```



`sresetentrylist` Resets `\glossaryentrynumbers`

```

6546 \newcommand*{\glsresetentrylist}{%
6547   \global\let\glossaryentrynumbers\org@glossaryentrynumbers}

```

`\glsnonextpages` Outside of `\printglossary` this does nothing.

```

6548 \newcommand*{\glsnonextpages}{}

```

`\glsnextpages` Outside of `\printglossary` this does nothing.

```

6549 \newcommand*{\glsnextpages}{}

```

Process entrycounter and then subentrycounter options (this ensures the sub-counter can pick up the main counter as the master if required):

```

6550 \@gls@define@glossaryentrycounter
6551 \@gls@define@glossarysubentrycounter

```

`subentrycounter` Resets the `glossarysubentry` counter.

```

6552 \newcommand*{\glsresetsubentrycounter}{%
6553   \ifglssubentrycounter
6554     \setcounter{glossarysubentry}{0}%
6555   \fi
6556 }

```

`subentrycounter` Resets the `glossaryentry` counter.

```

6557 \newcommand*{\glsresetentrycounter}{%
6558   \ifglsentrycounter
6559     \setcounter{glossaryentry}{0}%
6560   \fi
6561 }

```

`\glsstepentry` Advance the `glossaryentry` counter if in use. The argument is the label associated with the entry.

```

6562 \newcommand*{\glsstepentry}[1]{%
6563   \ifglsentrycounter
6564     \refstepcounter{glossaryentry}%
6565     \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
6566   \fi
6567 }

```

`glsstepsubentry` Advance the `glossarysubentry` counter if in use. The argument is the label associated with the subentry.

```

6568 \newcommand*{\glsstepsubentry}[1]{%
6569   \ifglssubentrycounter

6570     \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
6571     \refstepcounter{glossarysubentry}%
6572     \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
6573   \fi
6574 }

```

`\glsrefentry` Reference the entry or sub-entry counter if in use, otherwise just do `\gls`.

```
6575 \newcommand*{\glsrefentry}[1]{%
6576   \ifglentrycounter
6577     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
6578   \else
6579     \ifglssubentrycounter
6580       \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
6581     \else
6582       \gls{#1}%
6583     \fi
6584   \fi
6585 }
```

`trycounterlabel` Defines how to display the glossaryentry counter.

```
6586 \newcommand*{\glentrycounterlabel}{%
6587   \ifglentrycounter
6588     \theglossaryentry.\space
6589   \fi
6590 }
```

`trycounterlabel` Defines how to display the glossarysubentry counter.

```
6591 \newcommand*{\glssubentrycounterlabel}{%
6592   \ifglssubentrycounter
6593     \theglossarysubentry)\space
6594   \fi
6595 }
```

`\glentryitem` Step and display glossaryentry counter, if appropriate.

```
6596 \newcommand*{\glentryitem}[1]{%
6597   \ifglentrycounter
6598     \glsstepentry{#1}\glentrycounterlabel
6599   \else
6600     \glsresetsubentrycounter
6601   \fi
6602 }
```

`glssubentryitem` Step and display glossarysubentry counter, if appropriate.

```
6603 \newcommand*{\glssubentryitem}[1]{%
6604   \ifglssubentrycounter
6605     \glsstepsubentry{#1}\glssubentrycounterlabel
6606   \fi
6607 }
```

`theglossary` If the `theglossary` environment has already been defined, a warning will be issued. This environment should be redefined by glossary styles.

```
6608 \ifcsundef{theglossary}%
6609 {%
6610   \newenvironment{theglossary}{}{}}%
```

```

6611 }%
6612 {%
6613   \@gls@warnontheGLOSSdefined
6614   \renewenvironment{theglossary}{-}{-}%
6615 }

```

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`

```

6616 \newcommand*{\glossaryheader}{}

```

`\glstarget` `\glstarget{<label>}{<name>}`

Provide user interface to `\glstarget` to make it easier to modify the glossary style in the document.

```

6617 \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>}`

```

6618 \providecommand*{\compatibleglossentry}[2]{%
6619   \toks0{#2}%
6620   \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
6621     {\noexpand\glsnamefont
6622       {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
6623     {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
6624     {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
6625     {\the\toks0}}%
6626   }%
6627   \@do@glossentry
6628 }

```

`\glossentryname`

```

6629 \newcommand*{\glossentryname}[1]{%
6630   \glsdoifexistsorwarn{#1}%
6631   {%
6632     \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
6633     \expandafter\glsnamefont\expandafter{\glo@name}%

```

```

6634 }%
6635 }

```

\Glossentryname

```

6636 \newcommand*{\Glossentryname}[1]{%
6637   \glsdoifexistsorwarn{#1}%
6638   {%
6639     \glsnamefont{\Glsentryname{#1}}%
6640   }%
6641 }

```

\glossentrydesc

```

6642 \newcommand*{\glossentrydesc}[1]{%
6643   \glsdoifexistsorwarn{#1}%
6644   {%
6645     \glsentrydesc{#1}%
6646   }%
6647 }

```

\Glossentrydesc

```

6648 \newcommand*{\Glossentrydesc}[1]{%
6649   \glsdoifexistsorwarn{#1}%
6650   {%
6651     \Glsentrydesc{#1}%
6652   }%
6653 }

```

\glossentrysymbol

```

6654 \newcommand*{\glossentrysymbol}[1]{%
6655   \glsdoifexistsorwarn{#1}%
6656   {%
6657     \glsentrysymbol{#1}%
6658   }%
6659 }

```

\Glossentrysymbol

```

6660 \newcommand*{\Glossentrysymbol}[1]{%
6661   \glsdoifexistsorwarn{#1}%
6662   {%
6663     \Glsentrysymbol{#1}%
6664   }%
6665 }

```

\subglossentry

\subglossentry{<level>}{<label>}{<page-list>}

```

6666 \providecommand*{\compatiblesubglossentry}[3]{%
6667   \toks@{#3}%

```

```

6668 \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
6669   {#2}}%
6670   {\noexpand\glsnamefont
6671     {\expandafter\expandonce\csname glo@#2@name\endcsname}}}%
6672   {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
6673   {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
6674   {\the\toks@}}%
6675   }%
6676 \@do@subglossentry
6677 }

```

rycompatibility

```

6678 \newcommand*\setglossentrycompatibility{%
6679   \let\glossentry\compatibleglossentry
6680   \let\subglossentry\compatiblesubglossentry
6681 }
6682 \setglossentrycompatibility

```

ssaryentryfield

```

\glossaryentryfield{\langle label \rangle}{\langle name \rangle}{\langle description \rangle}{\langle symbol \rangle}
{\langle page-list \rangle}

```

This command formerly governed how each entry row should be formatted in the glossary. Now deprecated.

```

6683 \newcommand{\glossaryentryfield}[5]{%
6684   \GlossariesWarning
6685   {Deprecated use of \string\glossaryentryfield.^^J
6686     I recommend you change to \string\glossentry.^^J
6687     If you've just upgraded, try removing your gls auxiliary
6688     files^^J and recompile}%
6689   \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}

```

rysubentryfield

```

\glossarysubentryfield{\langle level \rangle}{\langle label \rangle}{\langle name \rangle}{\langle description \rangle}{\langle symbol \rangle}
{\langle page-list \rangle}

```

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.)

```

6690 \newcommand*\glossarysubentryfield}[6]{%
6691   \GlossariesWarning
6692   {Deprecated use of \string\glossarysubentryfield.^^J
6693     I recommend you change to \string\subglossentry.^^J
6694     If you've just upgraded, try removing your gls auxiliary
6695     files^^J and recompile}%
6696   \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`

```
6697 \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`

```
6698 \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`

```
6699 \newcommand*{\glsgetgrouptitle}[1]{%
6700   \@gls@getgrouptitle{#1}{\@gls@grptitle}%
6701   \@gls@grptitle
6702 }
```

`\@gls@getgrouptitle`

Gets the group title specified by the label (first argument) and stores in the second argument, which must be a control sequence.

```
6703 \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.

```

6704 \dtl@ifsingle{#1}%
6705 {%
6706   \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
6707 }%
6708 {%
6709   \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
6710               or test{\ifstrequal{#1}{glsnumbers}}}%
6711   {%
6712     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
6713   }%
6714   {%
6715     \def#2{#1}%
6716   }%
6717 }%
6718 }

```

`x@getgrouptitle` Version for the no-indexing app option:

```

6719 \newcommand*{\@gls@noidx@getgrouptitle}[2]{%
6720   \DTLifint{#1}%
6721   {\edef#2{\char#1\relax}}%
6722   {%
6723     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
6724   }%
6725 }

```

`\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`.

`lsgetgrouplabel`

```

6726 \newcommand*{\glsgetgrouplabel}[1]{%
6727 \ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
6728 \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`

```

6729 \newcommand*{\setentrycounter}[2][ ]{%
6730   \def\@glo@counterprefix{#1}%
6731   \ifx\@glo@counterprefix\empty
6732     \def\@glo@counterprefix{.}%
6733   \else

```

```

6734 \def\@glo@counterprefix{.#1.}%
6735 \fi
6736 \def\glsentrycounter{#2}%
6737 }

```

The current glossary style can be set using `\setglossarystyle{<style>}`.

`\setglossarystyle`

```

6738 \newcommand*{\setglossarystyle}[1]{%
6739 \ifcsundef{@glsstyle@#1}%
6740 {%
6741 \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
6742 }%
6743 {%
6744 \csname @glsstyle@#1\endcsname
6745 }%

```

Set the default style if it's not already set.

```

6746 \ifx\@glossary@default@style\relax
6747 \protected@edef\@glossary@default@style{#1}%
6748 \fi
6749 }

```

`\glossarystyle`

```

6750 \newcommand*{\glossarystyle}[1]{%
6751 \ifcsundef{@glsstyle@#1}%
6752 {%
6753 \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
6754 }%
6755 {%
6756 \GlossariesWarning
6757 {Deprecated command \string\glossarystyle.^~J
6758 I recommend you switch to \string\setglossarystyle\space unless
6759 you want to maintain backward compatibility}%
6760 \setglossentrycompatibility
6761 \csname @glsstyle@#1\endcsname

6762 \ifcsdef{@glscompstyle@#1}%
6763 {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
6764 {}%
6765 }%

```

Set the default style if it isn't already set so that `\printglossary` can warn if the fallback style is in use.

```

6766 \ifx\@glossary@default@style\relax
6767 \protected@edef\@glossary@default@style{#1}%
6768 \fi
6769 }

```

`\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.

```
6770 \newcommand{\newglossarystyle}[2]{%
6771   \ifcsundef{@glsstyle@#1}%
6772   {%
6773     \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
6774   }%
6775   {%
6776     \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
6777   }%
6778 }
```

`\ewglossarystyle` Code for this macro supplied by Marco Daniel.

```
6779 \newcommand{\renewglossarystyle}[2]{%
6780   \ifcsundef{@glsstyle@#1}%
6781   {%
6782     \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
6783   }%
6784   {%
6785     \csdef{@glsstyle@#1}{#2}%
6786   }%
6787 }
```

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`

```
6788 \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 `\glslinknumber`. 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`

```
6789 \ifcsundef{hyperlink}%
6790 {%
6791   \def\glshypernumber#1{#1}%
6792 }%
6793 {%
6794   \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
6795 }
```

`@glshypernumber` This code was provided by Heiko Oberdiek to allow material to be attached to the location.

```
6796 \def\@glshypernumber#1\nohyperpage#2#3\@nil{%
6797   \ifx\#1\%
6798   \else
6799     \@delimR#1\delimR\delimR\%
6800   \fi
6801   \ifx\#2\%
6802   \else
6803     #2%
6804   \fi
6805   \ifx\#3\%
6806   \else
6807     \@glshypernumber#3\@nil
6808   \fi
6809 }
```

`\@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`

```
6810 \def\@delimR#1\delimR #2\delimR #3\%
6811 \ifx\#2\%
6812   \@delimN{#1}%
6813 \else
6814   \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
6815 \fi}
```

`\@delimN` displays a list of individual numbers, instead of a range:

`\@delimN`

```
6816 \def\@delimN#1{\@delimN#1\delimN \delimN\%
6817 \def\@delimN#1\delimN #2\delimN#3\%
6818 \ifx\#3\%
6819   \@gls@numberlink{#1}%
6820 \else
6821   \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
6822 \fi
6823 }
```

The following code is modified from hyperref's \HyInd@pagelink where the name of the counter being used is given by \@gls@counter.

```

6824 \def\@gls@numberlink#1{%
6825 \begingroup
6826 \toks@={}%
6827 \@gls@removespaces#1 \@nil
6828 \endgroup}

6829 \def\@gls@removespaces#1 #2\@nil{%
6830 \toks@=\expandafter{\the\toks@#1}%
6831 \ifx\#2\%

6832 \edef\@gls@x{\the\toks@}%
6833 \ifx\@gls@x\empty
6834 \else

6835 \hyperlink{\glstrycounter\@glo@counterprefix\the\toks@}%
6836 {\the\toks@}%
6837 \fi
6838 \else
6839 \@gls@ReturnAfterFi{%
6840 \@gls@removespaces#2\@nil
6841 }%
6842 \fi
6843 }
6844 \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

```
6845 \newcommand*\hyperrm[1]{\textrm{\glshypernumber{#1}}}
```

\hypersf

```
6846 \newcommand*\hypersf[1]{\textsf{\glshypernumber{#1}}}
```

\hypertt

```
6847 \newcommand*\hypertt[1]{\texttt{\glshypernumber{#1}}}
```

\hyperbf

```
6848 \newcommand*\hyperbf[1]{\textbf{\glshypernumber{#1}}}
```

\hypermd

```
6849 \newcommand*\hypermd[1]{\textmd{\glshypernumber{#1}}}
```

\hyperit

```
6850 \newcommand*\hyperit[1]{\textit{\glshypernumber{#1}}}
```

`\hypersl`

```
6851 \newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
```

`\hyperup`

```
6852 \newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
```

`\hypersc`

```
6853 \newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
```

`\hyperemph`

```
6854 \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.

```
6855 \newcommand{\oldacronym}[4][\gls@label]{%
6856   \def\gls@label{#2}%
6857   \newacronym[#4]{#1}{#2}{#3}%
6858   \ifcsundef{xspace}%
6859     {%
6860       \expandafter\edef\csname#1\endcsname{%
6861         \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
6862       }%
6863     }%
6864     {%
6865       \expandafter\edef\csname#1\endcsname{%
6866         \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
6867           \noexpand\gls{#1}\noexpand\xspace}%
6868       }%
6869     }%
6870 }
```

```
\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`

```
6871 \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.

```
6872 \newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
```

If `garamondx` has been loaded, need to use `\textulc` instead of `\textup`.

`\glstextup`

```
6873 \newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
```

The following are defined for compatibility with version 2.07 and earlier.

`\glsshortkey`

```
6874 \newcommand*{\glsshortkey}{short}
```

`\glsshortpluralkey`

```
6875 \newcommand*{\glsshortpluralkey}{shortplural}
```

`\glslongkey`

```
6876 \newcommand*{\glslongkey}{long}
```

`\glslongpluralkey`

```
6877 \newcommand*{\glslongpluralkey}{longplural}
```

`\acrfull` Full form of the acronym.

```
6878 \newrobustcmd*{\acrfull}{\@gls@hyp@opt\@ns@acrfull}
```

```
6879 \newcommand*\@ns@acrfull[2] [] {%
```

```
6880 \new@ifnextchar[{\@acrfull{#1}{#2}}%
```

```
6881 \@acrfull{#1}{#2} []}%
```

```
6882 }
```

`\@acrfull` Low-level macro:

```

6883 \def\@acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
6884   \acrfullfmt{#1}{#2}{#3}%
6885 }

```

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.

```

6886 \newcommand*\acrfullfmt[3]{%
6887   \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
6888 }

```

`\acrlinkfullformat` Format for full links like `\acrfull`. Syntax: `\acrlinkfullformat{<long cs>}{<short cs>}{<options>}{<label>}{<insert>}`

```

6889 \newcommand*\acrlinkfullformat[5]{%
6890   \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
6891 }

```

`\acrfullformat` Default full form is *<long>* (*<short>*).

```

6892 \newcommand*\acrfullformat[2]{#1\glsspace(#2)}

```

`\glsspace` Robust space to ensure it's written to the `.glsdefs` file.

```

6893 \newrobustcmd*\glsspace{\space}

```

Default format for full acronym

`\Acrfull`

```

6894 \newrobustcmd*\Acrfull{\@gls@hyp@opt\ns@Acrfull}

```

```

6895 \newcommand*\ns@Acrfull[2][]{%
6896   \new@ifnextchar[\@Acrfull{#1}{#2}}%
6897   {\@Acrfull{#1}{#2}[]}%
6898 }

```

Low-level macro:

```

6899 \def\@Acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
6900   \Acrfullfmt{#1}{#2}{#3}%
6901 }

```

`\Acrfullfmt` First letter upper case full format.

```

6902 \newcommand*\Acrfullfmt[3]{%
6903   \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
6904 }

```

\ACRfull

```
6905 \newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
```

```
6906 \newcommand*\ns@ACRfull[2] [] {%
```

```
6907   \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
```

```
6908       {\@ACRfull{#1}{#2} []}%
```

```
6909 }
```

Low-level macro:

```
6910 \def\@ACRfull#1#2[#3] {%
```

Make it easier for acronym styles to change this:

```
6911   \ACRfullfmt{#1}{#2}{#3}%
```

```
6912 }
```

\ACRfullfmt All upper case full format.

```
6913 \newcommand*{\ACRfullfmt}[3] {%
```

```
6914   \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
```

```
6915 }
```

Plural:

\acrfullpl

```
6916 \newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
```

```
6917 \newcommand*\ns@acrfullpl[2] [] {%
```

```
6918   \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
```

```
6919       {\@acrfullpl{#1}{#2} []}%
```

```
6920 }
```

Low-level macro:

```
6921 \def\@acrfullpl#1#2[#3] {%
```

Make it easier for acronym styles to change this:

```
6922   \acrfullplfmt{#1}{#2}{#3}%
```

```
6923 }
```

\acrfullplfmt No case change plural full format.

```
6924 \newcommand*{\acrfullplfmt}[3] {%
```

```
6925   \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
```

```
6926 }
```

\Acrfullpl

```
6927 \newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
```

```
6928 \newcommand*\ns@Acrfullpl[2] [] {%
```

```
6929   \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
```

```
6930       {\@Acrfullpl{#1}{#2} []}%
```

```
6931 }
```

Low-level macro:

```
6932 \def\@Acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
6933 \Acrfullplfmt{#1}{#2}{#3}%  
6934 }
```

`\Acrfullplfmt` First letter upper case plural full format.

```
6935 \newcommand*\@Acrfullplfmt}[3]{%  
6936 \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%  
6937 }
```

`\ACRfullpl`

```
6938 \newrobustcmd*\@ACRfullpl{\@gls@hyp@opt\@ns@ACRfullpl}  
  
6939 \newcommand*\@ns@ACRfullpl[2][ ]{%  
6940 \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%  
6941 {\@ACRfullpl{#1}{#2}[ ]}%  
6942 }
```

Low-level macro:

```
6943 \def\@ACRfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
6944 \ACRfullplfmt{#1}{#2}{#3}%  
6945 }
```

`\ACRfullplfmt` All upper case plural full format.

```
6946 \newcommand*\@ACRfullplfmt}[3]{%  
6947 \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%  
6948 }
```

## 1.18 Predefined acronym styles

`\acronymfont` This is only used with the additional acronym styles:

```
6949 \newcommand{\acronymfont}[1]{#1}
```

`\firstacronymfont` This is only used with the additional acronym styles:

```
6950 \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.

```
6951 \newcommand*\acrnameformat[2]{\acronymfont{#1}}
```

Define some tokens used by `\newacronym`:

`\glskeylisttok`

```
6952 \newtoks\glskeylisttok
```



```

\glslabeltok
6953 \newtoks\glslabeltok

\glsshorttok
6954 \newtoks\glsshorttok

\gslongtok
6955 \newtoks\gslongtok

\newacronymhook  Provide a hook for \newacronym:
6956 \newcommand*{\newacronymhook}{}

\genericNewAcronym  New improved version of setting the acronym style.
6957 \newcommand*{\SetGenericNewAcronym}{%
    Change the behaviour of \Glsentryname to workaround expansion issues that cause a prob-
    lem for \makefirstuc
6958 \let\@Gls@entryname\@Gls@acentryname
    Change the way acronyms are defined:
6959 \renewcommand{\newacronym}[4][\{%
6960 \ifdefempty{\@glsacronymlists}%
6961 {%
6962 \def\@glo@type{\acronymtype}%
6963 \setkeys{glossentry}{##1}%
6964 \DeclareAcronymList{\@glo@type}%
6965 }%
6966 }%
6967 \glskeylisttok{##1}%
6968 \glslabeltok{##2}%
6969 \glsshorttok{##3}%
6970 \gslongtok{##4}%
6971 \newacronymhook
6972 \protected@edef\@do@newglossaryentry{%
6973 \noexpand\newglossaryentry{\the\glslabeltok}%
6974 {%
6975 type=\acronymtype,%
6976 name={\expandonce{\acronymentry{##2}}},%
6977 sort={\acronymssort{\the\glsshorttok}{\the\gslongtok}},%
6978 text={\the\glsshorttok},%
6979 short={\the\glsshorttok},%
6980 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
6981 long={\the\gslongtok},%
6982 longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
6983 \GenericAcronymFields,%
6984 \the\glskeylisttok
6985 }%
6986 }%
6987 \@do@newglossaryentry
6988 }%

```

Make sure that `\acrfull` etc reflects the new style:

```

6989 \renewcommand*{\acrfullfmt}[3]{%
6990   \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
6991 \renewcommand*{\Acrfullfmt}[3]{%
6992   \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
6993 \renewcommand*{\ACRfullfmt}[3]{%
6994   \glslink[##1]{##2}{%
6995     \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}}}%
6996 \renewcommand*{\acrfullplfmt}[3]{%
6997   \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
6998 \renewcommand*{\Acrfullplfmt}[3]{%
6999   \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
7000 \renewcommand*{\ACRfullplfmt}[3]{%
7001   \glslink[##1]{##2}{%
7002     \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}}}%

```

Make sure that `\glentryfull` etc reflects the new style:

```

7003 \renewcommand*{\glentryfull}[1]{\genacrfullformat{##1}{}}%
7004 \renewcommand*{\Glentryfull}[1]{\Genacrfullformat{##1}{}}%
7005 \renewcommand*{\glentryfullpl}[1]{\genplacrfullformat{##1}{}}%
7006 \renewcommand*{\Glentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
7007 }

```

`\GenericAcronymFields` Fields used by `\SetGenericNewAcronym` that can be changed by the acronym style.

```

7008 \newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}

```

`\acronymentry` `\acronymentry{<label>}`

Display style for the name field in the list of acronyms.

```

7009 \newcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{#1}}}

```

`\acronymsort` `\acronymsort{<short>}{<long>}`

Default sort format for acronyms.

```

7010 \newcommand*{\acronymsort}[2]{#1}

```

`\setacronymstyle` `\setacronymstyle{<style name>}`

```

7011 \newcommand*{\setacronymstyle}[1]{%
7012   \ifcsundef{@glsacr@dispstyle@#1}
7013   {%
7014     \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
7015   }%
7016   {%
7017     \ifdefempty{\@glsacronymlists}%

```

```

7018   {%
7019     \DeclareAcronymList{\acronymtype}%
7020   }%
7021   {%
7022     \SetGenericNewAcronym
7023     \GlsUseAcrStyleDefs{#1}%
7024     \@for\@gls@type:=\@glsacronymlists\do{%
7025       \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
7026     }%
7027   }%
7028 }

```

`\newacronymstyle`

`\newacronymstyle{<style name>}{<entry format definition>}{<display definitions>}`

Defines a new acronym style called <style name>.

```

7029 \newcommand*{\newacronymstyle}[3]{%
7030   \ifcsdef{@glsacr@dispstyle@#1}%
7031   {%
7032     \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
7033   }%
7034   {%
7035     \csdef{@glsacr@dispstyle@#1}{#2}%
7036     \csdef{@glsacr@styledefs@#1}{#3}%
7037   }%
7038 }

```

`\newacronymstyle` Redefines the given acronym style.

```

7039 \newcommand*{\renewacronymstyle}[3]{%
7040   \ifcsdef{@glsacr@dispstyle@#1}%
7041   {%
7042     \csdef{@glsacr@dispstyle@#1}{#2}%
7043     \csdef{@glsacr@styledefs@#1}{#3}%
7044   }%
7045   {%
7046     \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
7047   }%
7048 }

```

`\rEntryDispStyle`

```

7049 \newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}

```

`\UseAcrStyleDefs`

```

7050 \newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}

```

Predefined acronym styles:

long-short    *<long>* (*<short>*) acronym style.

```
7051 \newacronymstyle{long-short}%
7052 {%
    Check for long form in case this is a mixed glossary.
7053   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
7054 }%
7055 {%
7056   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
7057   \renewcommand*{\genacrfullformat}[2]{%
7058     \glentrylong{##1}##2\space
7059     (\protect\firstacronymfont{\glentryshort{##1}})%
7060   }%
7061   \renewcommand*{\Genacrfullformat}[2]{%
7062     \Glentrylong{##1}##2\space
7063     (\protect\firstacronymfont{\glentryshort{##1}})%
7064   }%
7065   \renewcommand*{\genplacrfullformat}[2]{%
7066     \glentrylongpl{##1}##2\space
7067     (\protect\firstacronymfont{\glentryshortpl{##1}})%
7068   }%
7069   \renewcommand*{\Genplacrfullformat}[2]{%
7070     \Glentrylongpl{##1}##2\space
7071     (\protect\firstacronymfont{\glentryshortpl{##1}})%
7072   }%
7073   \renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
7074   \renewcommand*{\acronymsort}[2]{##1}%
7075   \renewcommand*{\acronymfont}[1]{##1}%
7076   \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
7077   \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
7078 }
```

long-sp-short    Similar to the previous style but allows the space between the long and short form to be customized.

```
7079 \newacronymstyle{long-sp-short}%
7080 {%
    Check for long form in case this is a mixed glossary.
7081   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
7082 }%
7083 {%
7084   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
7085   \renewcommand*{\genacrfullformat}[2]{%
7086     \glentrylong{##1}##2\glsacspace{##1}%
7087     (\protect\firstacronymfont{\glentryshort{##1}})%
7088   }%
7089   \renewcommand*{\Genacrfullformat}[2]{%
7090     \Glentrylong{##1}##2\glsacspace{##1}%
7091     (\protect\firstacronymfont{\glentryshort{##1}})%
7092   }%
7093 }
```

```

7093 \renewcommand*{\genplacrfullformat}[2]{%
7094   \glentrylongpl{##1}##2\glsacspace{##1}%
7095   (\protect\firstacronymfont{\glentryshortpl{##1}})%
7096 }%
7097 \renewcommand*{\Genplacrfullformat}[2]{%
7098   \Glsentrylongpl{##1}##2\glsacspace{##1}%
7099   (\protect\firstacronymfont{\glentryshortpl{##1}})%
7100 }%
7101 \renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
7102 \renewcommand*{\acronymsort}[2]{##1}%
7103 \renewcommand*{\acronymfont}[1]{##1}%
7104 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
7105 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
7106 }

```

`\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.

```

7107 \newcommand*{\glsacspace}[1]{%
7108   \settowidth{\dimen@}{(\firstacronymfont{\glentryshort{##1}})}%
7109   \ifdim\dimen@<3em~\else\space\fi
7110 }

```

`short-long` *(short)* (*long*) acronym style.

```

7111 \newacronymstyle{short-long}%
7112 {%
  Check for long form in case this is a mixed glossary.
7113   \ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
7114 }%
7115 {%
7116   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
7117   \renewcommand*{\genacrfullformat}[2]{%
7118     \protect\firstacronymfont{\glentryshort{##1}}##2\space
7119     (\glentrylong{##1})%
7120   }%
7121   \renewcommand*{\Genacrfullformat}[2]{%
7122     \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
7123     (\glentrylong{##1})%
7124   }%
7125   \renewcommand*{\genplacrfullformat}[2]{%
7126     \protect\firstacronymfont{\glentryshortpl{##1}}##2\space
7127     (\glentrylongpl{##1})%
7128   }%
7129   \renewcommand*{\Genplacrfullformat}[2]{%
7130     \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
7131     (\glentrylongpl{##1})%
7132   }%
7133   \renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
7134   \renewcommand*{\acronymsort}[2]{##1}%

```

```

7135 \renewcommand*{\acronymfont}[1]{##1}%
7136 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
7137 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
7138 }

```

long-sc-short    *<long>* (\textsc{<short>}) acronym style.

```

7139 \newacronymstyle{long-sc-short}%
7140 {%
7141   \GlsUseAcrEntryDisplayStyle{long-short}%
7142 }%
7143 {%
7144   \GlsUseAcrStyleDefs{long-short}%
7145   \renewcommand{\acronymfont}[1]{\textsc{##1}}%
7146   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7147 }

```

long-sm-short    *<long>* (\textsmaller{<short>}) acronym style.

```

7148 \newacronymstyle{long-sm-short}%
7149 {%
7150   \GlsUseAcrEntryDisplayStyle{long-short}%
7151 }%
7152 {%
7153   \GlsUseAcrStyleDefs{long-short}%
7154   \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
7155   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7156 }

```

sc-short-long    *<short>* (\textsc{<long>}) acronym style.

```

7157 \newacronymstyle{sc-short-long}%
7158 {%
7159   \GlsUseAcrEntryDisplayStyle{short-long}%
7160 }%
7161 {%
7162   \GlsUseAcrStyleDefs{short-long}%
7163   \renewcommand{\acronymfont}[1]{\textsc{##1}}%
7164   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7165 }

```

sm-short-long    *<short>* (\textsmaller{<long>}) acronym style.

```

7166 \newacronymstyle{sm-short-long}%
7167 {%
7168   \GlsUseAcrEntryDisplayStyle{short-long}%
7169 }%
7170 {%
7171   \GlsUseAcrStyleDefs{short-long}%
7172   \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
7173   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7174 }

```

long-short-desc    *<long>* (*{<short>}*) acronym style that has an accompanying description (which the user needs to supply).

```

7175 \newacronymstyle{long-short-desc}%
7176 {%
7177   \GlsUseAcrEntryDisplayStyle{long-short}%
7178 }%
7179 {%
7180   \GlsUseAcrStyleDefs{long-short}%
7181   \renewcommand*{\GenericAcronymFields}{}%
7182   \renewcommand*{\acronymsort}[2]{##2}%
7183   \renewcommand*{\acronymentry}[1]{%
7184     \glstrylong{##1}\space (\acronymfont{\glstryshort{##1}})}%
7185 }

```

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.

```

7186 \newacronymstyle{long-sp-short-desc}%
7187 {%
7188   \GlsUseAcrEntryDisplayStyle{long-sp-short}%
7189 }%
7190 {%
7191   \GlsUseAcrStyleDefs{long-sp-short}%
7192   \renewcommand*{\GenericAcronymFields}{}%
7193   \renewcommand*{\acronymsort}[2]{##2}%
7194   \renewcommand*{\acronymentry}[1]{%
7195     \glstrylong{##1}\glsacspace{##1}(\acronymfont{\glstryshort{##1}})}%
7196 }

```

g-sc-short-desc    *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7197 \newacronymstyle{long-sc-short-desc}%
7198 {%
7199   \GlsUseAcrEntryDisplayStyle{long-sc-short}%
7200 }%
7201 {%
7202   \GlsUseAcrStyleDefs{long-sc-short}%
7203   \renewcommand*{\GenericAcronymFields}{}%
7204   \renewcommand*{\acronymsort}[2]{##2}%
7205   \renewcommand*{\acronymentry}[1]{%
7206     \glstrylong{##1}\space (\acronymfont{\glstryshort{##1}})}%
7207 }

```

g-sm-short-desc    *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7208 \newacronymstyle{long-sm-short-desc}%
7209 {%
7210   \GlsUseAcrEntryDisplayStyle{long-sm-short}%
7211 }%

```

```

7212 {%
7213   \GlsUseAcrStyleDefs{long-sm-short}%
7214   \renewcommand*{\GenericAcronymFields}{}%
7215   \renewcommand*{\acronymsort}[2]{##2}%
7216   \renewcommand*{\acronymentry}[1]{%
7217     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7218 }

```

short-long-desc    *<short>* (*<long>*) acronym style that has an accompanying description (which the user needs to supply).

```

7219 \newacronymstyle{short-long-desc}%
7220 {%
7221   \GlsUseAcrEntryDisplayStyle{short-long}%
7222 }%
7223 {%
7224   \GlsUseAcrStyleDefs{short-long}%
7225   \renewcommand*{\GenericAcronymFields}{}%
7226   \renewcommand*{\acronymsort}[2]{##2}%
7227   \renewcommand*{\acronymentry}[1]{%
7228     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7229 }

```

short-long-desc    *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7230 \newacronymstyle{sc-short-long-desc}%
7231 {%
7232   \GlsUseAcrEntryDisplayStyle{sc-short-long}%
7233 }%
7234 {%
7235   \GlsUseAcrStyleDefs{sc-short-long}%
7236   \renewcommand*{\GenericAcronymFields}{}%
7237   \renewcommand*{\acronymsort}[2]{##2}%
7238   \renewcommand*{\acronymentry}[1]{%
7239     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7240 }

```

short-long-desc    *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

7241 \newacronymstyle{sm-short-long-desc}%
7242 {%
7243   \GlsUseAcrEntryDisplayStyle{sm-short-long}%
7244 }%
7245 {%
7246   \GlsUseAcrStyleDefs{sm-short-long}%
7247   \renewcommand*{\GenericAcronymFields}{}%
7248   \renewcommand*{\acronymsort}[2]{##2}%
7249   \renewcommand*{\acronymentry}[1]{%
7250     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
7251 }

```



dua *<long>* only acronym style.

```
7252 \newacronymstyle{dua}%  
7253 {%
```

Check for long form in case this is a mixed glossary.

```
7254 \ifdefempty\glscustomtext  
7255 {%  
7256 \ifglshaslong{\glslabel}%  
7257 {%  
7258 \glsifplural  
7259 {%
```

Plural form:

```
7260 \glscapscase  
7261 {%
```

Plural form, don't adjust case:

```
7262 \glstrylongpl{\glslabel}\glsinsert  
7263 }%  
7264 {%
```

Plural form, make first letter upper case:

```
7265 \Glsentrylongpl{\glslabel}\glsinsert  
7266 }%  
7267 {%
```

Plural form, all caps:

```
7268 \mfirstucMakeUppercase  
7269 {\glstrylongpl{\glslabel}\glsinsert}%  
7270 }%  
7271 }%  
7272 {%
```

Singular form

```
7273 \glscapscase  
7274 {%
```

Singular form, don't adjust case:

```
7275 \glstrylong{\glslabel}\glsinsert  
7276 }%  
7277 {%
```

Subsequent singular form, make first letter upper case:

```
7278 \Glsentrylong{\glslabel}\glsinsert  
7279 }%  
7280 {%
```

Subsequent singular form, all caps:

```
7281 \mfirstucMakeUppercase  
7282 {\glstrylong{\glslabel}\glsinsert}%  
7283 }%  
7284 }%  
7285 }%  
7286 {%
```

Not an acronym:

```
7287 \glsgenentryfmt
7288 }%
7289 }%
7290 {\glscustomtext\glsinsert}%
7291 }%
7292 {%
7293 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

7294 \renewcommand*{\acrfullfmt}[3]{%
7295 \glslink{##1}{##2}{\glsentrylong{##2}##3\space
7296 (\acronymfont{\glsentryshort{##2}})}}%
7297 \renewcommand*{\Acrfullfmt}[3]{%
7298 \glslink{##1}{##2}{\Glsentrylong{##2}##3\space
7299 (\acronymfont{\glsentryshort{##2}})}}%
7300 \renewcommand*{\ACRfullfmt}[3]{%
7301 \glslink{##1}{##2}{%
7302 \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
7303 (\acronymfont{\glsentryshort{##2}})}}}%

7304 \renewcommand*{\acrfullplfmt}[3]{%
7305 \glslink{##1}{##2}{\glsentrylongpl{##2}##3\space
7306 (\acronymfont{\glsentryshortpl{##2}})}}%

7307 \renewcommand*{\Acrfullplfmt}[3]{%
7308 \glslink{##1}{##2}{\Glsentrylongpl{##2}##3\space
7309 (\acronymfont{\glsentryshortpl{##2}})}}%
7310 \renewcommand*{\ACRfullplfmt}[3]{%
7311 \glslink{##1}{##2}{%
7312 \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
7313 (\acronymfont{\glsentryshortpl{##2}})}}}%
7314 \renewcommand*{\glsentryfull}[1]{%
7315 \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
7316 }%
7317 \renewcommand*{\Glsentryfull}[1]{%
7318 \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
7319 }%
7320 \renewcommand*{\glsentryfullpl}[1]{%
7321 \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
7322 }%
7323 \renewcommand*{\Glsentryfullpl}[1]{%
7324 \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
7325 }%
7326 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
7327 \renewcommand*{\acronymsort}[2]{##1}%
7328 \renewcommand*{\acronymfont}[1]{##1}%
7329 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7330 }
```

dua-desc *<long>* only acronym style with user-supplied description.

```
7331 \newacronymstyle{dua-desc}%
7332 {%
7333   \GlsUseAcrEntryDispStyle{dua}%
7334 }%
7335 {%
7336   \GlsUseAcrStyleDefs{dua}%
7337   \renewcommand*{\GenericAcronymFields}{}%

7338   \renewcommand*{\acronymentry}[1]{\acronymfont{\glentrylong{##1}}}%
7339   \renewcommand*{\acronymsort}[2]{##2}%
7340 }%
```

footnote *<short>*\footnote{*<long>*} acronym style.

```
7341 \newacronymstyle{footnote}%
7342 {%

  Check for long form in case this is a mixed glossary.

7343   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
7344 }%
7345 {%
7346   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

Need to ensure hyperlinks are switched off on first use:

```
7347   \glshyperfirstfalse
7348   \renewcommand*{\genacrfullformat}[2]{%
7349     \protect\firstacronymfont{\glentryshort{##1}}##2%
7350     \protect\footnote{\glentrylong{##1}}%
7351   }%
7352   \renewcommand*{\Genacrfullformat}[2]{%
7353     \firstacronymfont{\Glentryshort{##1}}##2%
7354     \protect\footnote{\glentrylong{##1}}%
7355   }%
7356   \renewcommand*{\genplacrfullformat}[2]{%
7357     \protect\firstacronymfont{\glentryshortpl{##1}}##2%
7358     \protect\footnote{\glentrylongpl{##1}}%
7359   }%
7360   \renewcommand*{\Genplacrfullformat}[2]{%
7361     \protect\firstacronymfont{\Glentryshortpl{##1}}##2%
7362     \protect\footnote{\glentrylongpl{##1}}%
7363   }%
7364   \renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
7365   \renewcommand*{\acronymsort}[2]{##1}%
7366   \renewcommand*{\acronymfont}[1]{##1}%
7367   \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
```

Don't use footnotes for \acrfull:

```
7368   \renewcommand*{\acrfullfmt}[3]{%
7369     \glslink[##1]{##2}{\acronymfont{\glentryshort{##2}}##3\space
7370       (\glentrylong{##2})}%
```

```

7371 \renewcommand*{\Acrfullfmt}[3]{%
7372   \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
7373   (\glsentrylong{##2})}%
7374 \renewcommand*{\ACRfullfmt}[3]{%
7375   \glslink[##1]{##2}{%
7376     \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
7377     (\glsentrylong{##2})}}}%
7378 \renewcommand*{\acrfullplfmt}[3]{%
7379   \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
7380   (\glsentrylongpl{##2})}%
7381 \renewcommand*{\Acrfullplfmt}[3]{%
7382   \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
7383   (\glsentrylongpl{##2})}%
7384 \renewcommand*{\ACRfullplfmt}[3]{%
7385   \glslink[##1]{##2}{%
7386     \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
7387     (\glsentrylongpl{##2})}}}%

```

Similarly for \glsentryfull etc:

```

7388 \renewcommand*{\glsentryfull}[1]{%
7389   \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
7390 \renewcommand*{\Glsentryfull}[1]{%
7391   \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
7392 \renewcommand*{\glsentryfullpl}[1]{%
7393   \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
7394 \renewcommand*{\Glsentryfullpl}[1]{%
7395   \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
7396 }

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

7397 \newacronymstyle{footnote-sc}%
7398 {%
7399   \GlsUseAcrEntryDisplayStyle{footnote}%
7400 }%
7401 {%
7402   \GlsUseAcrStyleDefs{footnote}%
7403   \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
7404   \renewcommand{\acronymfont}[1]{\textsc{##1}}%
7405   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7406 }%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```

7407 \newacronymstyle{footnote-sm}%
7408 {%
7409   \GlsUseAcrEntryDisplayStyle{footnote}%
7410 }%
7411 {%
7412   \GlsUseAcrStyleDefs{footnote}%
7413   \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
7414   \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%

```

```

7415 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
7416 }%

```

footnote-desc *<short>*\footnote{*<long>*} acronym style that has an accompanying description (which the user needs to supply).

```

7417 \newacronymstyle{footnote-desc}%
7418 {%
7419 \GlsUseAcrEntryDisplayStyle{footnote}%
7420 }%
7421 {%
7422 \GlsUseAcrStyleDefs{footnote}%
7423 \renewcommand*{\GenericAcronymFields}{}%
7424 \renewcommand*{\acronymsort}[2]{##2}%
7425 \renewcommand*{\acronymentry}[1]{%
7426 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
7427 }

```

ootnote-sc-desc \textsc{*<short>*}\footnote{*<long>*} acronym style that has an accompanying description (which the user needs to supply).

```

7428 \newacronymstyle{footnote-sc-desc}%
7429 {%
7430 \GlsUseAcrEntryDisplayStyle{footnote-sc}%
7431 }%
7432 {%
7433 \GlsUseAcrStyleDefs{footnote-sc}%
7434 \renewcommand*{\GenericAcronymFields}{}%
7435 \renewcommand*{\acronymsort}[2]{##2}%
7436 \renewcommand*{\acronymentry}[1]{%
7437 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
7438 }

```

ootnote-sm-desc \textsmaller{*<short>*}\footnote{*<long>*} acronym style that has an accompanying description (which the user needs to supply).

```

7439 \newacronymstyle{footnote-sm-desc}%
7440 {%
7441 \GlsUseAcrEntryDisplayStyle{footnote-sm}%
7442 }%
7443 {%
7444 \GlsUseAcrStyleDefs{footnote-sm}%
7445 \renewcommand*{\GenericAcronymFields}{}%
7446 \renewcommand*{\acronymsort}[2]{##2}%
7447 \renewcommand*{\acronymentry}[1]{%
7448 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
7449 }

```

## AcronymSynonyms

```

7450 \newcommand*{\DefineAcronymSynonyms}{%

```

Short form

`\acs`  
 7451 `\let\acs\acrshort`  
 First letter uppercase short form

`\Acs`  
 7452 `\let\Acs\Acrshort`  
 Plural short form

`\acsp`  
 7453 `\let\acsp\acrshortpl`  
 First letter uppercase plural short form

`\Acsp`  
 7454 `\let\Acsp\Acrshortpl`  
 Long form

`\acl`  
 7455 `\let\acl\acllong`  
 Plural long form

`\aclp`  
 7456 `\let\aclp\acllongpl`  
 First letter upper case long form

`\Acl`  
 7457 `\let\Acl\Acllong`  
 First letter upper case plural long form

`\Aclp`  
 7458 `\let\Aclp\Acllongpl`  
 Full form

`\acf`  
 7459 `\let\acf\acrfull`  
 Plural full form

`\acfp`  
 7460 `\let\acfp\acrfullpl`  
 First letter upper case full form

`\Acf`  
 7461 `\let\Acf\Acrfull`

First letter upper case plural full form

`\Acfp`

```
7462 \let\Acfp\Acrfullpl
```

Standard form

`\ac`

```
7463 \let\ac\gls
```

First upper case standard form

`\Ac`

```
7464 \let\Ac\Gls
```

Standard plural form

`\acp`

```
7465 \let\acp\glspl
```

Standard first letter upper case plural form

`\Acp`

```
7466 \let\Acp\Glspl
```

```
7467 }
```

Define synonyms if required

```
7468 \ifglsacrshortcuts
```

```
7469 \DefineAcronymSynonyms
```

```
7470 \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.

```
7471 \newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
```

```
7472 \defglsentryfmt[#1]{\glsentryfmt}%
```

```
7473 }
```

`\glsNewAcronymDef` Sets up the acronym definition for the default style. The information is provided by the tokens `\glslabeltok`, `\glsshorttok`, `\gslongtok` and `\glskeylisttok`.

```
7474 \newcommand*{\DefaultNewAcronymDef}{%
```

```
7475 \edef\@do@newglossaryentry{%
```

```
7476 \noexpand\newglossaryentry{\the\glslabeltok}%
```

```
7477 {%
```

```
7478 type=\acronymtype,%
```

```
7479 name={\the\glsshorttok},%
```

```
7480 sort={\the\glsshorttok},%
```

```
7481 text={\the\glsshorttok},%
```

```
7482 first={\acrfullformat{\the\gslongtok}{\the\glsshorttok}},%
```

```
7483 plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
```

```

7484 firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
7485                  {\noexpand\expandonce\noexpand\@glo@shortpl}},%
7486 short={\the\glsshorttok},%
7487 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7488 long={\the\glslongtok},%
7489 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7490 description={\the\glslongtok},%
7491 descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```

Remaining options specified by the user:

```

7492 \the\glskeylisttok
7493 }%
7494 }%
7495 \let\@org@gls@assign@firstpl\gls@assign@firstpl
7496 \let\@org@gls@assign@plural\gls@assign@plural
7497 \let\@org@gls@assign@descplural\gls@assign@descplural
7498 \def\gls@assign@firstpl##1##2{%
7499   \@gls@expand@field{##1}{firstpl}{##2}%
7500 }%
7501 \def\gls@assign@plural##1##2{%
7502   \@gls@expand@field{##1}{plural}{##2}%
7503 }%
7504 \def\gls@assign@descplural##1##2{%
7505   \@gls@expand@field{##1}{descplural}{##2}%
7506 }%
7507 \@do@newglossaryentry
7508 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7509 \let\gls@assign@plural\@org@gls@assign@plural
7510 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural

```

If expansion is switched off, the default plurals won't be properly set.

```

7511 \@@gls@noexpand@assign@default@plurals
7512 }

```

default@plurals

```

7513 \newcommand*{\@@gls@noexpand@assign@default@plurals}{%
7514   \expandafter\ifx\csname glo@\the\glslabeltok @plural\endcsname\@@gls@@default@acr@plural
7515     \csxdef{glo@\the\glslabeltok @plural}{\expandonce\@glo@shortpl}%
7516   \fi
7517   \expandafter\ifx\csname glo@\the\glslabeltok @firstpl\endcsname\@@gls@@default@acr@firstplura
7518     \csxdef{glo@\the\glslabeltok @firstpl}{%
7519       \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
7520   \fi
7521   \expandafter\ifx\csname glo@\the\glslabeltok @descplural\endcsname\@@gls@@default@acr@descplu
7522     \csxdef{glo@\the\glslabeltok @descplural}{\expandonce\@glo@longpl}%
7523   \fi
7524 }

```

ault@acr@plural

```

7525 \newcommand*{\@@gls@@default@acr@plural}{\expandonce\@glo@shortpl}

```



acr@firstplural

```
7526 \newcommand*{\@@gls@@default@acr@firstplural}{%  
7527   \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%  
7528 }
```

@acr@descplural

```
7529 \newcommand*{\@@gls@@default@acr@descplural}{\expandonce\@glo@longpl}
```

ultAcronymStyle Set up the default acronym style:

```
7530 \newcommand*{\SetDefaultAcronymStyle}{%
```

Set the display style:

```
7531   \@for\@gls@type:=\@glsacronymlists\do{%  
7532     \SetDefaultAcronymDisplayStyle{\@gls@type}%  
7533   }%
```

Set up the definition of \newacronym:

```
7534   \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).

```
7535     \ifx\@glsacronymlists\@empty  
7536       \def\@glo@type{\acronymtype}%  
7537       \setkeys{glossentry}{##1}%  
7538       \DeclareAcronymList{\@glo@type}%  
7539       \SetDefaultAcronymDisplayStyle{\@glo@type}%  
7540     \fi  
7541     \glskeylisttok{##1}%  
7542     \glslabeltok{##2}%  
7543     \glsshorttok{##3}%  
7544     \glslongtok{##4}%  
7545     \newacronymhook  
7546     \DefaultNewAcronymDef  
7547   }%  
7548   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%  
7549 }
```

\acrfootnote Used by the footnote acronym styles.

```
7550 \newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
```

acrlinkfootnote

```
7551 \newcommand*{\acrlinkfootnote}[3]{%  
7552   \footnote{\glslink[#1]{#2}{#3}}%  
7553 }
```

rnolinkfootnote

```
7554 \newcommand*{\acrlinkfootnote}[3]{%  
7555   \footnote{#3}%  
7556 }
```

`nymDisplayStyle` Sets the acronym display style for given glossary for the description and footnote combination.

```

7557 \newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
7558   \defglentryfmt[#1]{%

7559     \ifdefempty\glscustomtext
7560     {%
7561       \ifglused{\glslabel}%
7562       {%
7563         \acronymfont{\glsgenentryfmt}%
7564       }%
7565     {%
7566       \firstacronymfont{\glsgenentryfmt}%
7567       \ifglshassymbol{\glslabel}%
7568       {%
7569         \expandafter\protect\expandafter\acrfootnote\expandafter
7570         {\@gls@link@opts}{\@gls@link@label}%
7571       {%
7572         \glsifplural
7573         {\glsentrysymbolplural{\glslabel}}%
7574         {\glsentrysymbol{\glslabel}}%
7575       }%
7576     }%
7577   }%
7578 }%
7579 {\glscustomtext\glsinsert}%
7580 }%
7581 }
```

`teNewAcronymDef`

```

7582 \newcommand*{\DescriptionFootnoteNewAcronymDef}{%
7583   \edef\@do@newglossaryentry{%
7584     \noexpand\newglossaryentry{\the\glslabeltok}%
7585     {%
7586       type=\acronymtype,%
7587       name={\noexpand\acronymfont{\the\glsshorttok}},%
7588       sort={\the\glsshorttok},%
7589       first={\the\glsshorttok},%
7590       firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7591       text={\the\glsshorttok},%
7592       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7593       short={\the\glsshorttok},%
7594       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7595       long={\the\glslongtok},%
7596       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7597       symbol={\the\glslongtok},%
7598       symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7599       \the\glskeylisttok
7600     }%

```

```

7601 }%
7602 \let\@org@gl@s@assign@firstpl\gl@s@assign@firstpl
7603 \let\@org@gl@s@assign@plural\gl@s@assign@plural
7604 \let\@org@gl@s@assign@symbolplural\gl@s@assign@symbolplural
7605 \def\gl@s@assign@firstpl##1##2{%
7606   \@@gl@s@expand@field{##1}{firstpl}{##2}%
7607 }%
7608 \def\gl@s@assign@plural##1##2{%
7609   \@@gl@s@expand@field{##1}{plural}{##2}%
7610 }%
7611 \def\gl@s@assign@symbolplural##1##2{%
7612   \@@gl@s@expand@field{##1}{symbolplural}{##2}%
7613 }%
7614 \do@newglossaryentry
7615 \let\gl@s@assign@plural\@org@gl@s@assign@plural
7616 \let\gl@s@assign@firstpl\@org@gl@s@assign@firstpl
7617 \let\gl@s@assign@symbolplural\@org@gl@s@assign@symbolplural
7618 }

```

**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`.

```

7619 \newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
7620   \ifgl@sacrsmallcaps
7621     \@gl@s@deprecated@acr@warning{footnote,smallcaps,description}%
7622     {footnote-sc-desc}{short-sc-footnote-desc}%
7623   \else
7624     \ifgl@sacrsmaller
7625       \@gl@s@deprecated@acr@warning{footnote,smaller,description}%
7626       {footnote-sm-desc}{short-sm-footnote-desc}%
7627     \else
7628       \@gl@s@deprecated@acr@warning{footnote,description}%
7629       {footnote-desc}{short-footnote-desc}%
7630     \fi
7631   \fi
7632   \renewcommand{\newacronym}[4][\]{%
7633     \ifx\@gl@sacronymlists\@empty
7634       \def\@glo@type{\acronymtype}%
7635       \setkeys{glossentry}{##1}%
7636       \DeclareAcronymList{\@glo@type}%
7637       \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
7638     \fi
7639     \gl@skeylisttok{##1}%
7640     \gl@slabeltok{##2}%
7641     \gl@sshorttok{##3}%
7642     \gl@slongtok{##4}%
7643     \newacronymhook
7644     \DescriptionFootnoteNewAcronymDef

```

7645 }%

If footnote package option is specified, set the first use to append the long form (stored in symbol) as a footnote.

```
7646 \@for\@gls@type:=\@glsacronymlists\do{%
7647   \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
7648 }%
```

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.

```
7649 \ifglsacrsmallcaps
7650   \renewcommand*\acronymfont}[1]{\textsc{##1}}%
7651   \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
7652 \else
7653   \ifglsacrsmaller
7654     \renewcommand*\acronymfont}[1]{\textsmaller{##1}}%
7655   \fi
7656 \fi
```

Check for package option clash

```
7657 \ifglsacrdua
7658   \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
7659     can’t both be set}{}%
7660 \fi
7661 }%
```

**nymDisplayStyle** Sets the acronym display style for given glossary with description and dua combination.

```
7662 \newcommand*\SetDescriptionDUAAcronymDisplayStyle}[1]{%
7663   \defglsentryfmt[#1]{\glsngenentryfmt}%
7664 }
```

**UANewAcronymDef**

```
7665 \newcommand*\DescriptionDUANewAcronymDef{%
7666   \edef\@do@newglossaryentry{%
7667     \noexpand\newglossaryentry{\the\glslabeltok}%
7668     {%
7669       type=\acronymtype,%
7670       name={\the\glslongtok},%
7671       sort={\the\glslongtok},%
7672       text={\the\glslongtok},%
7673       first={\the\glslongtok},%
7674       plural={\noexpand\expandonce\noexpand\@glo@longpl},%
7675       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7676       short={\the\glsshorttok},%
7677       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7678       long={\the\glslongtok},%
7679       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7680       symbol={\the\glsshorttok},%
7681       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
```

```

7682     \the\glskeylisttok
7683   }%
7684 }%
7685 \let\@org@gls@assign@firstpl\gls@assign@firstpl
7686 \let\@org@gls@assign@plural\gls@assign@plural
7687 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
7688 \def\gls@assign@firstpl##1##2{%
7689   \@gls@expand@field{##1}{firstpl}{##2}%
7690 }%
7691 \def\gls@assign@plural##1##2{%
7692   \@gls@expand@field{##1}{plural}{##2}%
7693 }%
7694 \def\gls@assign@symbolplural##1##2{%
7695   \@gls@expand@field{##1}{symbolplural}{##2}%
7696 }%
7697 \do@newglossaryentry
7698 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7699 \let\gls@assign@plural\@org@gls@assign@plural
7700 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
7701 }

```

**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`.

```

7702 \newcommand*{\SetDescriptionDUAAcronymStyle}{%
7703   \ifglsacrsmallcaps
7704     \PackageError{glossaries}{Option clash: 'smallcaps' and 'dua'
7705       can't both be set}{}%
7706   \else
7707     \ifglsacrsmaller
7708       \PackageError{glossaries}{Option clash: 'smaller' and 'dua'
7709         can't both be set}{}%
7710     \fi
7711   \fi
7712   \@gls@deprecated@acr@warning{dua,description}{dua-desc}{long-noshort-desc}%
7713   \renewcommand{\newacronym}[4][{}]{%
7714     \ifx\@glsacronymlists\@empty
7715       \def\@glo@type{\acronymtype}%
7716       \setkeys{glossentry}{##1}%
7717       \DeclareAcronymList{\@glo@type}%
7718       \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
7719     \fi
7720     \glskeylisttok{##1}%
7721     \glslabeltok{##2}%
7722     \glsshorttok{##3}%
7723     \glslongtok{##4}%
7724     \newacronymhook
7725     \DescriptionDUANewAcronymDef
7726   }%

```

Set display.

```
7727 \for\@gls@type:=\@glsacronymlists\do{%
7728   \SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
7729 }%
7730 }%
```

`\acronymDisplayStyle` Sets the acronym display style for given glossary using the description setting (but not footnote or dua).

```
7731 \newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
7732   \def\glsentryfmt[#1]{%

7733     \ifdefempty\glscustomtext
7734     {%
7735       \ifglsused{\glslabel}%
7736       {%
```

Move the inserted text outside of `\acronymfont`

```
7737       \let\gls@org@insert\glsinsert
7738       \let\glsinsert\@empty
7739       \acronymfont{\glsentryfmt}\gls@org@insert
7740     }%
7741     {%
7742       \glsentryfmt
7743       \ifgls hassymbol{\glslabel}%
7744       {%
7745         \glsifplural
7746         {%
7747           \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
7748         }%
7749         {%
7750           \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
7751         }%
7752         \space(\protect\firstacronymfont
7753         {\glscapscase
7754          {\@glo@symbol}
7755          {\@glo@symbol}
7756          {\mfirstucMakeUppercase{\@glo@symbol}}})%
7757       }%
7758     }%
7759   }%
7760 }%
7761 {\glscustomtext\glsinsert}%
7762 }%
7763 }
```

`\onNewAcronymDef`

```
7764 \newcommand*{\DescriptionNewAcronymDef}{%
7765   \edef\@do@newglossaryentry{%
7766     \noexpand\newglossaryentry{\the\glslabeltok}%
```

```

7767 {%
7768     type=\acronymtype,%
7769     name={\noexpand
7770         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
7771     sort={\the\glsshorttok},%
7772     first={\the\glslongtok},%
7773     firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7774     text={\the\glsshorttok},%
7775     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7776     short={\the\glsshorttok},%
7777     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7778     long={\the\glslongtok},%
7779     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7780     symbol={\noexpand\@glo@text},%
7781     symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7782     \the\glskeylisttok}%
7783 }%
7784 \let\@org@gls@assign@firstpl\gls@assign@firstpl
7785 \let\@org@gls@assign@plural\gls@assign@plural
7786 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
7787 \def\gls@assign@firstpl##1##2{%
7788     \@@gls@expand@field{##1}{firstpl}{##2}%
7789 }%
7790 \def\gls@assign@plural##1##2{%
7791     \@@gls@expand@field{##1}{plural}{##2}%
7792 }%
7793 \def\gls@assign@symbolplural##1##2{%
7794     \@@gls@expand@field{##1}{symbolplural}{##2}%
7795 }%
7796 \do@newglossaryentry
7797 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7798 \let\gls@assign@plural\@org@gls@assign@plural
7799 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
7800 }

```

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.

```

7801 \newcommand*{\SetDescriptionAcronymStyle}{%
7802     \ifglsacrsmallcaps
7803         \@gls@deprecated@acr@warning{smallcaps,description}%
7804         {long-sc-short-desc}{long-short-sc-desc}%
7805     \else
7806         \ifglsacrsmaller
7807             \@gls@deprecated@acr@warning{smaller,description}%
7808             {long-sm-short-desc}{long-short-sm-desc}%
7809         \else
7810             \@gls@deprecated@acr@warning{description}%
7811             {long-short-desc}{long-short-desc}%

```

```

7812 \fi
7813 \fi
7814 \renewcommand{\newacronym}[4][\%
7815 \ifx\@glsacronymlists\@empty
7816 \def\@glo@type{\acronymtype}%
7817 \setkeys{glossentry}{##1}%
7818 \DeclareAcronymList{\@glo@type}%
7819 \SetDescriptionAcronymDisplayStyle{\@glo@type}%
7820 \fi
7821 \glskeylisttok{##1}%
7822 \glslabeltok{##2}%
7823 \glsshorttok{##3}%
7824 \glslongtok{##4}%
7825 \newacronymhook
7826 \DescriptionNewAcronymDef
7827 }%

```

Set display.

```

7828 \@for\@gls@type:=\@glsacronymlists\do{%
7829 \SetDescriptionAcronymDisplayStyle{\@gls@type}%
7830 }%

```

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.

```

7831 \ifglsacrsmallcaps
7832 \renewcommand{\acronymfont}[1]{\textsc{##1}}
7833 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7834 \else
7835 \ifglsacrsmaller
7836 \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
7837 \fi
7838 \fi
7839 }%

```

`\acronymDisplayStyle` Sets the acronym display style for given glossary with footnote setting (but not description or dua).

```

7840 \newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
7841 \defglsentryfmt[##1]{%
7842 \ifdefempty\glscustomtext
7843 {%

```

Move the inserted text outside of `\acronymfont`

```

7844 \let\gls@org@insert\glsinsert
7845 \let\glsinsert\@empty
7846 \ifglsused{\glslabel}%
7847 {%
7848 \acronymfont{\glsentryfmt}\gls@org@insert
7849 }%
7850 {%

```



```

7851      \firstacronymfont{\glsgenentryfmt}\gls@org@insert
7852      \ifglshaslong{\glslabel}%
7853      {%
7854          \expandafter\protect\expandafter\acrfootnote\expandafter
7855          {\@gls@link@opts}{\@gls@link@label}%
7856          {%
7857              \glsifplural
7858                  {\glsentrylongpl{\glslabel}}%
7859                  {\glsentrylong{\glslabel}}%
7860              }%
7861          }%
7862      {%}
7863      }%
7864      }%
7865      {\glscustomtext\glsinsert}%
7866      }%
7867      }

```

teNewAcronymDef

```

7868 \newcommand*{\FootnoteNewAcronymDef}{%
7869   \edef\@do@newglossaryentry{%
7870     \noexpand\newglossaryentry{\the\glslabeltok}%
7871     {%
7872       type=\acronymtype,%
7873       name={\noexpand\acronymfont{\the\glsshorttok}},%
7874       sort={\the\glsshorttok},%
7875       text={\the\glsshorttok},%
7876       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7877       first={\the\glsshorttok},%
7878       firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7879       short={\the\glsshorttok},%
7880       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
7881       long={\the\glslongtok},%
7882       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
7883       description={\the\glslongtok},%
7884       descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
7885       \the\glskeylisttok
7886     }%
7887   }%
7888   \let\@org@gls@assign@plural\gls@assign@plural
7889   \let\@org@gls@assign@firstpl\gls@assign@firstpl
7890   \let\@org@gls@assign@descplural\gls@assign@descplural
7891   \def\gls@assign@firstpl##1##2{%
7892     \@gls@expand@field{##1}{firstpl}{##2}%
7893   }%
7894   \def\gls@assign@plural##1##2{%
7895     \@gls@expand@field{##1}{plural}{##2}%
7896   }%
7897   \def\gls@assign@descplural##1##2{%

```

```

7898 \@@gls@expand@field{##1}{descplural}{##2}%
7899 }%
7900 \do@newglossaryentry
7901 \let\gls@assign@plural\@org@gls@assign@plural
7902 \let\gls@assign@firstpl\@org@gls@assign@firstpl
7903 \let\gls@assign@descplural\@org@gls@assign@descplural
7904 }

```

`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`.

```

7905 \newcommand*{\SetFootnoteAcronymStyle}{%
7906 \ifglsacrsmallcaps
7907 \@gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
7908 \else
7909 \ifglsacrsmaller
7910 \@gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
7911 \else
7912 \@gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
7913 \fi
7914 \fi
7915 \renewcommand{\newacronym}[4][\]{%
7916 \ifx\@glsacronymlists\@empty
7917 \def\@glo@type{\acronymtype}%
7918 \setkeys{glossentry}{##1}%
7919 \DeclareAcronymList{\@glo@type}%
7920 \SetFootnoteAcronymDisplayStyle{\@glo@type}%
7921 \fi
7922 \glskeylisttok{##1}%
7923 \glslabeltok{##2}%
7924 \glsshorttok{##3}%
7925 \glslongtok{##4}%
7926 \newacronymhook
7927 \FootnoteNewAcronymDef
7928 }%

```

Set display

```

7929 \@for\@gls@type:=\@glsacronymlists\do{%
7930 \SetFootnoteAcronymDisplayStyle{\@gls@type}%
7931 }%

```

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.

```

7932 \ifglsacrsmallcaps
7933 \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
7934 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
7935 \else
7936 \ifglsacrsmaller
7937 \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
7938 \fi

```

7939 \fi

Check for option clash

7940 \ifglsacrdua

7941 \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’

7942 can’t both be set}{}%

7943 \fi

7944 }%

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.

7945 \DeclareRobustCommand\*{\glsdoparenifnotempty}[2]{%

7946 \protected@edef\gls@tmp{#1}%

7947 \ifdefempty\gls@tmp

7948 {}%

7949 {%

7950 \ifx\gls@tmp\@gls@default@value

7951 \else

7952 \space (#2{#1})%

7953 \fi

7954 }%

7955 }

nymDisplayStyle Sets the acronym display style for given glossary where neither footnote nor description is required, but smallcaps or smaller specified.

7956 \newcommand\*{\SetSmallAcronymDisplayStyle}[1]{%

7957 \defglsentryfmt[#1]{%

7958 \ifdefempty\glscustomtext

7959 {%

Move the inserted text outside of \acronymfont

7960 \let\gls@org@insert\glsinsert

7961 \let\glsinsert\@empty

7962 \ifglsused{\glslabel}%

7963 {%

7964 \acronymfont{\glsgenentryfmt}\gls@org@insert

7965 }%

7966 {%

7967 \glsgenentryfmt

7968 \ifgls hassymbol{\glslabel}%

7969 {%

7970 \glsifplural

7971 {%

7972 \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%

7973 }%

7974 {%

7975 \def\@glo@symbol{\glsentrysymbol{\glslabel}}%

7976 }%

7977 \space

```

7978      (\glscapscase
7979      {\firstacronymfont{\@glo@symbol}}}%
7980      {\firstacronymfont{\@glo@symbol}}}%
7981      {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
7982      }%
7983      {}%
7984      }%
7985      }%
7986      {\glscustomtext\glsinsert}%
7987      }%
7988 }

```

# 11NewAcronymDef

```

7989 \newcommand*{\SmallNewAcronymDef}{%
7990   \edef\@do@newglossaryentry{%
7991     \noexpand\newglossaryentry{\the\glslabeltok}%
7992     {%
7993       type=\acronymtype,%
7994       name={\noexpand\acronymfont{\the\glsshorttok}},%
7995       sort={\the\glsshorttok},%
7996       text={\the\glsshorttok},%
7997       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
7998       first={\the\glslongtok},%
7999       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
8000       short={\the\glsshorttok},%
8001       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
8002       long={\the\glslongtok},%
8003       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
8004       description={\noexpand\@glo@first},%
8005       descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
8006       symbol={\the\glsshorttok},%
8007       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
8008       \the\glskeylisttok
8009     }%
8010   }%
8011   \let\@org@gls@assign@firstpl\gls@assign@firstpl
8012   \let\@org@gls@assign@plural\gls@assign@plural
8013   \let\@org@gls@assign@descplural\gls@assign@descplural
8014   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
8015   \def\gls@assign@firstpl##1##2{%
8016     \@@gls@expand@field{##1}{firstpl}{##2}%
8017   }%
8018   \def\gls@assign@plural##1##2{%
8019     \@@gls@expand@field{##1}{plural}{##2}%

```

```

8020 }%
8021 \def\gls@assign@descplural##1##2{%
8022   \@gls@expand@field{##1}{descplural}{##2}%
8023 }%
8024 \def\gls@assign@symbolplural##1##2{%
8025   \@gls@expand@field{##1}{symbolplural}{##2}%
8026 }%
8027 \do@newglossaryentry
8028 \let\gls@assign@firstpl\@org@gls@assign@firstpl
8029 \let\gls@assign@plural\@org@gls@assign@plural
8030 \let\gls@assign@descplural\@org@gls@assign@descplural
8031 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
8032 }

```

`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`.

```

8033 \newcommand*{\SetSmallAcronymStyle}{%
8034   \ifglsacrsmallcaps
8035     \@gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
8036   \else
8037     \@gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
8038   \fi
8039   \renewcommand{\newacronym}[4][]{%
8040     \ifx\@glsacronymlists\@empty
8041       \def\@glo@type{\acronymtype}%
8042       \setkeys{glossentry}{##1}%
8043       \DeclareAcronymList{\@glo@type}%
8044       \SetSmallAcronymDisplayStyle{\@glo@type}%
8045     \fi
8046     \glskeylisttok{##1}%
8047     \glslabeltok{##2}%
8048     \glsshorttok{##3}%
8049     \glslongtok{##4}%
8050     \newacronymhook
8051     \SmallNewAcronymDef
8052   }%

```

Change the display since first only contains long form.

```

8053   \@for\@gls@type:=\@glsacronymlists\do{%
8054     \SetSmallAcronymDisplayStyle{\@gls@type}%
8055   }%

```

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.

```

8056   \ifglsacrsmallcaps
8057     \renewcommand*{\acronymfont}[1]{\textsc{##1}}
8058     \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
8059   \else
8060     \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}

```

```

8061 \fi
      check for option clash
8062 \ifglsacrdua
8063   \ifglsacrsmallcaps
8064     \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
8065       can’t both be set}{}%
8066   \else
8067     \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
8068       can’t both be set}{}%
8069   \fi
8070 \fi
8071 }%

```

**DUADisplayStyle** Sets the acronym display style for given glossary with dua setting.

```

8072 \newcommand*{\SetDUADisplayStyle}[1]{%
8073   \defglsentryfmt[#1]{\glsentryfmt}%
8074 }

```

**UANewAcronymDef**

```

8075 \newcommand*{\DUANewAcronymDef}{%
8076   \edef\@do@newglossaryentry{%
8077     \noexpand\newglossaryentry{\the\glslabeltok}%
8078     {%
8079       type=\acronymtype,%
8080       name={\the\glsshorttok},%
8081       text={\the\glslongtok},%
8082       first={\the\glslongtok},%
8083       plural={\noexpand\expandonce\noexpand\@glo@longpl},%
8084       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
8085       short={\the\glsshorttok},%
8086       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
8087       long={\the\glslongtok},%
8088       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
8089       description={\the\glslongtok},%
8090       descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
8091       symbol={\the\glsshorttok},%
8092       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
8093       \the\glskeylisttok
8094     }%
8095   }%
8096   \let\@org@gls@assign@firstpl\gls@assign@firstpl
8097   \let\@org@gls@assign@plural\gls@assign@plural
8098   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
8099   \let\@org@gls@assign@descplural\gls@assign@descplural
8100   \def\gls@assign@firstpl##1##2{%
8101     \@@gls@expand@field{##1}{firstpl}{##2}%
8102   }%
8103   \def\gls@assign@plural##1##2{%
8104     \@@gls@expand@field{##1}{plural}{##2}%

```

```

8105 }%
8106 \def\gls@assign@symbolplural##1##2{%
8107   \@gls@expand@field{##1}{symbolplural}{##2}%
8108 }%
8109 \def\gls@assign@descplural##1##2{%
8110   \@gls@expand@field{##1}{descplural}{##2}%
8111 }%
8112 \do@newglossaryentry
8113 \let\gls@assign@firstpl\@org@gls@assign@firstpl
8114 \let\gls@assign@plural\@org@gls@assign@plural
8115 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
8116 \let\gls@assign@descplural\@org@gls@assign@descplural
8117 }

```

`\SetDUASStyle` Always expand acronyms. This style was deprecated with the introduction of `\newacronymstyle`.

```

8118 \newcommand*{\SetDUASStyle}{%
8119   \@gls@deprecated@acr@warning{dua}{dua}{long-noshort}%
8120   \renewcommand{\newacronym}[4][ ]{%
8121     \ifx\@glsacronymlists\@empty
8122       \def\@glo@type{\acronymtype}%
8123       \setkeys{glossentry}{##1}%
8124       \DeclareAcronymList{\@glo@type}%
8125       \SetDUADisplayStyle{\@glo@type}%
8126     \fi
8127     \glskeylisttok{##1}%
8128     \glslabeltok{##2}%
8129     \glsshorttok{##3}%
8130     \glslongtok{##4}%
8131     \newacronymhook
8132     \DUANewAcronymDef
8133   }%

```

Set the display

```

8134   \@for\@gls@type:=\@glsacronymlists\do{%
8135     \SetDUADisplayStyle{\@gls@type}%
8136   }%
8137 }

```

`ted@acr@warning`

```

8138 \newcommand*{\@gls@deprecated@acr@warning}[3]{%
8139   \GlossariesWarning{Acronym style option #1 deprecated.
8140   Use \string\setacronymstyle{#2} with just glossaries.sty or
8141   use \string\setabbreviationstyle[acronym]{#3} with
8142   glossaries-extra.sty}%
8143 }

```

`SetAcronymStyle`

```

8144 \newcommand*{\SetAcronymStyle}{%

```

```

8145 \SetDefaultAcronymStyle
8146 \ifglssacrdescription
8147   \ifglssacrfootnote
8148     \SetDescriptionFootnoteAcronymStyle
8149   \else
8150     \ifglssacrdua
8151       \SetDescriptionDUAAcronymStyle
8152     \else
8153       \SetDescriptionAcronymStyle
8154     \fi
8155   \fi
8156 \else
8157   \ifglssacrfootnote
8158     \SetFootnoteAcronymStyle
8159   \else
8160     \ifthenelse{\boolean{glssacrsmalldcaps}}\OR
8161       \boolean{glssacrsmaller}}{%
8162     {%
8163       \SetSmallAcronymStyle
8164     }%
8165     {%
8166       \ifglssacrdua
8167         \SetDUASStyle
8168       \fi
8169     }%
8170   \fi
8171 \fi
8172 }

```

Set the acronym style according to the package options

```
8173 \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.

```

8174 \newcommand*{\SetCustomDisplayStyle}[1]{%
8175   \defglssentryfmt[#1]{\glsgenentryfmt}%
8176 }

```

`\setacronymfields`

```

8177 \newcommand*{\CustomAcronymFields}{%
8178   name={\the\glsshorttok},%
8179   description={\the\glslongtok},%
8180   first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
8181   firstplural={\acrfullformat
8182     {\noexpand\glsgentrylongpl{\the\glslabeltok}}}%
8183     {\noexpand\glsgentryshortpl{\the\glslabeltok}}},%

```



```

8184 text={\the\glsshorttok},%
8185 plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
8186 }

```

omNewAcronymDef

```

8187 \newcommand*{\CustomNewAcronymDef}{%
8188   \protected@edef\@do@newglossaryentry{%
8189     \noexpand\newglossaryentry{\the\glslabeltok}%
8190     {%
8191       type=\acronymtype,%
8192       short={\the\glsshorttok},%
8193       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
8194       long={\the\glslongtok},%
8195       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
8196       user1={\the\glsshorttok},%
8197       user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
8198       user3={\the\glslongtok},%
8199       user4={\the\glslongtok\noexpand\acrpluralsuffix},%
8200       \CustomAcronymFields,%
8201       \the\glskeylisttok
8202     }%
8203   }%
8204   \@do@newglossaryentry
8205 }

```

\SetCustomStyle

```

8206 \newcommand*{\SetCustomStyle}{%
8207   \renewcommand{\newacronym}[4][]{%
8208     \ifx\@glsacronymlists\@empty
8209       \def\@glo@type{\acronymtype}%
8210       \setkeys{glossentry}{##1}%
8211       \DeclareAcronymList{\@glo@type}%
8212       \SetCustomDisplayStyle{\@glo@type}%
8213     \fi
8214     \glskeylisttok{##1}%
8215     \glslabeltok{##2}%
8216     \glsshorttok{##3}%
8217     \glslongtok{##4}%
8218     \newacronymhook
8219     \CustomNewAcronymDef
8220   }%

   Set the display
8221   \@for\@gls@type:=\@glsacronymlists\do{%
8222     \SetCustomDisplayStyle{\@gls@type}%
8223   }%
8224 }

```

## 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.

```
8225 \RequirePackage{glossary-hypernav}
```

The styles that use list-like environments. These are not loaded if the nolist option is used:

```
8226 \@gls@loadlist
```

The styles that use the longtable environment. These are not loaded if the nolong package option is used.

```
8227 \@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.

```
8228 \@gls@loadsuper
```

The tree-like styles. These are not loaded if the notree package option is used.

```
8229 \@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.

```
8230 \ifx\@glossary@default@style\relax
```

```
8231 \else
```

```
8232   \setglossarystyle{\@glossary@default@style}
```

```
8233 \fi
```

## 1.20 Debugging Commands

`\showgloparent`

```
\showgloparent{<label>}
```

```
8234 \newcommand*{\showgloparent}[1]{%
```

```
8235   \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
```

```
8236 }
```

`\showglolevel`

```
\showglolevel{<label>}
```

```
8237 \newcommand*{\showglolevel}[1]{%
```

```
8238   \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
```

```
8239 }
```

`\showglotext`

```
\showglotext{<label>}
```

```

8240 \newcommand*{\showglotext}[1]{%
8241   \expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
8242 }

```

\showgloplural    `\showgloplural{\langle label \rangle}`

```

8243 \newcommand*{\showgloplural}[1]{%
8244   \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
8245 }

```

\showglofirst    `\showglofirst{\langle label \rangle}`

```

8246 \newcommand*{\showglofirst}[1]{%
8247   \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
8248 }

```

\showglofirstpl    `\showglofirstpl{\langle label \rangle}`

```

8249 \newcommand*{\showglofirstpl}[1]{%
8250   \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
8251 }

```

\showglotype    `\showglotype{\langle label \rangle}`

```

8252 \newcommand*{\showglotype}[1]{%
8253   \expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
8254 }

```

\showglocounter    `\showglocounter{\langle label \rangle}`

```

8255 \newcommand*{\showglocounter}[1]{%
8256   \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
8257 }

```

\showglouser    `\showglouser{\langle label \rangle}`

```

8258 \newcommand*{\showglouser}[1]{%

```

```
8259 \expandafter\show\csname glo@\glsdetoklabel{#1}@useri\endcsname
8260 }
```

\showglouserii     \showglouserii{<label>}

```
8261 \newcommand*{\showglouserii}[1]{%
8262   \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
8263 }
```

\showglouseriii    \showglouseriii{<label>}

```
8264 \newcommand*{\showglouseriii}[1]{%
8265   \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
8266 }
```

\showglouseriv     \showglouseriv{<label>}

```
8267 \newcommand*{\showglouseriv}[1]{%
8268   \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
8269 }
```

\showglouserv      \showglouserv{<label>}

```
8270 \newcommand*{\showglouserv}[1]{%
8271   \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
8272 }
```

\showglouservi     \showglouservi{<label>}

```
8273 \newcommand*{\showglouservi}[1]{%
8274   \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
8275 }
```

\showgloname        \showgloname{<label>}

```
8276 \newcommand*{\showgloname}[1]{%
8277   \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
8278 }
```

\showglodesc      \showglodesc{<label>}

```
8279 \newcommand*{\showglodesc}[1]{%
8280   \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname
8281 }
```

owglodescplural      \showglodescplural{<label>}

```
8282 \newcommand*{\showglodescplural}[1]{%
8283   \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
8284 }
```

\showglosort      \showglosort{<label>}

```
8285 \newcommand*{\showglosort}[1]{%
8286   \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
8287 }
```

\showglosymbol      \showglosymbol{<label>}

```
8288 \newcommand*{\showglosymbol}[1]{%
8289   \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
8290 }
```

glosymbolplural      \showglosymbolplural{<label>}

```
8291 \newcommand*{\showglosymbolplural}[1]{%
8292   \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname
8293 }
```

\showgloshort      \showgloshort{<label>}

```
8294 \newcommand*{\showgloshort}[1]{%
8295   \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname
8296 }
```

`\showglolong` `\showglolong{<label>}`

```
8297 \newcommand*{\showglolong}[1]{%
8298   \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
8299 }
```

`\showgloindex` `\showgloindex{<label>}`

```
8300 \newcommand*{\showgloindex}[1]{%
8301   \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
8302 }
```

`\showgloflag` `\showgloflag{<label>}`

```
8303 \newcommand*{\showgloflag}[1]{%
8304   \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
8305 }
```

`\showgloloclist` `\showgloloclist{<label>}`

```
8306 \newcommand*{\showgloloclist}[1]{%
8307   \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
8308 }
```

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

```
8309 \newcommand*{\showglofield}[2]{%
8310   \csshow{glo@\glsdetoklabel{#1}@#2}%
8311 }
```

`\showacronymlists` `\showacronymlists`

Show list of glossaries that have been flagged as a list of acronyms.

```
8312 \newcommand*{\showacronymlists}{%
8313   \show\@glsacronymlists
8314 }
```

`\showglossaries` `\showglossaries`

Show list of defined glossaries.

```

8315 \newcommand*{\showglossaries}{%
8316   \show\@glo@types
8317 }

```

\showglossaryin `\showglossaryin{<glossary-label>}`

Show the ‘in’ extension for the given glossary.

```

8318 \newcommand*{\showglossaryin}[1]{%
8319   \expandafter\show\csname @glo@types@#1@in\endcsname
8320 }

```

showglossaryout `\showglossaryout{<glossary-label>}`

Show the ‘out’ extension for the given glossary.

```

8321 \newcommand*{\showglossaryout}[1]{%
8322   \expandafter\show\csname @glo@types@#1@out\endcsname
8323 }

```

owglossarytitle `\showglossarytitle{<glossary-label>}`

Show the title for the given glossary.

```

8324 \newcommand*{\showglossarytitle}[1]{%
8325   \expandafter\show\csname @glo@types@#1@title\endcsname
8326 }

```

glossarycounter `\showglossarycounter{<glossary-label>}`

Show the counter for the given glossary.

```

8327 \newcommand*{\showglossarycounter}[1]{%
8328   \expandafter\show\csname @glo@types@#1@counter\endcsname
8329 }

```

glossaryentries `\showglossaryentries{<glossary-label>}`

Show the list of entry labels for the given glossary.

```

8330 \newcommand*{\showglossaryentries}[1]{%
8331   \expandafter\show\csname glolist@#1\endcsname
8332 }

```

## 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.

```
8333 \csname ifglscpatible-2.07\endcsname
8334   \RequirePackage{glossaries-compatible-207}
8335 \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.

```
8336 \NeedsTeXFormat{LaTeX2e}
```

```
8337 \ProvidesPackage{glossaries-prefix}[2021/11/01 v4.49 (NLCT)]
```

Pass all options to glossaries:

```
8338 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
8339 \ProcessOptions
```

Load glossaries:

```
8340 \RequirePackage{glossaries}
```

Add the new keys:

```
8341 \define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
```

```
8342 \define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
```

```
8343 \define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
```

```
8344 \define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
```

Add them to `\@gls@keymap`:

```
8345 \appto\@gls@keymap{,%
```

```
8346   {prefixfirst}{prefixfirst},%
```

```
8347   {prefixfirstplural}{prefixfirstplural},%
```

```
8348   {prefix}{prefix},%
```

```
8349   {prefixplural}{prefixplural}}%
```

```
8350 }
```

Set the default values:

```
8351 \appto\@newglossaryentryprehook{%
```

```
8352   \def\@glo@entryprefix{}}%
```

```
8353   \def\@glo@entryprefixplural{}}%
```

```
8354   \let\@glo@entryprefixfirst\@gls@default@value
```

```
8355   \let\@glo@entryprefixfirstplural\@gls@default@value
```

```
8356 }
```

Set the assignment code:

```
8357 \appto\@newglossaryentryposthook{%
```

```
8358   \gls@assign@field{}{\@glo@label}{prefix}{\@glo@entryprefix}}%
```

```
8359   \gls@assign@field{}{\@glo@label}{prefixplural}{\@glo@entryprefixplural}}%
```

If `prefixfirst` has not been supplied, make it the same as `prefix`.

```
8360 \expandafter\gls@assign@field\expandafter
```

```
8361   {\csname glo@\@glo@label @prefix\endcsname}{\@glo@label}{prefixfirst}}%
```

```
8362   {\@glo@entryprefixfirst}}%
```

If prefixfirstplural has not been supplied, make it the same as prefixplural.

```

8363 \expandafter\gls@assign@field\expandafter
8364 {\csname glo@\glo@label @prefixplural\endcsname}{\@glo@label}%
8365 {prefixfirstplural}{\@glo@entryprefixfirstplural}%
8366 }

```

Define commands to access these fields:

entryprefixfirst

```

8367 \newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}

```

entryfirstplural

```

8368 \newcommand*{\glsentryprefixfirstplural}[1]{%
8369 \csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}

```

\glsentryprefix

```

8370 \newcommand*{\glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}

```

entryprefixplural

```

8371 \newcommand*{\glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}

```

Now for the initial upper case variants:

entryprefixfirst

```

8372 \newrobustcmd*{\Glsentryprefixfirst}[1]{%
8373 \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
8374 \xmakefirstuc\@glo@text
8375 }

```

entryfirstplural

```

8376 \newrobustcmd*{\Glsentryprefixfirstplural}[1]{%
8377 \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
8378 \xmakefirstuc\@glo@text
8379 }

```

\Glsentryprefix

```

8380 \newrobustcmd*{\Glsentryprefix}[1]{%
8381 \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
8382 \xmakefirstuc\@glo@text
8383 }

```

entryprefixplural

```

8384 \newrobustcmd*{\Glsentryprefixplural}[1]{%
8385 \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
8386 \xmakefirstuc\@glo@text
8387 }

```

Define commands to determine if the prefix keys have been set:

`\ifglshasprefix`

```
8388 \newcommand*{\ifglshasprefix}[3]{%
8389   \ifcempty{glo@glstdetoklabel{#1}@prefix}%
8390   {#3}%
8391   {#2}%
8392 }
```

`hasprefixplural`

```
8393 \newcommand*{\ifglshasprefixplural}[3]{%
8394   \ifcempty{glo@glstdetoklabel{#1}@prefixplural}%
8395   {#3}%
8396   {#2}%
8397 }
```

`shasprefixfirst`

```
8398 \newcommand*{\ifglshasprefixfirst}[3]{%
8399   \ifcempty{glo@glstdetoklabel{#1}@prefixfirst}%
8400   {#3}%
8401   {#2}%
8402 }
```

`efixfirstplural`

```
8403 \newcommand*{\ifglshasprefixfirstplural}[3]{%
8404   \ifcempty{glo@glstdetoklabel{#1}@prefixfirstplural}%
8405   {#3}%
8406   {#2}%
8407 }
```

`fix@record@hook` Need to take into account the possibility that glossaries-extra might be loaded with the record option.

```
8408 \providecommand{\@glsprefix@record@hook}[2]{%
8409   \ifdef\@glsextr@record
8410   {\@glsextr@record{#1}{#2}{glslink}}%
8411   {}%
8412 }
```

`\glsprefixsep` Separator between prefix and term. Does nothing by default.

```
8413 \newcommand{\glsprefixsep}{}%
```

Define commands that insert the prefix before commands like `\gls`:

`\pgls`

```
8414 \newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
```

`\@pgls` Unstarred version.

```
8415 \newcommand*{\@pgls}[2][{}]{%
8416   \new@ifnextchar[%
8417   {\@pgls@{#1}{#2}}%
```

```

8418 {\@pgls@{#1}{#2}[]}%
8419 }

```

\@pgls@ Read in the final optional argument:

```

8420 \def\@pgls@#1#2[#3]{%
8421   \@glsprefix@record@hook{#1}{#2}%
8422   \glsdoifexists{#2}%
8423   {%
8424     \ifglsused{#2}%
8425     {%
8426       \ifglshasprefix{#2}{\glstryprefix{#2}\glsprefixsep}{}%
8427     }%
8428     {%
8429       \ifglshasprefixfirst{#2}{\glstryprefixfirst{#2}\glsprefixsep}{}%
8430     }%
8431     \@gls@{#1}{#2}[#3]%
8432   }%
8433 }

```

Similarly for the plural version:

```

\pglsp1
8434 \newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}

```

\@pglsp1 Unstarred version.

```

8435 \newcommand*{\@pglsp1}[2][]{%
8436   \new@ifnextchar[%
8437   {\@pglsp1@{#1}{#2}}%
8438   {\@pglsp1@{#1}{#2}[]}%
8439 }

```

\@pglsp1@ Read in the final optional argument:

```

8440 \def\@pglsp1@#1#2[#3]{%
8441   \@glsprefix@record@hook{#1}{#2}%
8442   \glsdoifexists{#2}%
8443   {%
8444     \ifglsused{#2}%
8445     {%
8446       \ifglshasprefixplural{#2}{\glstryprefixplural{#2}\glsprefixsep}{}%
8447     }%
8448     {%
8449       \ifglshasprefixfirstplural{#2}%
8450       {\glstryprefixfirstplural{#2}\glsprefixsep}{}%
8451     }%
8452     \@glspl@{#1}{#2}[#3]%
8453   }%
8454 }

```

Now for the first letter upper case versions:

\Pgl

```
8455 \newrobustcmd{\Pgl}{\@gls@hyp@opt\Pgl}
```

\@Pgl Unstarred version.

```
8456 \newcommand*{\@Pgl}[2][]{%
8457   \new@ifnextchar[%
8458     {\@Pgl@{#1}{#2}}%
8459     {\@Pgl@{#1}{#2} []}%
8460 }
```

\@Pgl@ Read in the final optional argument:

```
8461 \def\@Pgl@#1#2[#3]{%
8462   \glsprefix@record@hook{#1}{#2}%
8463   \glsdoifexists{#2}%
8464   {%
8465     \ifglsused{#2}%
8466     {%
8467       \ifglshasprefix{#2}%
8468       {%
8469         \Glsentryprefix{#2}%
8470         \glsprefixsep
8471         \@gls@{#1}{#2}[#3]%
8472       }%
8473       {\@Gls@{#1}{#2}[#3]}%
8474     }%
8475     {%
8476       \ifglshasprefixfirst{#2}%
8477       {%
8478         \Glsentryprefixfirst{#2}%
8479         \glsprefixsep
8480         \@gls@{#1}{#2}[#3]%
8481       }%
8482       {\@Gls@{#1}{#2}[#3]}%
8483     }%
8484   }%
8485 }
```

Similarly for the plural version:

\Pglpl

```
8486 \newrobustcmd{\Pglpl}{\@gls@hyp@opt\Pglpl}
```

\@Pglpl Unstarred version.

```
8487 \newcommand*{\@Pglpl}[2][]{%
8488   \new@ifnextchar[%
8489     {\@Pglpl@{#1}{#2}}%
8490     {\@Pglpl@{#1}{#2} []}%
8491 }
```

\@Pglsp1@ Read in the final optional argument:

```

8492 \def\@Pglsp1@#1#2[#3]{%
8493   \@glsprefix@record@hook{#1}{#2}%
8494   \glsoifexists{#2}%
8495   {%
8496     \ifglused{#2}%
8497     {%
8498       \ifglshasprefixplural{#2}%
8499       {%
8500         \Glsentryprefixplural{#2}%
8501         \glsprefixsep
8502         \@glsp1@{#1}{#2}[#3]%
8503       }%
8504       {\@Glspl@{#1}{#2}[#3]}%
8505     }%
8506     {%
8507       \ifglshasprefixfirstplural{#2}%
8508       {%
8509         \Glsentryprefixfirstplural{#2}%
8510         \glsprefixsep
8511         \@glsp1@{#1}{#2}[#3]%
8512       }%
8513       {\@Glspl@{#1}{#2}[#3]}%
8514     }%
8515   }%
8516 }

```

Finally the all upper case versions:

\PGLS

```

8517 \newrobustcmd{\PGLS}{\@glshyp@opt\@PGLS}

```

\@PGLS Unstarred version.

```

8518 \newcommand*{\@PGLS}[2][ ]{%
8519   \new@ifnextchar[%
8520   {\@PGLS@{#1}{#2}}%
8521   {\@PGLS@{#1}{#2}[ ]}%
8522 }

```

\@PGLS@ Read in the final optional argument:

```

8523 \def\@PGLS@#1#2[#3]{%
8524   \@glsprefix@record@hook{#1}{#2}%
8525   \glsoifexists{#2}%
8526   {%
8527     \ifglused{#2}%
8528     {%
8529       \ifglshasprefix{#2}%
8530       {\mfirstucMakeUppercase{\Glsentryprefix{#2}\glsprefixsep}}{%

```

```

8531 }%
8532 {%
8533     \ifglshasprefixfirst{#2}%
8534     {\mfirstucMakeUppercase{\glstryentryprefixfirst{#2}\glsprefixsep}}{}%
8535 }%
8536 \@GLS@{#1}{#2}[#3]%
8537 }%
8538 }

```

Plural version:

\PGLSp1

```

8539 \newrobustcmd{\PGLSp1}{\@glshyp@opt\PGLSp1}

```

\@PGLSp1 Unstarred version.

```

8540 \newcommand*{\@PGLSp1}[2][ ]{%
8541     \new@ifnextchar[%
8542     {\@PGLSp1@{#1}{#2}}%
8543     {\@PGLSp1@{#1}{#2}[ ]}%
8544 }

```

\@PGLSp1@ Read in the final optional argument:

```

8545 \def\@PGLSp1@#1#2[#3]{%
8546     \@glsprefix@record@hook{#1}{#2}%
8547     \glstdoifexists{#2}%
8548     {%
8549         \ifglused{#2}%
8550         {%
8551             \ifglshasprefixplural{#2}%
8552             {\mfirstucMakeUppercase{\glstryentryprefixplural{#2}\glsprefixsep}}{}%
8553         }%
8554         {%
8555             \ifglshasprefixfirstplural{#2}%
8556             {\mfirstucMakeUppercase{\glstryentryprefixfirstplural{#2}\glsprefixsep}}{}%
8557         }%
8558         \@GLSp1@{#1}{#2}[#3]%
8559     }%
8560 }

```

## 3 Glossary Styles

### 3.1 Glossary hyper-navigation definitions (glossary-hypernav package)

Package Definition:

```
8561 \ProvidesPackage{glossary-hypernav}[2021/11/01 v4.49 (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`

```
8562 \newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
```

```
8563   \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
```

```
8564   \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
```

`\glsnavhyperlinkname`

Expands to the hypertarget name. The first argument is the glossary type. The second argument is the group label.

```
8565 \newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
```

`\glsnavhypertarget`

```
\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.

```
8566 \newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
```

```
8567   \@glsnavhypertarget{#1}{#2}{#3}%
```

```
8568 }
```

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

```
8569 \newcommand*{\@glsnavhypertarget}[3]{%
    Add this group to the aux file for re-run check.
8570   \protected@write\auxout{}\string\@gls@hypergroup{#1}{#2}}%
    Add the target.
8571   \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
    Check list of known groups to determine if a re-run is required.
8572   \expandafter\let
8573     \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
    Iterate through list and terminate loop if this group is found.
8574   \@for\@gls@elem:=\@gls@list\do{%
8575     \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
    Check if list terminated prematurely.
8576   \if@endfor
8577   \else
    This group was not included in the list, so issue a warning.
8578     \GlossariesWarningNoLine{Navigation panel
8579       for glossary type ‘#1’~Jmissing group ‘#2’}%
8580     \gdef\gls@hypergroup@rerun{%
8581       \GlossariesWarningNoLine{Navigation panel
8582         has changed. Rerun LaTeX}}%
8583   \fi
8584 }
```

hypergroup@rerun Give a warning at the end if re-run required

```
8585 \let\gls@hypergroup@rerun\relax
8586 \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.

```
8587 \newcommand*{\@gls@hypergroup}[2]{%
8588   \@ifundefined{\@gls@hypergroup@list@#1}{%
8589     \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
8590   }{%
8591     \expandafter\let\expandafter\@gls@tmp
8592       \csname @gls@hypergroup@list@#1\endcsname
8593     \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
8594       \@gls@tmp,#2}%
8595   }%
8596 }
```

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`

```

8597 \newcommand*{\glsnavigation}{%
8598   \def\@gls@between{}%
8599   \ifcsundef{\@gls@hypergroup@list@\@glo@type}%
8600   {%
8601     \def\@gls@list{}%
8602   }%
8603   {%
8604     \expandafter\let\expandafter\@gls@list
8605       \csname \@gls@hypergroup@list@\@glo@type\endcsname
8606   }%
8607   \@for\@gls@tmp:=\@gls@list\do{%
8608     \@gls@between

8609     \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
8610     \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
8611     \let\@gls@between\glshypernavsep
8612   }%
8613 }
```

`\glshypernavsep` Separator for the hyper navigation bar.

```

8614 \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`

```

8615 \newcommand*{\glssymbolnav}{%
8616   \glsnavhyperlink{glssymbols}{\@gls@getgrouptitle{glssymbols}}%
8617   \glshypernavsep
8618   \glsnavhyperlink{glsnumbers}{\@gls@getgrouptitle{glsnumbers}}%
8619   \glshypernavsep
8620 }
```

## 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.

```

8621 \ProvidesPackage{glossary-inline}[2021/11/01 v4.49 (NLCT)]
```

`inline` Define the inline style.

```

8622 \newglossarystyle{inline}{%
```

Start of glossary sets up first empty separator between entries. (This is then changed by `\glossentry`)

```
8623 \renewenvironment{theglossary}%
8624 {%
8625     \def\gls@inlinesep{}%
8626     \def\gls@inlinesubsep{}%
8627     \def\gls@inlinepostchild{}%
8628 }%
8629 {\glspostinline}%
```

No header:

```
8630 \renewcommand*{\glossaryheader}{}%
```

No group headings (if heading is required, add `\glsinlinedopostchild` to start definition in case heading follows a child entry):

```
8631 \renewcommand*{\glsgroupheading}[1]{}%
```

Just display separator followed by name and description:

```
8632 \renewcommand{\glossentry}[2]{%
8633     \glsinlinedopostchild
8634     \gls@inlinesep
8635     \glsentryitem{##1}%
8636     \glsinlinenameformat{##1}%
8637     \glossentryname{##1}%
8638 }%
8639 \ifglshasdescsuppressed{##1}%
8640 {%
8641     \glsinlineemptydescformat
8642     {%
8643         \glossentrysymbol{##1}%
8644     }%
8645     {%
8646         ##2%
8647     }%
8648 }%
8649 {%
8650     \ifglshasdesc{##1}%
8651     {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
8652     {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
8653 }%
8654 \ifglshaschildren{##1}%
8655 {%
8656     \glsresetsubentrycounter
8657     \glsinlineparentchildseparator
8658     \def\gls@inlinesubsep{}%
8659     \def\gls@inlinepostchild{\glsinlinepostchild}%
8660 }%
8661 {}%
8662 \def\gls@inlinesep{\glsinlineseparator}%
8663 }%
```

Sub-entries display description:

```
8664 \renewcommand{\subglossentry}[3]{%
8665   \gls@inlinesubsep%
8666   \glsinlinesubnameformat{##2}{%
8667     \glossentryname{##2}}%
8668   \glssubentryitem{##2}%
8669   \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
8670   \def\gls@inlinesubsep{\glsinlinesubseparator}%
8671 }%
```

Nothing special between groups:

```
8672 \renewcommand*{\glsgroupskip}{}%
8673 }
```

linedopostchild

```
8674 \newcommand*{\glsinlinedopostchild}{%
8675   \gls@inlinepostchild
8676   \def\gls@inlinepostchild{}%
8677 }
```

inlineseparator Separator to use between entries.

```
8678 \newcommand*{\glsinlineseparator}{;\space}
```

inlinesubseparator Separator to use between sub-entries.

```
8679 \newcommand*{\glsinlinesubseparator}{,\space}
```

parentchildseparator Separator to use between parent and children.

```
8680 \newcommand*{\glsinlineparentchildseparator}{:\space}
```

inlinepostchild Hook to use between child and next entry

```
8681 \newcommand*{\glsinlinepostchild}{}
```

\glspostinline Terminator for inline glossary.

```
8682 \newcommand*{\glspostinline}{\glspostdescription\space}
```

inlinenameformat Formats the name of the entry (first argument label, second argument name):

```
8683 \newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
```

inlinedescformat Formats the entry's description, symbol and location list:

```
8684 \newcommand*{\glsinlinedescformat}[3]{\space#1}
```

emptydescformat Formats the entry's symbol and location list when the description is empty:

```
8685 \newcommand*{\glsinlineemptydescformat}[2]{}
```

inlinesubnameformat Formats the name of the subentry (first argument label, second argument name):

```
8686 \newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
```

inlinesubdescformat Formats the subentry's description, symbol and location list:

```
8687 \newcommand*{\glsinlinesubdescformat}[3]{#1}
```

### 3.3 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
8688 \ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
8689 \providecommand{\indexspace}{%
8690   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
8691 }
8692 \newcommand*{\glslistgroupheaderfmt}[1]{#1}
8693 \newcommand*{\glslistnavigationitem}[1]{\item[#1]}
8694 \newglossarystyle{list}{%
8695   \renewenvironment{theglossary}%
8696     {\begin{description}}{\end{description}}%
8697   \renewcommand*{\glossaryheader}{}%
8698   \renewcommand*{\glsgroupheading}[1]{}%
8699   \renewcommand*{\glossentry}[2]{%
8700     \item[\glsentryitem{##1}%
8701       \glstarget{##1}{\glossentryname{##1}}]
8702     \glossentrydesc{##1}\glspostdescription\space ##2}%
8703   \renewcommand*{\subglossentry}[3]{%
8704     \glssubentryitem{##2}%
8705     \glstarget{##2}{\strut}\space
8706     \glossentrydesc{##2}\glspostdescription\space ##3.}%
8707   \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
8708 }
8709 \newglossarystyle{listgroup}{%
8710   \setglossarystyle{list}%
8711   \renewcommand*{\glsgroupheading}[1]{%
8712     \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
8713 \newglossarystyle{listhypergroup}{%
8714   \setglossarystyle{list}%
8715   \renewcommand*{\glossaryheader}{%
8716     \glslistnavigationitem{\glsnavigation}}%
8717   \renewcommand*{\glsgroupheading}[1]{%
8718     \item[\glslistgroupheaderfmt
8719       {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
8720 \newglossarystyle{altlist}{%
8721   \setglossarystyle{list}%
8722   \renewcommand*{\glossentry}[2]{%
8723     \item[\glsentryitem{##1}%
8724       \glstarget{##1}{\glossentryname{##1}}]
8725     \mbox{}\par\nobreak\@afterheading
8726     \glossentrydesc{##1}\glspostdescription\space ##2}%
8727   \renewcommand*{\subglossentry}[3]{%
8728     \par
8729     \glssubentryitem{##2}%
8730     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
8731 }
8732 \newglossarystyle{altlistgroup}{%
8733   \setglossarystyle{altlist}%

```

```

8734 \renewcommand*{\glsgroupheading}[1]{%
8735   \item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
8736 \newglossarystyle{altlisthypergroup}{%
8737   \setglossarystyle{altlist}%
8738   \renewcommand*{\glossaryheader}{%
8739     \glslistnavigationitem{\glsnavigation}}%
8740   \renewcommand*{\glsgroupheading}[1]{%
8741     \item[\glslistgroupheaderfmt
8742       {\glsnavhypertarget{##1}{\glsgrouptitle{##1}}]}
8743 \newglossarystyle{listdotted}{%
8744   \setglossarystyle{list}%
8745   \renewcommand*{\glossentry}[2]{%
8746     \item[]\makebox[\glslistdottedwidth][l]{%
8747       \glstryitem{##1}%
8748       \glstarget{##1}{\glossentryname{##1}}%
8749       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
8750   \renewcommand*{\subglossentry}[3]{%
8751     \item[]\makebox[\glslistdottedwidth][l]{%
8752       \glssubentryitem{##2}%
8753       \glstarget{##2}{\glossentryname{##2}}%
8754       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
8755 }
8756 \newlength\glslistdottedwidth
8757 \setlength{\glslistdottedwidth}{.5\hsize}
8758 \newglossarystyle{sublistdotted}{%
8759   \setglossarystyle{listdotted}%
8760   \renewcommand*{\glossentry}[2]{%
8761     \item[\glstryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
8762 }

```

### 3.4 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.

```
8763 \NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
8764 \DeclareRelease{v4.46}{2020-03-19}{glossary-list-2020-03-19.sty}
```

```
8765 \DeclareCurrentRelease{v4.49}{2021-11-01}
```

Declare package:

```
8766 \ProvidesPackage{glossary-list}[2021/11/01 v4.49 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```

8767 \providecommand{\indexspace}{%
8768   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
8769 }

```

`groupheaderfmt` Provide a way of adjusting the format of the group headings.

```
8770 \newcommand*{\glslistgroupheaderfmt}[1]{#1}
```

`navigationitem` 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.

```
8771 \newcommand*{\glslistnavigationitem}[1]{\item[#1]}
```

`\glslistinit` Provide a way to ensure appropriate expansion of the name if `entrycounter` is used with `getttitlestring`:

```
8772 \newcommand{\glslistinit}{%
8773   \ifdef\GetTitleStringDisableCommands
8774     {%
8775       \GetTitleStringSetup{expand}%
8776       \GetTitleStringDisableCommands{%
8777         \let\glsentryitem@gobble
8778         \let\glstarget@secondoftwo
8779         \let\glossentryname\glslistexpandedname
8780         \let\glslistgroupheaderfmt@firstofone
8781         \let\glsgetgrouptitle@firstofone
8782         \let\glsnavhypertarget@secondoftwo
8783         \let\glsnavigation\relax
8784       }%
8785     }%
8786   }%
8787 }
```

`listexpandedname`

```
8788 \newcommand{\glslistexpandedname}[1]{%
8789   \glsunexpandedfieldvalue{#1}{name}%
8790 }
```

`list` The list glossary style uses the description environment. The group separator `\glsgroupskip` is redefined as `\indexspace` which produces a gap between groups. The glossary heading and the group headings do nothing. Sub-entries immediately follow the main entry without the sub-entry name. This style does not use the entry's symbol. This is used as the default style for the glossaries package.

```
8791 \newglossarystyle{list}{%
  Use description environment:
8792   \renewenvironment{theglossary}%
8793     {\glslistinit\begin{description}}{\end{description}}%
  No header at the start of the environment:
8794   \renewcommand*{\glossaryheader}{}%
}
```

No group headings:

```
8795 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries start a new item in the list:

```
8796 \renewcommand*{\glossentry}[2]{%
8797   \item[\glstryitem{##1}%
8798     \glstarget{##1}{\glossentryname{##1}}]
8799   \glossentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries continue on the same line:

```
8800 \renewcommand*{\subglossentry}[3]{%
8801   \glssubentryitem{##2}%
8802   \glstarget{##2}{\strut}\space
8803   \glossentrydesc{##2}\glspostdescription\space ##3.}%
```

Add vertical space between groups:

```
8804 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
8805 }
```

**listgroup** The listgroup style is like the list style, but the glossary groups have headings.

```
8806 \newglossarystyle{listgroup}{%
```

Base it on the list style:

```
8807 \setglossarystyle{list}%
```

Each group has a heading:

```
8808 \renewcommand*{\glsgroupheading}[1]{%
8809   \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.

```
8810 \newglossarystyle{listhypergroup}{%
```

Base it on the list style:

```
8811 \setglossarystyle{list}%
```

Add navigation links at the start of the environment.

```
8812 \renewcommand*{\glossaryheader}{%
8813   \glslistnavigationitem{\glslnavigation}}%
```

Each group has a heading with a hypertext:

```
8814 \renewcommand*{\glsgroupheading}[1]{%
8815   \item[\glslistgroupheaderfmt
8816     {\glslnavhypertarget{##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.

```
8817 \newglossarystyle{altlist}{%
```

Base it on the list style:

```
8818 \setglossarystyle{list}%
```



Main (level 0) entries start a new item in the list with a line break after the entry name:

```
8819 \renewcommand*{\glossentry}[2]{%
8820   \item[\glentryitem{##1}%
8821     \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.

```
8822   \mbox{}\par\nobreak\@afterheading
8823   \glossentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries start a new paragraph:

```
8824 \renewcommand{\subglossentry}[3]{%
8825   \par
8826   \glssubentryitem{##2}%
8827   \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
8828 }
```

`altlistgroup` The `altlistgroup` glossary style is like the `altlist` style, but the glossary groups have headings.

```
8829 \newglossarystyle{altlistgroup}{%
```

Base it on the `altlist` style:

```
8830 \setglossarystyle{altlist}%
```

Each group has a heading:

```
8831 \renewcommand*{\glsgroupheading}[1]{%
8832   \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.

```
8833 \newglossarystyle{altlisthypergroup}{%
```

Base it on the `altlist` style:

```
8834 \setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
8835 \renewcommand*{\glossaryheader}{%
8836   \glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a `hypertarget`:

```
8837 \renewcommand*{\glsgroupheading}[1]{%
8838   \item[\glslistgroupheaderfmt
8839     {\glsnavigationhypertarget{##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.

```
8840 \newglossarystyle{listdotted}{%
```

Base it on the `list` style:

```
8841 \setglossarystyle{list}%
```

Each main (level 0) entry starts a new item:

```

8842 \renewcommand*{\glossentry}[2]{%
8843   \item[]\makebox[\glslistdottedwidth][l]{%
8844     \glsentryitem{##1}%
8845     \glstarget{##1}{\glossentryname{##1}}%
8846     \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%

```

Sub entries have the same format as main entries:

```

8847 \renewcommand*{\subglossentry}[3]{%
8848   \item[]\makebox[\glslistdottedwidth][l]{%
8849     \glssubentryitem{##2}%
8850     \glstarget{##2}{\glossentryname{##2}}%
8851     \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
8852 }

```

listdottedwidth

```

8853 \newlength\glslistdottedwidth
8854 \setlength{\glslistdottedwidth}{.5\hsize}

```

sublistdotted This style is similar to the `glostylelistdotted` style, except that the main entries just have the name displayed.

```
8855 \newglossarystyle{sublistdotted}{%
```

Base it on the `listdotted` style:

```
8856 \setglossarystyle{listdotted}%
```

Main (level 0) entries just display the name:

```

8857 \renewcommand*{\glossentry}[2]{%
8858   \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}}%
8859 }

```

### 3.5 Glossary Styles using `longtable` (the `glossary-long` package)

The glossary styles defined in the package used the `longtable` environment in the glossary.

```
8860 \ProvidesPackage{glossary-long}[2021/11/01 v4.49 (NLCT)]
```

Requires the package:

```
8861 \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`.)

```

8862 \@ifundefined{glsdescwidth}{%
8863   \newlength\glsdescwidth
8864   \setlength{\glsdescwidth}{0.6\hsize}
8865 }{}

```

`lspagelistwidth` This is a length that governs the width of the page list column.

```
8866 \ifundefined{glspagelistwidth}{%
8867   \newlength{glspagelistwidth
8868   \setlength{glspagelistwidth}{0.1\hsize}
8869 }{}
```

`long` The long glossary style command which uses the `longtable` environment:

```
8870 \newglossarystyle{long}{%
```

Use `longtable` with two columns:

```
8871 \renewenvironment{theglossary}{%
8872   {\begin{longtable}{lp{glsgdescwidth}}}%
8873   {\end{longtable}}%
```

Do nothing at the start of the environment:

```
8874 \renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
8875 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
8876 \renewcommand{\glossentry}[2]{%
8877   \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
8878   \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
8879 }%
```

Sub entries displayed on the following row without the name:

```
8880 \renewcommand{\subglossentry}[3]{%
8881   &
8882   \glssubentryitem{##2}%
8883   \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
8884   ##3\tabularnewline
8885 }%
```

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>)

```
8886 \ifglsgnogroupskip
8887   \renewcommand*{\glsgroupskip}{}%
8888 \else
8889   \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
8890 \fi
8891 }
```

`longborder` The `longborder` style is like the above, but with horizontal and vertical lines:

```
8892 \newglossarystyle{longborder}{%
```

Base it on the `glostylelong` style:

```
8893 \setglossarystyle{long}%
```

Use `longtable` with two columns with vertical lines between each column:

```
8894 \renewenvironment{theglossary}{%
8895   \begin{longtable}{|lp{glsgdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
8896 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8897 }
```

`longheader` The `longheader` style is like the `long` style but with a header:

```
8898 \newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
8899 \setglossarystyle{long}%
```

Set the table's header:

```
8900 \renewcommand*{\glossaryheader}{%
8901 \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
8902 }
```

`longheaderborder` The `longheaderborder` style is like the `long` style but with a header and border:

```
8903 \newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
8904 \setglossarystyle{longborder}%
```

Set the table's header and add horizontal line to table's foot:

```
8905 \renewcommand*{\glossaryheader}{%
8906 \hline\bfseries \entryname & \bfseries
8907 \descriptionname\tabularnewline\hline
8908 \endhead
8909 \hline\endfoot}%
8910 }
```

`long3col` The `long3col` style is like `long` but with 3 columns

```
8911 \newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
8912 \renewenvironment{theglossary}%
8913 {\begin{longtable}{lp{\glstdescwidth}p{\glspagelistwidth}}}%
8914 {\end{longtable}}%
```

No table header:

```
8915 \renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
8916 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
8917 \renewcommand{\glossentry}[2]{%
8918 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
8919 \glossentrydesc{##1} & ##2\tabularnewline
8920 }%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
8921 \renewcommand{\subglossentry}[3]{%
8922     &
8923     \glssubentryitem{##2}%
8924     \glstarget{##2}{\strut}\glossentrydesc{##2} &
8925     ##3\tabularnewline
8926 }%
```

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>)

```
8927 \ifglsgroupskip
8928 \renewcommand*{\glsgroupskip}{}%
8929 \else
8930 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
8931 \fi
8932 }
```

**long3colborder** The long3colborder style is like the long3col style but with a border:

```
8933 \newglossarystyle{long3colborder}{%
    Base it on the glostylelong3col style:
8934 \setglossarystyle{long3col}%
    Use a longtable with 3 columns with vertical lines around them:
8935 \renewenvironment{theglossary}%
8936 {\begin{longtable}{|l|p{\glstdescwidth}|p{\glspagelistwidth}|}%
8937 {\end{longtable}}%
    Place horizontal lines at the head and foot of the table:
8938 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8939 }
```

**long3colheader** The long3colheader style is like long3col but with a header row:

```
8940 \newglossarystyle{long3colheader}{%
    Base it on the glostylelong3col style:
8941 \setglossarystyle{long3col}%
    Set the table's header:
8942 \renewcommand*{\glossaryheader}{%
8943     \bfseries\entryname&\bfseries\descriptionname&
8944     \bfseries\pagelistname\tabularnewline\endhead}%
8945 }
```

**colheaderborder** The long3colheaderborder style is like the above but with a border

```
8946 \newglossarystyle{long3colheaderborder}{%
    Base it on the glostylelong3colborder style:
8947 \setglossarystyle{long3colborder}%
}
```

Set the table's header and add horizontal line at table's foot:

```
8948 \renewcommand*{\glossaryheader}{%
8949 \hline
8950 \bfseries\entryname&\bfseries\descriptionname&
8951 \bfseries\pagelistname\tabularnewline\hline\endhead
8952 \hline\endfoot}%
8953 }
```

`long4col` The `long4col` style has four columns where the third column contains the value of the associated symbol key.

```
8954 \newglossarystyle{long4col}{%
```

Use a longtable with 4 columns:

```
8955 \renewenvironment{theglossary}%
8956 {\begin{longtable}{l1l1l}}%
8957 {\end{longtable}}%
```

No table header:

```
8958 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
8959 \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):

```
8960 \renewcommand{\glossentry}[2]{%
8961 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
8962 \glossentrydesc{##1} &
8963 \glossentrysymbol{##1} &
8964 ##2\tabularnewline
8965 }%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
8966 \renewcommand{\subglossentry}[3]{%
8967 &
8968 \glssubentryitem{##2}%
8969 \glstarget{##2}{\strut}\glossentrydesc{##2} &
8970 \glossentrysymbol{##2} & ##3\tabularnewline
8971 }%
```

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>)

```
8972 \ifglsnogroupskip
8973 \renewcommand*{\glsgroupskip}{}%
8974 \else
8975 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
8976 \fi
8977 }
```

**long4colheader** The long4colheader style is like long4col but with a header row.

```

8978 \newglossarystyle{long4colheader}{%
      Base it on the glostylelong4col style:
8979   \setglossarystyle{long4col}%
      Table has a header:
8980   \renewcommand*{\glossaryheader}{%
8981     \bfseries\entryname&\bfseries\descriptionname&
8982     \bfseries \symbolname&
8983     \bfseries\pagelistname\tabularnewline\endhead}%
8984 }
```

**long4colborder** The long4colborder style is like long4col but with a border.

```

8985 \newglossarystyle{long4colborder}{%
      Base it on the glostylelong4col style:
8986   \setglossarystyle{long4col}%
      Use a longtable with 4 columns surrounded by vertical lines:
8987   \renewenvironment{theglossary}%
8988     {\begin{longtable}{|l|l|l|l|}}%
8989     {\end{longtable}}%
      Add horizontal lines to the head and foot of the table:
8990   \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8991 }
```

**colheaderborder** The long4colheaderborder style is like the above but with a border.

```

8992 \newglossarystyle{long4colheaderborder}{%
      Base it on the glostylelong4col style:
8993   \setglossarystyle{long4col}%
      Use a longtable with 4 columns surrounded by vertical lines:
8994   \renewenvironment{theglossary}%
8995     {\begin{longtable}{|l|l|l|l|}}%
8996     {\end{longtable}}%
      Add table header and horizontal line at the table's foot:
8997   \renewcommand*{\glossaryheader}{%
8998     \hline\bfseries\entryname&\bfseries\descriptionname&
8999     \bfseries \symbolname&
9000     \bfseries\pagelistname\tabularnewline\hline\endhead
9001     \hline\endfoot}%
9002 }
```

**altlong4col** The altlong4col style is like the long4col style but can have multiline descriptions and page lists.

```

9003 \newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
9004 \setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9005 \renewenvironment{theglossary}%  
9006 {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%  
9007 {\end{longtable}}%  
9008 }
```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```
9009 \newglossarystyle{altlong4colheader}{%
```

Base it on the `glostylelong4colheader` style:

```
9010 \setglossarystyle{long4colheader}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9011 \renewenvironment{theglossary}%  
9012 {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%  
9013 {\end{longtable}}%  
9014 }
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```
9015 \newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
9016 \setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9017 \renewenvironment{theglossary}%  
9018 {\begin{longtable}{|lp{\glstdescwidth}|lp{\glspagelistwidth}|}}}%  
9019 {\end{longtable}}%  
9020 }
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
9021 \newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `glostylelong4colheaderborder` style:

```
9022 \setglossarystyle{long4colheaderborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9023 \renewenvironment{theglossary}%  
9024 {\begin{longtable}{|lp{\glstdescwidth}|lp{\glspagelistwidth}|}}}%  
9025 {\end{longtable}}%  
9026 }
```



## 3.6 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>

```
9027 \ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
```

Requires booktabs package:

```
9028 \RequirePackage{booktabs}
```

and the base packages for long styles:

```
9029 \RequirePackage{glossary-long}
```

```
9030 \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.

```
9031 \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.

```
9032 \glspatchLToutput
```

As with the longheader style, use the long style as a base.

```
9033 \setglossarystyle{long}{%
```

Add a header with rules.

```
9034 \renewcommand*{\glossaryheader}{%
```

```
9035 \toprule \bfseries \entryname & \bfseries
```

```
9036 \descriptionname\tabularnewline\midrule\endhead
```

```
9037 \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.

```
9038 \ifglsgnogroupskip
```

```
9039 \renewcommand*{\glsgroupskip}{}%
```

```
9040 \else
```

```
9041 \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
```

```
9042 \fi
```

```
9043 }
```

**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.

```
9044 \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.

```
9045 \glspatchLToutput
```

Use the long3col style as a base.

```
9046 \setglossarystyle{long3col}%
```

Add a header with rules.

```
9047 \renewcommand*{\glossaryheader}{%
9048   \toprule \bfseries \entryname &
9049   \bfseries \descriptionname &
9050   \bfseries \pagelistname
9051   \tabularnewline\midrule\endhead
9052   \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.

```
9053 \ifglsgnogroupskip
9054   \renewcommand*{\glsgroupskip}{}%
9055 \else
9056   \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
9057 \fi
9058 }
```

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.

```
9059 \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.

```
9060 \glspatchLToutput
```

Use the long4col style as a base.

```
9061 \setglossarystyle{long4col}%
```

Add a header with rules.

```
9062 \renewcommand*{\glossaryheader}{%
9063   \toprule \bfseries \entryname &
9064   \bfseries \descriptionname &
9065   \bfseries \symbolname &
9066   \bfseries \pagelistname
9067   \tabularnewline\midrule\endhead
9068   \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.

```
9069 \ifglsgnogroupskip
9070   \renewcommand*{\glsgroupskip}{}%
9071 \else
9072   \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
9073 \fi
9074 }
```

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.

```
9075 \newglossarystyle{altlong4col-booktabs}{%
```

The patch \glspatchLToutput is already applied in long4col-booktabs and so doesn't need to be here.

```
9076 \glspatchLToutput
```

Use the long4col-booktabs style as a base.

```
9077 \setglossarystyle{long4col-booktabs}{%
```

Change the column specifications:

```
9078 \renewenvironment{theglossary}{%
```

```
9079 {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
```

```
9080 {\end{longtable}}}%
```

```
9081 }
```

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.

```
9082 \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.

```
9083 \glspatchLToutput
```

Use the long-booktabs style as a base.

```
9084 \setglossarystyle{long-booktabs}{%
```

Adjust the column specification.

```
9085 \renewenvironment{theglossary}{%
```

```
9086 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}}}%
```

```
9087 {\end{longtable}}}%
```

```
9088 }
```

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.

```
9089 \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.

```
9090 \glspatchLToutput
```

Use the long3col-booktabs style as a base.

```
9091 \setglossarystyle{long3col-booktabs}{%
```

Adjust the column specification.

```
9092 \renewenvironment{theglossary}{%
```

```
9093 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}}%
```

```
9094 >{\raggedright}p{\glspagelistwidth}}}%
```

```
9095 {\end{longtable}}}%
```

```
9096 }
```

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.

```
9097 \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.

```
9098 \glspatchLToutput
```

Use the altlong4col-booktabs style as a base.

```
9099 \setglossarystyle{altlong4col-booktabs}%
```

Adjust the column specification.

```
9100 \renewenvironment{theglossary}%
```

```
9101 {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
```

```
9102 >{\raggedright}p{\glspagelistwidth}}}%
```

```
9103 {\end{longtable}}%
```

```
9104 }
```

sLTpenaltycheck

```
9105 \newcommand*{\glslTpenaltycheck}{%
```

```
9106 \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
```

```
9107 }
```

enaltygroupskip

```
9108 \newcommand{\glspenaltygroupskip}{%
```

```
9109 \noalign{\penalty-50\vskip\normalbaselineskip}}
```

restoreLToutput Provide a way of restoring \LT@output for the user.

```
9110 \let\@gls@org@LT@output\LT@output
```

```
9111 \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

```
9112 \newcommand*{\glspatchLToutput}{%
```

```
9113 \renewcommand*{\LT@output}{%
```

```
9114 \ifnum\outputpenalty <-\@Mi
```

```
9115 \ifnum\outputpenalty > -\LT@end@pen
```

```
9116 \LT@err{floats and marginpars not allowed in a longtable}\@ehc
```

```
9117 \else
```

```
9118 \setbox\z@\vbox{\unvbox\@cclv}%
```

```
9119 \ifdim \ht\LT@lastfoot>\ht\LT@foot
```

```
9120 \dimen@\pagegoal
```

```
9121 \advance\dimen@-\ht\LT@lastfoot
```

```
9122 \ifdim\dimen@<\ht\z@
```

```
9123 \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
```

```
9124 \makecol
```

```
9125 \@outputpage
```

```

9126         \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
9127     \fi
9128 \fi
9129 \global\@colroom\@colht
9130 \global\vsizel\@colht
9131 {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
9132 \fi
9133 \else
9134 \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
9135 \@makecol
9136 \@outputpage
9137 \global\vsizel\@colroom
9138 \copy\LT@head
9139 \glsLTpenaltycheck
9140 \nobreak
9141 \fi
9142 }%
9143 }

```

### 3.7 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.

9144 \ProvidesPackage{glossary-longragged}[2021/11/01 v4.49 (NLCT)]

Requires the package:

9145 \RequirePackage{array}

Requires the package:

9146 \RequirePackage{longtable}

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```

9147 \@ifundefined{glsdescwidth}{%
9148 \newlength\glsdescwidth
9149 \setlength{\glsdescwidth}{0.6\hsize}
9150 }{}

```

`lspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```

9151 \@ifundefined{glspagelistwidth}{%
9152 \newlength\glspagelistwidth
9153 \setlength{\glspagelistwidth}{0.1\hsize}
9154 }{}

```

`longragged` The longragged glossary style is like the long but uses ragged right formatting for the description column.

```
9155 \newglossarystyle{longragged}{%
```

Use longtable with two columns:

```
9156 \renewenvironment{theglossary}%
9157     {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%
9158     {\end{longtable}}%
```

Do nothing at the start of the environment:

```
9159 \renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
9160 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
9161 \renewcommand{\glossentry}[2]{%
9162     \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9163     \glossentrydesc{##1}\glspostdescription\space ##2%
9164     \tabularnewline
9165 }%
```

Sub entries displayed on the following row without the name:

```
9166 \renewcommand{\subglossentry}[3]{%
9167     &
9168     \glssubentryitem{##2}%
9169     \glstarget{##2}{\strut}\glossentrydesc{##2}%
9170     \glspostdescription\space ##3%
9171     \tabularnewline
9172 }%
```

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>)

```
9173 \ifglsgroupskip
9174     \renewcommand*{\glsgroupskip}{}%
9175 \else
9176     \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
9177 \fi
9178 }
```

`ongraggedborder` The `longraggedborder` style is like the above, but with horizontal and vertical lines:

```
9179 \newglossarystyle{longraggedborder}{%
```

Base it on the `glostylelongragged` style:

```
9180 \setglossarystyle{longragged}%
```

Use longtable with two columns with vertical lines between each column:

```
9181 \renewenvironment{theglossary}{%
9182     \begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|}%
9183     {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
9184 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9185 }
```

longraggedheader The longraggedheader style is like the longragged style but with a header:

```
9186 \newglossarystyle{longraggedheader}{%
    Base it on the glostylelongragged style:
9187 \setglossarystyle{longragged}%
    Set the table's header:
9188 \renewcommand*{\glossaryheader}{%
9189 \bfseries \entryname & \bfseries \descriptionname
9190 \tabularnewline\endhead}%
9191 }
```

longraggedheaderborder The longraggedheaderborder style is like the longragged style but with a header and border:

```
9192 \newglossarystyle{longraggedheaderborder}{%
    Base it on the glostylelongraggedborder style:
9193 \setglossarystyle{longraggedborder}%
    Set the table's header and add horizontal line to table's foot:
9194 \renewcommand*{\glossaryheader}{%
9195 \hline\bfseries \entryname & \bfseries \descriptionname
9196 \tabularnewline\hline
9197 \endhead
9198 \hline\endfoot}%
9199 }
```

longragged3col The longragged3col style is like longragged but with 3 columns

```
9200 \newglossarystyle{longragged3col}{%
    Use a longtable with 3 columns:
9201 \renewenvironment{theglossary}%
9202 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}%
9203 >{\raggedright}p{\glspagelistwidth}}}%
9204 {\end{longtable}}%
    No table header:
9205 \renewcommand*{\glossaryheader}{}%
    No headings between groups:
9206 \renewcommand*{\glsgroupheading}[1]{}%
    Main (level 0) entries on a row (name in first column, description in second column, page list
    in last column):
9207 \renewcommand{\glossentry}[2]{%
9208 \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9209 \glossentrydesc{##1} & ##2\tabularnewline
9210 }%
    Sub-entries on a separate row (no name, description in second column, page list in third
    column):
9211 \renewcommand{\subglossentry}[3]{%
9212 &
```

```

9213 \glssubentryitem{##2}%
9214 \glstarget{##2}{\strut}\glossentrydesc{##2} &
9215 ##3\tabularnewline
9216 }%

```

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>)

```

9217 \ifglsgroupskip
9218 \renewcommand*{\glsgroupskip}{}%
9219 \else
9220 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
9221 \fi
9222 }

```

agged3colborder The longragged3colborder style is like the longragged3col style but with a border:

```

9223 \newglossarystyle{longragged3colborder}{%
  Base it on the glostylelongragged3col style:
9224 \setglossarystyle{longragged3col}%
  Use a longtable with 3 columns with vertical lines around them:
9225 \renewenvironment{theglossary}%
9226 {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|%
9227 >{\raggedright}p{\glspagelistwidth}|}%
9228 {\end{longtable}}%

```

Place horizontal lines at the head and foot of the table:

```

9229 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9230 }

```

agged3colheader The longragged3colheader style is like longragged3col but with a header row:

```

9231 \newglossarystyle{longragged3colheader}{%
  Base it on the glostylelongragged3col style:
9232 \setglossarystyle{longragged3col}%
  Set the table's header:
9233 \renewcommand*{\glossaryheader}{%
9234 \bfseries\entryname&\bfseries\descriptionname&
9235 \bfseries\pagelistname\tabularnewline\endhead}%
9236 }

```

colheaderborder The longragged3colheaderborder style is like the above but with a border

```

9237 \newglossarystyle{longragged3colheaderborder}{%
  Base it on the glostylelongragged3colborder style:
9238 \setglossarystyle{longragged3colborder}%
  Set the table's header and add horizontal line at table's foot:
9239 \renewcommand*{\glossaryheader}{%
9240 \hline
9241 \bfseries\entryname&\bfseries\descriptionname&

```



```

9242 \bfseries\pagelistname\tabularnewline\hline\endhead
9243 \hline\endfoot}%
9244 }

```

`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.

```

9245 \newglossarystyle{altlongragged4col}{%

```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```

9246 \renewenvironment{theglossary}%
9247 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}l%
9248 >{\raggedright}p{\glspagelistwidth}}}%
9249 {\end{longtable}}%

```

No table header:

```

9250 \renewcommand*{\glossaryheader}{}%

```

No group headings:

```

9251 \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):

```

9252 \renewcommand{\glossentry}[2]{%
9253 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9254 \glossentrydesc{##1} & \glossentrysymbol{##1} &
9255 ##2\tabularnewline
9256 }%

```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```

9257 \renewcommand{\subglossentry}[3]{%
9258 &
9259 \glssubentryitem{##2}%
9260 \glstarget{##2}{\strut}\glossentrydesc{##2} &
9261 \glossentrysymbol{##2} & ##3\tabularnewline
9262 }%

```

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>)

```

9263 \ifglsgroupskip
9264 \renewcommand*{\glsgroupskip}{}%
9265 \else
9266 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
9267 \fi
9268 }

```

`altlongragged4colheader` The `altlongragged4colheader` style is like `altlongragged4col` but with a header row.

```

9269 \newglossarystyle{altlongragged4colheader}{%

```

Base it on the `glostylealtlongragged4col` style:

```

9270 \setglossarystyle{altlongragged4col}%

```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9271 \renewenvironment{theglossary}%
9272   {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}l%
9273     >{\raggedright}p{\glspagelistwidth}}}%
9274   {\end{longtable}}%
```

Table has a header:

```
9275 \renewcommand*{\glossaryheader}{%
9276   \bfseries\entryname&\bfseries\descriptionname&
9277   \bfseries \symbolname&
9278   \bfseries\pagelistname\tabularnewline\endhead}%
9279 }
```

**ragged4colborder** The altlongragged4colborder style is like altlongragged4col but with a border.

```
9280 \newglossarystyle{altlongragged4colborder}{%
```

Base it on the glostylealtlongragged4col style:

```
9281 \setglossarystyle{altlongragged4col}%
```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9282 \renewenvironment{theglossary}%
9283   {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|l|}%
9284     >{\raggedright}p{\glspagelistwidth}|}%
9285   {\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```
9286 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9287 }
```

**colheaderborder** The altlongragged4colheaderborder style is like the above but with a header as well as a border.

```
9288 \newglossarystyle{altlongragged4colheaderborder}{%
```

Base it on the glostylealtlongragged4col style:

```
9289 \setglossarystyle{altlongragged4col}%
```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9290 \renewenvironment{theglossary}%
9291   {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|l|}%
9292     >{\raggedright}p{\glspagelistwidth}|}%
9293   {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
9294 \renewcommand*{\glossaryheader}{%
9295   \hline\bfseries\entryname&\bfseries\descriptionname&
9296   \bfseries \symbolname&
9297   \bfseries\pagelistname\tabularnewline\hline\endhead
9298   \hline\endfoot}%
9299 }
```

## 3.8 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.

9300 \ProvidesPackage{glossary-mcols}[2021/11/01 v4.49 (NLCT)]

Required packages:

9301 \RequirePackage{multicol}

9302 \RequirePackage{glossary-tree}

\indexspace The are a few classes that don't define \indexspace, so provide a definition if it hasn't been defined.

9303 \providecommand{\indexspace}{%

9304 \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax

9305 }

\glsmcols Define macro in which to store the number of columns. (Defaults to 2.)

9306 \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 multicol, but the title isn't part of the glossary style.)

9307 \newglossarystyle{mcolindex}{%

9308 \setglossarystyle{index}%

9309 \renewenvironment{theglossary}%

9310 {%

9311 \begin{multicols}{\glsmcols}

9312 \setlength{\parindent}{0pt}%

9313 \setlength{\parskip}{0pt plus 0.3pt}%

9314 \let\item\glstreeitem

9315 \let\subitem\glstreesubitem

9316 \let\subsubitem\glstreesubsubitem

9317 }%

9318 {\end{multicols}}%

9319 }

mcolindexgroup As mcolindex but has headings:

9320 \newglossarystyle{mcolindexgroup}{%

9321 \setglossarystyle{mcolindex}%

9322 \renewcommand\*{\glsgroupheading}[1]{%

9323 \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%

9324 }

indexhypergroup The mcolindexhypergroup style is like the mcolindexgroup style but has hyper navigation.

9325 \newglossarystyle{mcolindexhypergroup}{%

Base it on the `glostylemcolindex` style:

```
9326 \setglossarystyle{mcolindex}%
```

Put navigation links to the groups at the start of the glossary:

```
9327 \renewcommand*{\glossaryheader}{%
```

```
9328   \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.

```
9329 \renewcommand*{\glsgroupheading}[1]{%
```

```
9330   \item\glstreegroupheaderfmt
```

```
9331     {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
```

```
9332   \indexspace}%
```

```
9333 }
```

`colindexspannav` Similar to `mcolindexhypergroup`, but puts the navigation line in the optional argument of `multicols`.

```
9334 \newglossarystyle{mcolindexspannav}{%
```

```
9335   \setglossarystyle{index}%
```

```
9336   \renewenvironment{theglossary}%
```

```
9337     {%
```

```
9338       \begin{multicols}{\glsmcols}\noindent\glstreenavigationfmt{\glsnavigation}]
```

```
9339       \setlength{\parindent}{0pt}%
```

```
9340       \setlength{\parskip}{0pt plus 0.3pt}%
```

```
9341       \let\item\glstreeitem}%
```

```
9342     {\end{multicols}}}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
9343 \renewcommand*{\glsgroupheading}[1]{%
```

```
9344   \item\glstreegroupheaderfmt
```

```
9345     {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
```

```
9346   \indexspace}%
```

```
9347 }
```

`mcoltree` Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
9348 \newglossarystyle{mcoltree}{%
```

```
9349   \setglossarystyle{tree}%
```

```
9350   \renewenvironment{theglossary}%
```

```
9351     {%
```

```
9352       \begin{multicols}{\glsmcols}
```

```
9353       \setlength{\parindent}{0pt}%
```

```
9354       \setlength{\parskip}{0pt plus 0.3pt}%
```

```
9355     }%
```

```
9356     {\end{multicols}}}%
```

```
9357 }
```

**mcoltreegroup** Like the **mcoltree** style but the glossary groups have headings.

```

9358 \newglossarystyle{mcoltreegroup}{%
    Base it on the glostylemcoltree style:
9359   \setglossarystyle{mcoltree}%

    Each group has a heading (in bold) followed by a vertical gap):
9360   \renewcommand{\glsgroupheading}[1]{\par
9361     \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
9362 }
```

**ltreehypergroup** The **mcoltreehypergroup** style is like the **treegroup** style, but has a set of links to the groups at the start of the glossary.

```

9363 \newglossarystyle{mcoltreehypergroup}{%
    Base it on the glostylemcoltree style:
9364   \setglossarystyle{mcoltree}%

    Put navigation links to the groups at the start of the theglossary environment:
9365   \renewcommand*{\glossaryheader}{%
9366     \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

    Each group has a heading (in bold with a target) followed by a vertical gap):
9367   \renewcommand*{\glsgroupheading}[1]{%
9368     \par\noindent
9369     \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9370     \indexspace}%
9371 }
```

**mcoltreespannav** Similar to the **mcoltreehypergroup** style but the navigation line is put in the optional argument of the **multicols** environment.

```

9372 \newglossarystyle{mcoltreespannav}{%
9373   \setglossarystyle{tree}%
9374   \renewenvironment{theglossary}%
9375   {%

9376     \begin{multicols}{\glsmcols}\noindent\glstreenavigationfmt{\glsnavigation}]
9377     \setlength{\parindent}{0pt}%
9378     \setlength{\parskip}{0pt plus 0.3pt}%
9379   }%
9380   {\end{multicols}}%

    Each group has a heading (in bold with a target) followed by a vertical gap):
9381   \renewcommand*{\glsgroupheading}[1]{%
9382     \par\noindent
9383     \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9384     \indexspace}%
9385 }
```

`mcoltreenoname` Multi-column index style. Same as the `treenoname`, but puts the glossary in multiple columns.

```

9386 \newglossarystyle{mcoltreenoname}{%
9387   \setglossarystyle{treenoname}%
9388   \renewenvironment{theglossary}%
9389   {%
9390     \begin{multicols}{\glsmcols}
9391     \setlength{\parindent}{0pt}%
9392     \setlength{\parskip}{0pt plus 0.3pt}%
9393   }%
9394   {\end{multicols}}%
9395 }
```

`treenonamegroup` Like the `mcoltreenoname` style but the glossary groups have headings.

```

9396 \newglossarystyle{mcoltreenonamegroup}{%
    Base it on the glostylemcoltreenoname style:
9397   \setglossarystyle{mcoltreenoname}%
    Give each group a heading:
9398   \renewcommand{\glsgroupheading}[1]{\par
9399     \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
9400 }
```

`onamehypergroup` The `mcoltreenonamehypergroup` style is like the `mcoltreenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```

9401 \newglossarystyle{mcoltreenonamehypergroup}{%
    Base it on the glostylemcoltreenoname style:
9402   \setglossarystyle{mcoltreenoname}%
    Put navigation links to the groups at the start of the theglossary environment:
9403   \renewcommand*{\glossaryheader}{%
9404     \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    Each group has a heading (in bold with a target) followed by a vertical gap):
9405   \renewcommand*{\glsgroupheading}[1]{%
9406     \par\noindent
9407     \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
9408     \indexspace}%
9409 }
```

`eenonamespannav` Similar to the `mcoltreenonamehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

9410 \newglossarystyle{mcoltreenonamespannav}{%
9411   \setglossarystyle{treenoname}%
9412   \renewenvironment{theglossary}%
9413   {%
9414     \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
9415     \setlength{\parindent}{0pt}%
9416     \setlength{\parskip}{0pt plus 0.3pt}%

```

```

9417 }%
9418 {\end{multicols}}%

Each group has a heading (in bold with a target) followed by a vertical gap):
9419 \renewcommand*{\glsgroupheading}[1]{%
9420 \par\noindent
9421 \glstreegroupheaderfmt{\glssnavhypertarget{##1}{\glsgsetgrouptitle{##1}}}\par
9422 \indexspace}%
9423 }

```

**mcolalmtree** Multi-column index style. Same as the almtree, but puts the glossary in multiple columns.

```

9424 \newglossarystyle{mcolalmtree}{%
9425 \setglossarystyle{almtree}%
9426 \renewenvironment{theglossary}%
9427 {%

9428 \begin{multicols}{\glsmcols}
9429 \def\@gls@prevlevel{-1}%
9430 \mbox{}\par
9431 }%
9432 {\par\end{multicols}}%
9433 }

```

**colalmtreegroup** Like the mcolalmtree style but the glossary groups have headings.

```

9434 \newglossarystyle{mcolalmtreegroup}{%

Base it on the glostylemcolalmtree style:
9435 \setglossarystyle{mcolalmtree}%

Give each group a heading.
9436 \renewcommand{\glsgroupheading}[1]{\par
9437 \def\@gls@prevlevel{-1}%
9438 \hangindent0pt\relax
9439 \parindent0pt\relax
9440 \glstreegroupheaderfmt{\glsgsetgrouptitle{##1}}\par\indexspace}%
9441 }

```

**almtreehypergroup** The mcolalmtreehypergroup style is like the mcolalmtreegroup style, but has a set of links to the groups at the start of the glossary.

```

9442 \newglossarystyle{mcolalmtreehypergroup}{%

Base it on the glostylemcolalmtree style:
9443 \setglossarystyle{mcolalmtree}%

Put the navigation links in the header
9444 \renewcommand*{\glossaryheader}{%
9445 \par
9446 \def\@gls@prevlevel{-1}%
9447 \hangindent0pt\relax
9448 \parindent0pt\relax
9449 \glstreenavigationfmt{\glssnavigation}\par\indexspace}%

```

Put a hypertarget at the start of each group

```

9450 \renewcommand*{\glsgroupheading}[1]{%
9451 \par
9452 \def\@gls@prevlevel{-1}%
9453 \hangindent0pt\relax
9454 \parindent0pt\relax
9455 \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
9456 \indexspace}%
9457 }

```

`\alttreespannav` Similar to the `\mcolalttreehypergroup` style but the navigation line is put in the optional argument of the `\multicols` environment.

```

9458 \newglossarystyle{mcolalttreespannav}{%
9459 \setglossarystyle{alttree}%
9460 \renewenvironment{theglossary}%
9461 {%
9462 \begin{multicols}{\glsmlcols}\noindent\glstreenavigationfmt{\glsnavigation}]
9463 \def\@gls@prevlevel{-1}%
9464 \mbox{} \par
9465 }%
9466 {\par\end{multicols}}}%

```

Put a hypertarget at the start of each group

```

9467 \renewcommand*{\glsgroupheading}[1]{%
9468 \par
9469 \def\@gls@prevlevel{-1}%
9470 \hangindent0pt\relax
9471 \parindent0pt\relax
9472 \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
9473 \indexspace}%
9474 }

```

### 3.9 Glossary Styles using supertabular environment (glossary-super package)

The glossary styles defined in the package use the `supertabular` environment.

```

9475 \ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]

```

Requires the package:

```

9476 \RequirePackage{supertabular}

```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined if has been loaded.

```

9477 \@ifundefined{glsdescwidth}{%
9478 \newlength\glsdescwidth
9479 \setlength{\glsdescwidth}{0.6\hsize}
9480 }{}

```



`glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if has been loaded.

```

9481 \@ifundefined{glspagelistwidth}{%
9482   \newlength{glspagelistwidth
9483   \setlength{glspagelistwidth}{0.1\hsize}
9484 }{}

```

`super` The super glossary style uses the supertabular environment (it uses lengths defined in the package.)

```

9485 \newglossarystyle{super}{%

```

Put the glossary in a supertabular environment with two columns and no head or tail:

```

9486 \renewenvironment{theglossary}%
9487   {\tablehead{}\tabletail{}}%
9488   \begin{supertabular}{lp{glsglwidth}}%
9489   {\end{supertabular}}%

```

Do nothing at the start of the table:

```

9490 \renewcommand*{glossaryheader}{}%

```

No group headings:

```

9491 \renewcommand*{glsgroupheading}[1]{}%

```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```

9492 \renewcommand{glossentry}[2]{%
9493   \glstarget{##1}{\glossentryname{##1}} &
9494   \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
9495 }%

```

Sub entries put in a row (no name, description and page list in second column):

```

9496 \renewcommand{subglossentry}[3]{%
9497   &
9498   \glssubentryitem{##2}%
9499   \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
9500   ##3\tabularnewline
9501 }%

```

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>)

```

9502 \ifglsgroupskip
9503   \renewcommand*{glsgroupskip}{}%
9504 \else
9505   \renewcommand*{glsgroupskip}{& \tabularnewline}%
9506 \fi
9507 }

```

`superborder` The superborder style is like the above, but with horizontal and vertical lines:

```

9508 \newglossarystyle{superborder}{%

```

Base it on the `glostylessuper` style:

```

9509 \setglossarystyle{super}%

```

Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:

```

9510 \renewenvironment{theglossary}%
9511   {\tablehead{\hline}\tabletail{\hline}%
9512    \begin{supertabular}{|l|p{\glsdescwidth}|}%
9513    {\end{supertabular}}%
9514 }
```

**superheader** The superheader style is like the super style, but with a header:

```

9515 \newglossarystyle{superheader}{%
```

Base it on the glostylessuper style:

```

9516 \setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and no tail:

```

9517 \renewenvironment{theglossary}%
9518   {\tablehead{\bfseries \entryname &
9519    \bfseries\descriptionname\tabularnewline}%
9520    \tabletail{}}%
9521    \begin{supertabular}{lp{\glsdescwidth}}%
9522    {\end{supertabular}}%
9523 }
```

**superheaderborder** The superheaderborder style is like the super style but with a header and border:

```

9524 \newglossarystyle{superheaderborder}{%
```

Base it on the glostylessuper style:

```

9525 \setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and horizontal lines above and below the table:

```

9526 \renewenvironment{theglossary}%
9527   {\tablehead{\hline\bfseries \entryname &
9528    \bfseries \descriptionname\tabularnewline\hline}%
9529    \tabletail{\hline}
9530    \begin{supertabular}{|l|p{\glsdescwidth}|}%
9531    {\end{supertabular}}%
9532 }
```

**super3col** The super3col style is like the super style, but with 3 columns:

```

9533 \newglossarystyle{super3col}{%
```

Put the glossary in a supertabular environment with three columns and no head or tail:

```

9534 \renewenvironment{theglossary}%
9535   {\tablehead{}\tabletail{}}%
9536   \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
9537   {\end{supertabular}}%
```

Do nothing at the start of the table:

```

9538 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9539 \renewcommand*{\glsgrouphheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
9540 \renewcommand{\glossentry}[2]{%
9541   \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9542   \glossentrydesc{##1} & ##2\tabularnewline
9543 }%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
9544 \renewcommand{\subglossentry}[3]{%
9545   &
9546   \glssubentryitem{##2}%
9547   \glstarget{##2}{\strut}\glossentrydesc{##2} &
9548   ##3\tabularnewline
9549 }%
```

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>)

```
9550 \ifglsgroupskip
9551 \renewcommand*{\glsgroupskip}{}%
9552 \else
9553 \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9554 \fi
9555 }
```

super3colborder The super3colborder style is like the super3col style, but with a border:

```
9556 \newglossarystyle{super3colborder}{%
```

Base it on the glostylesuper3col style:

```
9557 \setglossarystyle{super3col}%
```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```
9558 \renewenvironment{theglossary}%
9559   {\tablehead{\hline}\tabletail{\hline}%
9560   \begin{supertabular}{|l|p{\glsglwdwidth}|p{\glspagelstwidth|}}%
9561   {\end{supertabular}}%
9562 }
```

super3colheader The super3colheader style is like the super3col style but with a header row:

```
9563 \newglossarystyle{super3colheader}{%
```

Base it on the glostylesuper3col style:

```
9564 \setglossarystyle{super3col}%
```

Put the glossary in a supertabular environment with three columns, a header and no tail:

```
9565 \renewenvironment{theglossary}%
9566   {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9567     \bfseries\pagelistname\tabularnewline}\tabletail{}}%
```

```

9568     \begin{supertabular}{lp{\glstdescwidth}p{\glspagelistwidth}}}%
9569     {\end{supertabular}}}%
9570 }

```

colheaderborder The super3colheaderborder style is like the super3col style but with a header and border:

```

9571 \newglossarystyle{super3colheaderborder}{%
    Base it on the glostylesuper3colborder style:
9572 \setglossarystyle{super3colborder}%
    Put the glossary in a supertabular environment with three columns, a header with horizontal
    lines and a horizontal line in the tail:
9573 \renewenvironment{theglossary}%
9574     {\tablehead{\hline
9575         \bfseries\entryname&\bfseries\descriptionname&
9576         \bfseries\pagelistname\tabularnewline\hline}%
9577     \tabletail{\hline}%
9578     \begin{supertabular}{|lp{\glstdescwidth}|p{\glspagelistwidth}|}%
9579     {\end{supertabular}}}%
9580 }

```

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.

```

9581 \newglossarystyle{super4col}{%
    Put the glossary in a supertabular environment with four columns and no head or tail:
9582 \renewenvironment{theglossary}%
9583     {\tablehead{}\tabletail{}}%
9584     \begin{supertabular}{|llll|}%
9585     \end{supertabular}}%
    Do nothing at the start of the table:
9586 \renewcommand*{\glossaryheader}{}%
    No group headings:
9587 \renewcommand*{\glsgroupheading}[1]{}%
    Main (level 0) entries on a row with the name in the first column, description in second col-
    umn, symbol in third column and page list in last column:
9588 \renewcommand{\glossentry}[2]{%
9589     \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9590     \glossentrydesc{##1} &
9591     \glossentrysymbol{##1} & ##2\tabularnewline
9592 }%
    Sub entries on a row with no name, the description in the second column, symbol in third
    column and page list in last column:
9593 \renewcommand{\subglossentry}[3]{%
9594     &
9595     \glssubentryitem{##2}%
9596     \glstarget{##2}{\strut}\glossentrydesc{##2} &

```

```

9597 \glossentrysymbol{##2} & ##3\tabularnewline
9598 }%

```

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>)

```

9599 \ifglsgroupskip
9600 \renewcommand*{\glsgroupskip}{}%
9601 \else
9602 \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9603 \fi
9604 }

```

**super4colheader** The super4colheader style is like the super4col but with a header row.

```
9605 \newglossarystyle{super4colheader}{%
```

Base it on the glostylessuper4col style:

```
9606 \setglossarystyle{super4col}%
```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```

9607 \renewenvironment{theglossary}%
9608 {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9609 \bfseries\symbolname &
9610 \bfseries\pagelistname\tabularnewline}%
9611 \tabletail{}}%
9612 \begin{supertabular}{l1l1l}%
9613 {\end{supertabular}}%
9614 }

```

**super4colborder** The super4colborder style is like the super4col but with a border.

```
9615 \newglossarystyle{super4colborder}{%
```

Base it on the glostylessuper4col style:

```
9616 \setglossarystyle{super4col}%
```

Put the glossary in a supertabular environment with four columns and a horizontal line in the head and tail:

```

9617 \renewenvironment{theglossary}%
9618 {\tablehead{\hline}\tabletail{\hline}%
9619 \begin{supertabular}{l1l1l1l1}%
9620 {\end{supertabular}}%
9621 }

```

**colheaderborder** The super4colheaderborder style is like the super4col but with a header and border.

```
9622 \newglossarystyle{super4colheaderborder}{%
```

Base it on the glostylessuper4col style:

```
9623 \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:

```
9624 \renewenvironment{theglossary}%
```

```

9625     {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
9626         \bfseries\symbolname &
9627         \bfseries\pagelistname\tabularnewline\hline}%
9628     \tabletail{\hline}%
9629     \begin{supertabular}{|l|l|l|l|}%
9630     {\end{supertabular}}%
9631 }

```

**altsuper4col** The altsuper4col glossary style is like super4col but has provision for multiline descriptions.

```

9632 \newglossarystyle{altsuper4col}{%

```

Base it on the glostylessuper4col style:

```

9633 \setglossarystyle{super4col}%

```

Put the glossary in a supertabular environment with four columns and no head or tail:

```

9634 \renewenvironment{theglossary}%
9635     {\tablehead{}\tabletail{}}%
9636     \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}%
9637     {\end{supertabular}}%
9638 }

```

**super4colheader** The altsuper4colheader style is like the altsuper4col but with a header row.

```

9639 \newglossarystyle{altsuper4colheader}{%

```

Base it on the glostylessuper4colheader style:

```

9640 \setglossarystyle{super4colheader}%

```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```

9641 \renewenvironment{theglossary}%
9642     {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9643         \bfseries\symbolname &
9644         \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9645     \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}%
9646     {\end{supertabular}}%
9647 }

```

**super4colborder** The altsuper4colborder style is like the altsuper4col but with a border.

```

9648 \newglossarystyle{altsuper4colborder}{%

```

Base it on the glostylessuper4colborder style:

```

9649 \setglossarystyle{super4colborder}%

```

Put the glossary in a supertabular environment with four columns and a horizontal line in the head and tail:

```

9650 \renewenvironment{theglossary}%
9651     {\tablehead{\hline}\tabletail{\hline}%
9652     \begin{supertabular}%
9653         {|l|p{\glsgdescwidth}|l|p{\glspagelistwidth}|}%
9654     {\end{supertabular}}%
9655 }

```

`colheaderborder` The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```

9656 \newglossarystyle{altsuper4colheaderborder}{%
      Base it on the glostylessuper4colheaderborder style:
9657   \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:
9658   \renewenvironment{theglossary}%
9659     {\tablehead{\hline
9660       \bfseries\entryname &
9661       \bfseries\descriptionname &
9662       \bfseries\symbolname &
9663       \bfseries\pagelistname\tabularnewline\hline}%
9664     \tabletail{\hline}%
9665     \begin{supertabular}%
9666       {1|p{\glsdescwidth}|1|p{\glspagelistwidth}|}%
9667     {\end{supertabular}}%
9668 }
```

### 3.10 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.

```

9669 \ProvidesPackage{glossary-superragged}[2021/11/01 v4.49 (NLCT)]
```

Requires the package:

```

9670 \RequirePackage{array}
```

Requires the package:

```

9671 \RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```

9672 \@ifundefined{glsdescwidth}{%
9673   \newlength\glsdescwidth
9674   \setlength{\glsdescwidth}{0.6\hsize}
9675 }{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```

9676 \@ifundefined{glspagelistwidth}{%
9677   \newlength\glspagelistwidth
9678   \setlength{\glspagelistwidth}{0.1\hsize}
9679 }{}
```

superragged The superragged glossary style uses the supertabular environment.

```
9680 \newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
9681 \renewenvironment{theglossary}%  
9682 {\tablehead{}\tabletail{}}%  
9683 \begin{supertabular}{1>{\raggedright}p{\glsgdescwidth}}}%  
9684 {\end{supertabular}}%
```

Do nothing at the start of the table:

```
9685 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9686 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
9687 \renewcommand{\glossentry}[2]{%  
9688 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
9689 \glossentrydesc{##1}\glspostdescription\space ##2%  
9690 \tabularnewline  
9691 }%
```

Sub entries put in a row (no name, description and page list in second column):

```
9692 \renewcommand{\subglossentry}[3]{%  
9693 &  
9694 \glssubentryitem{##2}%  
9695 \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space  
9696 ##3%  
9697 \tabularnewline  
9698 }%
```

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>)

```
9699 \ifglsgroupskip  
9700 \renewcommand*{\glsgroupskip}{}%  
9701 \else  
9702 \renewcommand*{\glsgroupskip}{& \tabularnewline}%  
9703 \fi  
9704 }
```

perraggedborder The superraggedborder style is like the above, but with horizontal and vertical lines:

```
9705 \newglossarystyle{superraggedborder}{%
```

Base it on the `glostylesuperragged` style:

```
9706 \setglossarystyle{superragged}%
```

Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:

```
9707 \renewenvironment{theglossary}%  
9708 {\tablehead{\hline}\tabletail{\hline}%  
9709 \begin{supertabular}{1|1>{\raggedright}p{\glsgdescwidth}|}}%
```



```

9710     {\end{supertabular}}}%
9711 }

```

**superraggedheader** The superraggedheader style is like the super style, but with a header:

```

9712 \newglossarystyle{superraggedheader}{%
    Base it on the glostylesuperragged style:
9713   \setglossarystyle{superragged}%
    Put the glossary in a supertabular environment with two columns, a header and no tail:
9714 \renewenvironment{theglossary}%
9715   {\tablehead{\bfseries \entryname & \bfseries \descriptionname
9716     \tabularnewline}%
9717   \tabletail{}}%
9718   \begin{supertabular}{l>{\raggedright}p{\glstdescwidth}}}%
9719   {\end{supertabular}}}%
9720 }

```

**superraggedheaderborder** The superraggedheaderborder style is like the superragged style but with a header and border:

```

9721 \newglossarystyle{superraggedheaderborder}{%
    Base it on the glostylesuper style:
9722   \setglossarystyle{superragged}%
    Put the glossary in a supertabular environment with two columns, a header and horizontal
    lines above and below the table:
9723 \renewenvironment{theglossary}%
9724   {\tablehead{\hline\bfseries \entryname &
9725     \bfseries \descriptionname\tabularnewline\hline}%
9726   \tabletail{\hline}%
9727   \begin{supertabular}{ll>{\raggedright}p{\glstdescwidth}l}}}%
9728   {\end{supertabular}}}%
9729 }

```

**superragged3col** The superragged3col style is like the superragged style, but with 3 columns:

```

9730 \newglossarystyle{superragged3col}{%
    Put the glossary in a supertabular environment with three columns and no head or tail:
9731 \renewenvironment{theglossary}%
9732   {\tablehead{}\tabletail{}}%
9733   \begin{supertabular}{l>{\raggedright}p{\glstdescwidth}%
9734     >{\raggedright}p{\glspagelistwidth}}}%
9735   {\end{supertabular}}}%
    Do nothing at the start of the table:
9736 \renewcommand*{\glossaryheader}{}%
    No group headings:
9737 \renewcommand*{\glsgroupheading}[1]{}%

```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```

9738 \renewcommand{\glossentry}[2]{%
9739 \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9740 \glossentrydesc{##1} &
9741 ##2\tabularnewline
9742 }%

```

Sub entries on a row (no name, description in second column, page list in last column):

```

9743 \renewcommand{\subglossentry}[3]{%
9744 &
9745 \glssubentryitem{##2}%
9746 \glstarget{##2}{\strut}\glossentrydesc{##2} &
9747 ##3\tabularnewline
9748 }%

```

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>)

```

9749 \ifglsgroupskip
9750 \renewcommand*{\glsgroupskip}{}%
9751 \else
9752 \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9753 \fi
9754 }

```

ragged3colborder The superragged3colborder style is like the superragged3col style, but with a border:

```

9755 \newglossarystyle{superragged3colborder}{%

```

Base it on the glostylesuperragged3col style:

```

9756 \setglossarystyle{superragged3col}%

```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```

9757 \renewenvironment{theglossary}%
9758 {\tablehead{\hline}\tabletail{\hline}%
9759 \begin{supertabular}{|l|>{\raggedright}p{\glsgdescwidth}|}%
9760 >{\raggedright}p{\glspagelistwidth}|}%
9761 {\end{supertabular}}%
9762 }

```

ragged3colheader The superragged3colheader style is like the superragged3col style but with a header row:

```

9763 \newglossarystyle{superragged3colheader}{%

```

Base it on the glostylesuperragged3col style:

```

9764 \setglossarystyle{superragged3col}%

```

Put the glossary in a supertabular environment with three columns, a header and no tail:

```

9765 \renewenvironment{theglossary}%
9766 {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9767 \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9768 \begin{supertabular}{l>{\raggedright}p{\glsgdescwidth}%

```

```

9769         >\raggedright}p{\glspagelistwidth}}}%
9770     {\end{supertabular}}}%
9771 }

```

**colheaderborder** The `superragged3colheaderborder` style is like the `superragged3col` style but with a header and border:

```

9772 \newglossarystyle{superragged3colheaderborder}{%
    Base it on the glostypesuperragged3colborder style:
9773     \setglossarystyle{superragged3colborder}%
    Put the glossary in a supertabular environment with three columns, a header with horizontal
    lines and a horizontal line in the tail:
9774     \renewenvironment{theglossary}%
9775     {\tablehead{\hline
9776         \bfseries\entryname&\bfseries\descriptionname&
9777         \bfseries\pagelistname\tabularnewline\hline}%
9778     \tabletail{\hline}%
9779     \begin{supertabular}{|l|>\raggedright}p{\glsdescwidth}|%
9780         >\raggedright}p{\glspagelistwidth}|}%
9781     {\end{supertabular}}}%
9782 }

```

**superragged4col** The `altsuperragged4col` glossary style is like `altsuper4col` style in the package but uses ragged right formatting in the description and page list columns.

```

9783 \newglossarystyle{altsuperragged4col}{%
    Put the glossary in a supertabular environment with four columns and no head or tail:
9784     \renewenvironment{theglossary}%
9785     {\tablehead{}\tabletail{}%
9786         \begin{supertabular}{|l>\raggedright}p{\glsdescwidth}l%
9787         >\raggedright}p{\glspagelistwidth}}}%
9788     {\end{supertabular}}}%
    Do nothing at the start of the table:
9789     \renewcommand*{\glossaryheader}{}%
    No group headings:
9790     \renewcommand*{\glsgroupheading}[1]{}%
    Main (level 0) entries on a row with the name in the first column, description in second col-
    umn, symbol in third column and page list in last column:
9791     \renewcommand{\glossentry}[2]{%
9792         \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9793         \glossentrydesc{##1} &
9794         \glossentrysymbol{##1} & ##2\tabularnewline
9795     }%
    Sub entries on a row with no name, the description in the second column, symbol in third
    column and page list in last column:
9796     \renewcommand{\subglossentry}[3]{%

```

```

9797      &
9798      \glssubentryitem{##2}%
9799      \glstarget{##2}{\strut}\glossentrydesc{##2} &
9800      \glossentrysymbol{##2} & ##3\tabularnewline
9801  }%

```

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>)

```

9802  \ifglsnogroupskip
9803    \renewcommand*{\glsgroupskip}{}%
9804  \else
9805    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9806  \fi
9807 }

```

ragged4colheader The altsuperragged4colheader style is like the altsuperragged4col style but with a header row.

```

9808 \newglossarystyle{altsuperragged4colheader}{%

```

Base it on the glostylealtsuperragged4col style:

```

9809   \setglossarystyle{altsuperragged4col}%

```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```

9810   \renewenvironment{theglossary}%
9811   {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9812     \bfseries\symbolname &
9813     \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9814   \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
9815     >{\raggedright}p{\glspagelistwidth}}}%
9816   {\end{supertabular}}%
9817 }

```

ragged4colborder The altsuperragged4colborder style is like the altsuperragged4col style but with a border.

```

9818 \newglossarystyle{altsuperragged4colborder}{%

```

Base it on the glostylealtsuperragged4col style:

```

9819   \setglossarystyle{altsuper4col}%

```

Put the glossary in a supertabular environment with four columns and a horizontal line in the head and tail:

```

9820   \renewenvironment{theglossary}%
9821   {\tablehead{\hline}\tabletail{\hline}%
9822   \begin{supertabular}%
9823     {l|>{\raggedright}p{\glsdescwidth}|l|}%
9824     >{\raggedright}p{\glspagelistwidth}|}}%
9825   {\end{supertabular}}%
9826 }

```

colheaderborder The altsuperragged4colheaderborder style is like the altsuperragged4col style but with a header and border.

```

9827 \newglossarystyle{altsuperragged4colheaderborder}{%

```

Base it on the `glostylealtsuperragged4col` style:

```
9828 \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:

```
9829 \renewenvironment{theglossary}%
9830   {\tablehead{\hline
9831     \bfseries\entryname &
9832     \bfseries\descriptionname &
9833     \bfseries\symbolname &
9834     \bfseries\pagelistname\tabularnewline\hline}%
9835   \tabletail{\hline}%
9836   \begin{supertabular}%
9837     {||>\raggedright}p{\glstdescwidth}|||%
9838     >\raggedright}p{\glspagelistwidth}||}%
9839   {\end{supertabular}}%
9840 }
```

### 3.11 Tree Styles (`glossary-tree.sty`)

The style file defines glossary styles that have a tree-like structure. These are designed for hierarchical glossaries.

```
9841 \ProvidesPackage{glossary-tree}[2021/11/01 v4.49 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
9842 \providecommand{\indexspace}{%
9843   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
9844 }
```

`\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.

```
9845 \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.

```
9846 \newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```
9847 \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.

```

9848 \ifdef\@idxitem
9849 {\newcommand{\glstreeitem}{\@idxitem}}
9850 {\newcommand{\glstreeitem}{\par\hangindent40\p@}}

```

`\glstreesubitem` Level 1 item used in index style.

```

9851 \ifdef\subitem
9852 {\let\glstreesubitem\subitem}
9853 {\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}

```

`\glstreesubsubitem` Level 1 item used in index style.

```

9854 \ifdef\subsubitem
9855 {\let\glstreesubsubitem\subsubitem}
9856 {\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}

```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `almtree` style).

```

9857 \newcommand{\glstreepredesc}{\space}

```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treename` and `almtree` style).

```

9858 \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.

```

9859 \newglossarystyle{index}{%

```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```

9860 \renewenvironment{theglossary}%
9861 {\setlength{\parindent}{0pt}%
9862 \setlength{\parskip}{0pt plus 0.3pt}%
9863 \let\item\glstreeitem
9864 \let\subitem\glstreesubitem
9865 \let\subsubitem\glstreesubsubitem
9866 }%

```

```

9867 {\par}%

```

Do nothing at the start of the environment:

```

9868 \renewcommand*{\glossaryheader}{}%

```

No group headers:

```

9869 \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.

```

9870 \renewcommand*{\glossentry}[2]{%

```

```

9871 \item\glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9872 \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9873 \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
9874 }%

```

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.

```

9875 \renewcommand{\subglossentry}[3]{%
9876 \ifcase##1\relax
9877 % level 0
9878 \item
9879 \or
9880 % level 1
9881 \subitem
9882 \glssubentryitem{##2}%
9883 \else
9884 % all other levels
9885 \subsubitem
9886 \fi
9887 \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
9888 \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
9889 \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
9890 }%

```

Vertical gap between groups is the same as that used by indices:

```

9891 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}

```

**indexgroup** The `indexgroup` style is like the `index` style but has headings.

```

9892 \newglossarystyle{indexgroup}{%

```

Base it on the `glostyleindex` style:

```

9893 \setglossarystyle{index}%

```

Add a heading for each group. This puts the group's title in bold followed by a vertical gap.

```

9894 \renewcommand*{\glsgroupheading}[1]{%
9895 \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
9896 \indexspace
9897 }%
9898 }

```

**indexhypergroup** The `indexhypergroup` style is like the `indexgroup` style but has hyper navigation.

```

9899 \newglossarystyle{indexhypergroup}{%

```

Base it on the `glostyleindex` style:

```

9900 \setglossarystyle{index}%

```

Put navigation links to the groups at the start of the glossary:

```

9901 \renewcommand*{\glossaryheader}{%
9902 \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.

```

9903 \renewcommand*\glsgroupheading}[1]{%
9904   \item\glstreegroupheaderfmt
9905     {\glshnavhypertarget{##1}{\glsgrouptitle{##1}}}%
9906   \indexspace}%
9907 }
```

**tree** The tree glossary style is similar in style to the index style, but can have arbitrary levels.

```

9908 \newglossarystyle{tree}{%
```

Set the paragraph indentation and skip:

```

9909 \renewenvironment{theglossary}%
9910   {\setlength{\parindent}{0pt}%
9911    \setlength{\parskip}{0pt plus 0.3pt}}%
9912   {}%
```

Do nothing at the start of the theglossary environment:

```

9913 \renewcommand*\glossaryheader{}%
```

No group headings:

```

9914 \renewcommand*\glsgroupheading}[1]{}%
```

Main (level 0) entries: name in bold, followed by symbol in brackets (if it exists), the description and the page list:

```

9915 \renewcommand{\glossentry}[2]{%
9916   \hangindent0pt\relax
9917   \parindent0pt\relax
9918   \glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9919   \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9920   \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
9921   }%
```

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.

```

9922 \renewcommand{\subglossentry}[3]{%
9923   \hangindent##1\glstreeindent\relax
9924   \parindent##1\glstreeindent\relax
9925   \ifnum##1=1\relax
9926     \glssubentryitem{##2}%
9927     \fi
9928     \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
9929     \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
9930     \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
9931   }%
```

Vertical gap between groups is the same as that used by indices:

```

9932 \renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}
```

**treegroup** Like the tree style but the glossary groups have headings.

```

9933 \newglossarystyle{treegroup}{%
```



Base it on the glostyletree style:

```
9934 \setglossarystyle{tree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
9935 \renewcommand{\glsgroupheading}[1]{\par
9936 \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
9937 \indexspace}%
9938 }
```

**treehypergroup** The treehypergroup style is like the treegroup style, but has a set of links to the groups at the start of the glossary.

```
9939 \newglossarystyle{treehypergroup}{%
```

Base it on the glostyletree style:

```
9940 \setglossarystyle{tree}%
```

Put navigation links to the groups at the start of the theglossary environment:

```
9941 \renewcommand*{\glossaryheader}{%
9942 \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
9943 \renewcommand*{\glsgroupheading}[1]{%
9944 \par\noindent
9945 \glstreegroupheaderfmt
9946 {\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
9947 \indexspace}%
9948 }
```

**\glstreeindent** Length governing left indent for each level of the tree style.

```
9949 \newlength\glstreeindent
9950 \setlength{\glstreeindent}{10pt}
```

**treenoname** The treenoname glossary style is like the tree style, but doesn't print the name or symbol for sub-levels.

```
9951 \newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
9952 \renewenvironment{theglossary}%
9953 {\setlength{\parindent}{0pt}%
9954 \setlength{\parskip}{0pt plus 0.3pt}}%
9955 {}%
```

No header:

```
9956 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9957 \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.

```
9958 \renewcommand{\glossentry}[2]{%
```

```

9959 \hangindentOpt\relax
9960 \parindentOpt\relax
9961 \glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9962 \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9963 \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
9964 }%

```

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.

```

9965 \renewcommand{\subglossentry}[3]{%
9966 \hangindent##1\glstreeindent\relax
9967 \parindent##1\glstreeindent\relax
9968 \ifnum##1=1\relax
9969 \glssubentryitem{##2}%
9970 \fi
9971 \glstarget{##2}{\strut}%
9972 \glossentrydesc{##2}\glspostdescription\space##3\par
9973 }%

```

Vertical gap between groups is the same as that used by indices:

```

9974 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
9975 }

```

**treenonamegroup** Like the `treenoname` style but the glossary groups have headings.

```

9976 \newglossarystyle{treenonamegroup}{%

```

Base it on the `glostytreetreenoname` style:

```

9977 \setglossarystyle{treenoname}%

```

Give each group a heading:

```

9978 \renewcommand{\glsgroupheading}[1]{\par
9979 \noindent\glstreegroupheaderfmt
9980 {\glsggetgrouptitle{##1}}\par\indexspace}%
9981 }

```

**onamehypergroup** The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```

9982 \newglossarystyle{treenonamehypergroup}{%

```

Base it on the `glostytreetreenoname` style:

```

9983 \setglossarystyle{treenoname}%

```

Put navigation links to the groups at the start of the `theglossary` environment:

```

9984 \renewcommand*{\glossaryheader}{%
9985 \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```

9986 \renewcommand*{\glsgroupheading}[1]{%
9987 \par\noindent
9988 \glstreegroupheaderfmt
9989 {\glsnavhypertarget{##1}{\glsggetgrouptitle{##1}}}\par
9990 \indexspace}%
9991 }

```

**esttoplevelname** Find the widest name over all parentless entries in the given glossary or glossaries.

```

9992 \newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
9993   \dimen@=0pt\relax
9994   \gls@tmplen=0pt\relax
9995   \forallglossaries[#1]{\@gls@type}%
9996   {%
9997     \forallglsentries[\@gls@type]{\@glo@label}%
9998     {%
9999       \ifglsahasparent{\@glo@label}%
10000     }%
10001     {%
10002       \settowidth{\dimen@}%
10003         {\glstreenamfmt{\glsentryname{\@glo@label}}}%
10004       \ifdim\dimen@>\gls@tmplen
10005         \gls@tmplen=\dimen@
10006       \letcs{\@glswidestname}{glo\glsdetoklabel{\@glo@label}@name}%
10007     \fi
10008   }%
10009 }%
10010 }%
10011 }

```

**\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.

```

10012 \newcommand*{\glssetwidest}[2][0]{%
10013   \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
10014     #2}%
10015 }

```

**\@glswidestname** Initialise \@glswidestname.

```

10016 \newcommand*{\@glswidestname}{}

```

**\glstreenamebox** Used by the alttree style to create the box for the name and associated information.

```

10017 \newcommand*{\glstreenamebox}[2]{%
10018   \makebox[#1][l]{#2}%
10019 }

```

**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.

```

10020 \newglossarystyle{alttree}{%
  Redefine theglossary environment.
10021   \renewenvironment{theglossary}%
10022     {\def\@gls@prevlevel{-1}%
10023     \mbox{}\par}%
10024     {\par}%
  Set the header and group headers to nothing.
10025   \renewcommand*{\glossaryheader}{}%
10026   \renewcommand*{\glsgroupheading}[1]{}%

```

Redefine the way that the level 0 entries are displayed.

```
10027 \renewcommand{\glossentry}[2]{%
10028   \ifnum\@gls@prevlevel=0\relax
10029   \else
```

Find out how big the indentation should be by measuring the widest entry.

```
10030     \settowidth{\glstreeindent}{\glstreenamfmt{\@glswidestname\space}}%
10031   \fi
```

Set the hangindent and paragraph indent.

```
10032   \hangindent\glstreeindent
10033   \parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
10034   \makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%
10035     \glssentryitem{##1}\glstreenamfmt{\glstarget{##1}{\glossentryname{##1}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
10036   \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
10037   \glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
10038   \def\@gls@prevlevel{0}%
10039 }%
```

Redefine the way sub-entries are displayed.

```
10040 \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.

```
10041   \ifnum##1=1\relax
10042     \glssubentryitem{##2}%
10043   \fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
10044   \ifnum\@gls@prevlevel=##1\relax
10045   \else
```

Compute the widest entry for this level, or for level 0 if not defined for this level. Store in `\gls@tmplen`

```
10046     \@ifundefined{@glswidestname\romannumeral##1}{%
10047       \settowidth{\gls@tmplen}{\glstreenamfmt{\@glswidestname\space}}}%
10048     \settowidth{\gls@tmplen}{\glstreenamfmt{%
10049       \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
10050   \ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
10051      \setlength\glstreeindent\gls@tmplen
10052      \addtolength\glstreeindent\parindent
10053      \parindent\glstreeindent
10054      \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`.

```
10055      \ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
10056      \settowidth{\glstreeindent}{\glstreenamfmt{%
10057      \@glswidestname\space}}}{%
10058      \settowidth{\glstreeindent}{\glstreenamfmt{%
10059      \csname @glswidestname\romannumeral\@gls@prevlevel
10060      \endcsname\space}}}{%}
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```
10061      \addtolength\parindent{-\glstreeindent}%
10062      \setlength\glstreeindent\parindent
10063      \fi
10064      \fi
```

Set the hanging indentation.

```
10065      \hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```
10066      \makebox[0pt][r]{\glstreenambox{\gls@tmplen}{%
10067      \glstreenamfmt{\glstarget{##2}{\glossentryname{##2}}}}}%}
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
10068      \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%}
```

Do the description followed by the description terminator and location list.

```
10069      \glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```
10070      \def\@gls@prevlevel{##1}%
10071      }%
```

Vertical gap between groups is the same as that used by indices:

```
10072      \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
10073 }
```

**almtreegroup** Like the `almtree` style but the glossary groups have headings.

```
10074 \newglossarystyle{almtreegroup}{%
```

Base it on the `glostylealmtree` style:

```
10075      \setglossarystyle{almtree}%}
```

Give each group a heading.

```
10076      \renewcommand{\glsgroupheading}[1]{\par
10077      \def\@gls@prevlevel{-1}%
10078      \hangindent0pt\relax
```

```

10079     \parindent0pt\relax
10080     \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}}%
10081     \par\indexspace}%
10082 }

```

alttreehypergroup The alttreehypergroup style is like the alttreegroup style, but has a set of links to the groups at the start of the glossary.

```

10083 \newglossarystyle{alttreehypergroup}{%
    Base it on the glostylealttree style:
10084     \setglossarystyle{alttree}%
    Put the navigation links in the header
10085     \renewcommand*{\glossaryheader}{%
10086         \par
10087         \def\@gls@prevlevel{-1}%
10088         \hangindent0pt\relax
10089         \parindent0pt\relax
10090         \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    Put a hypertarget at the start of each group
10091     \renewcommand*{\glsgroupheading}[1]{%
10092         \par
10093         \def\@gls@prevlevel{-1}%
10094         \hangindent0pt\relax
10095         \parindent0pt\relax
10096         \glstreegroupheaderfmt
10097         {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
10098         \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.

```
10099 \NeedsTeXFormat{LaTeX2e}
10100 \ProvidesPackage{glossaries-compatible-207}[2021/11/01 v4.49 (NLCT)]
```

**AddXdyAttribute** Adds an attribute in old format.

```
10101 \ifglsxindy
10102   \renewcommand*\GlsAddXdyAttribute[1]{%
10103     \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
10104     \expandafter\toks@\expandafter{\@xdylocref}%
10105     \edef\@xdylocref{\the\toks@ ^^J}%
10106     (markup-locref
10107     :open \string"\string~n\string\setentrycounter
10108       {\noexpand\glscounter}%
10109       \expandafter\string\csname#1\endcsname
10110       \expandafter\@gobble\string\{\string" ^^J
10111       :close \string"\expandafter\@gobble\string\}\string" ^^J
10112       :attr \string"#1\string"))}}
```

Only has an effect before `\writeist`:

```
10113 \fi
```

**sAddXdyCounters**

```
10114 \renewcommand*\GlsAddXdyCounters[1]{%
10115   \GlossariesWarning{\string\GlsAddXdyCounters\space not available
10116     in compatibility mode.}%
10117 }
```

Add predefined attributes

```
10118 \GlsAddXdyAttribute{glsnumberformat}
10119 \GlsAddXdyAttribute{textrm}
10120 \GlsAddXdyAttribute{textsf}
10121 \GlsAddXdyAttribute{texttt}
10122 \GlsAddXdyAttribute{textbf}
10123 \GlsAddXdyAttribute{textmd}
10124 \GlsAddXdyAttribute{textit}
10125 \GlsAddXdyAttribute{textup}
10126 \GlsAddXdyAttribute{textsl}
```

```

10127 \GlsAddXdyAttribute{textsc}
10128 \GlsAddXdyAttribute{emph}
10129 \GlsAddXdyAttribute{glshypernumber}
10130 \GlsAddXdyAttribute{hyperrm}
10131 \GlsAddXdyAttribute{hypersf}
10132 \GlsAddXdyAttribute{hypertt}
10133 \GlsAddXdyAttribute{hyperbf}
10134 \GlsAddXdyAttribute{hypermd}
10135 \GlsAddXdyAttribute{hyperit}
10136 \GlsAddXdyAttribute{hyperup}
10137 \GlsAddXdyAttribute{hypersl}
10138 \GlsAddXdyAttribute{hypersc}
10139 \GlsAddXdyAttribute{hyperemph}

```

sAddXdyLocation Restore v2.07 definition:

```

10140 \ifglxindy
10141 \renewcommand*{\GlsAddXdyLocation}[2]{%
10142   \edef\@xdyuserlocationdefs{%
10143     \@xdyuserlocationdefs ^^J%
10144     (define-location-class \string"#1\string"^^J\space\space
10145     \space(#2))
10146   }%
10147   \edef\@xdyuserlocationnames{%
10148     \@xdyuserlocationnames^^J\space\space\space
10149     \string"#1\string"}%
10150 }
10151 \fi

```

\@do@wrglossary

```

10152 \renewcommand{\@do@wrglossary}[1]{%
  Determine whether to use xindy or makeindex syntax
10153 \ifglxindy
  Need to determine if the formatting information starts with a ( or ) indicating a range.
10154 \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
10155 \def\@glo@range{}%
10156 \expandafter\if\@glo@prefix(\relax
10157   \def\@glo@range{:open-range}%
10158 \else
10159   \expandafter\if\@glo@prefix)\relax
10160   \def\@glo@range{:close-range}%
10161 \fi
10162 \fi

  Get the location and escape any special characters
10163 \protected@edef\@glslocref{\theglentrycounter}%
10164 \@gls@checkmkidxchars\@glslocref

  Write to the glossary file using xindy syntax.
10165 \glossary[\csname glo@#1@type\endcsname]{%

```



```

10166 (indexentry :tkey (\csname glo@#1@index\endcsname)
10167 :locoref \string"\@glslocoref\string" %
10168 :attr \string"\@glo@suffix\string" \@glo@range
10169 )
10170 }%
10171 \else

```

Convert the format information into the format required for makeindex

```

10172 \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat

```

Write to the glossary file using makeindex syntax.

```

10173 \glossary[\csname glo@#1@type\endcsname]{%
10174 \string\glossaryentry{\csname glo@#1@index\endcsname
10175 \@gls@encapchar\@glo@numfmt}{\theglsentrycounter}}%
10176 \fi
10177 }

```

t@glo@numformat Only had 3 arguments in v2.07

```

10178 \def\@set@glo@numformat#1#2#3{%
10179 \expandafter\@glo@check@mkidxrangechar#3\@nil
10180 \protected@edef#1{%
10181 \@glo@prefix setentrycounter[]{\#2}%
10182 \expandafter\string\csname\@glo@suffix\endcsname
10183 }%
10184 \@gls@checkmkidxchars#1%
10185 }

```

\writeist Redefine \writeist back to the way it was in v2.07, but change \istfile to \glswrite.

```

10186 \ifglxindy
10187 \def\writeist{%
10188 \openout\glswrite=\istfilename
10189 \write\glswrite{;; xindy style file created by the glossaries
10190 package in compatible-2.07 mode}%
10191 \write\glswrite{;; for document '\jobname' on
10192 \the\year-\the\month-\the\day}%
10193 \write\glswrite{^^J; required styles^^J}
10194 \@for\@xdystyle:=\@xdyrequiredstyles\do{%
10195 \ifx\@xdystyle\@empty
10196 \else
10197 \protected@write\glswrite{{(require
10198 \string"\@xdystyle.xdy\string")}}%
10199 \fi
10200 }%
10201 \write\glswrite{^^J%
10202 ; list of allowed attributes (number formats)^^J}%
10203 \write\glswrite{(define-attributes ((\@xdyattributes)))}%
10204 \write\glswrite{^^J; user defined alphabets^^J}%
10205 \write\glswrite{\@xdyuseralphabets}%
10206 \write\glswrite{^^J; location class definitions^^J}%
10207 \protected@edef\@gls@roman{\@roman{0}\string"

```

```

10208     \string"roman-numbers-lowercase\string" :sep \string"}}%
10209 \@onelevel@sanitize\@gls@roman
10210 \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
10211     :sep \string"}}%
10212 \@onelevel@sanitize\@tmp
10213 \ifx\@tmp\@gls@roman
10214     \write\glswrite{(define-location-class
10215         \string"roman-page-numbers\string"^^J\space\space\space
10216         (\string"roman-numbers-lowercase\string")
10217         :min-range-length \@glsminrange)}}%
10218 \else
10219     \write\glswrite{(define-location-class
10220         \string"roman-page-numbers\string"^^J\space\space\space
10221         (:sep "\@gls@roman")
10222         :min-range-length \@glsminrange)}}%
10223 \fi
10224 \write\glswrite{(define-location-class
10225     \string"Roman-page-numbers\string"^^J\space\space\space
10226     (\string"roman-numbers-uppercase\string")
10227     :min-range-length \@glsminrange)}}%
10228 \write\glswrite{(define-location-class
10229     \string"arabic-page-numbers\string"^^J\space\space\space
10230     (\string"arabic-numbers\string")
10231     :min-range-length \@glsminrange)}}%
10232 \write\glswrite{(define-location-class
10233     \string"alpha-page-numbers\string"^^J\space\space\space
10234     (\string"alpha\string")
10235     :min-range-length \@glsminrange)}}%
10236 \write\glswrite{(define-location-class
10237     \string"Alpha-page-numbers\string"^^J\space\space\space
10238     (\string"ALPHA\string")
10239     :min-range-length \@glsminrange)}}%
10240 \write\glswrite{(define-location-class
10241     \string"Appendix-page-numbers\string"^^J\space\space\space
10242     (\string"ALPHA\string"
10243     :sep \string"\@glsAlphacompositor\string"
10244     \string"arabic-numbers\string")
10245     :min-range-length \@glsminrange)}}%
10246 \write\glswrite{(define-location-class
10247     \string"arabic-section-numbers\string"^^J\space\space\space
10248     (\string"arabic-numbers\string"
10249     :sep \string"\glscompositor\string"
10250     \string"arabic-numbers\string")
10251     :min-range-length \@glsminrange)}}%
10252 \write\glswrite{^^J; user defined location classes}%
10253 \write\glswrite{\@xdyuserlocationdefs}%
10254 \write\glswrite{^^J; define cross-reference class^^J}%
10255 \write\glswrite{(define-crossref-class \string"see\string"
10256     :unverified )}%

```

```

10257 \write\glswrite{(markup-crossref-list
10258 :class \string"see\string"^^J\space\space\space
10259 :open \string"\string\glseeformat\string"
10260 :close \string"{}\string")}%
10261 \write\glswrite{^^J; define the order of the location classes}%
10262 \write\glswrite{(define-location-class-order
10263 (\@xdylocationclassorder))}%
10264 \write\glswrite{^^J; define the glossary markup^^J}%
10265 \write\glswrite{(markup-index^^J\space\space\space
10266 :open \string"\string
10267 \glossarysection[\string\glossarytoctitle]{\string
10268 \glossarytitle}\string\glossarypreamble\string~n\string\begin
10269 {theglossary}\string\glossaryheader\string~n\string" ^^J\space
10270 \space\space:close \string"\expandafter\@gobble
10271 \string%\string~n\string
10272 \end{theglossary}\string\glossarypostamble
10273 \string~n\string" ^^J\space\space\space
10274 :tree)}}%
10275 \write\glswrite{(markup-letter-group-list
10276 :sep \string"\string\glsgroupskip\string~n\string")}%
10277 \write\glswrite{(markup-indexentry
10278 :open \string"\string\relax \string\glresetentrylist
10279 \string~n\string")}%
10280 \write\glswrite{(markup-locclass-list :open
10281 \string"\glsoopenbrace\string\glossaryentrynumbers
10282 \glsoopenbrace\string\relax\space \string"^^J\space\space\space
10283 :sep \string", \string"
10284 :close \string"\glsclosebrace\glsclosebrace\string")}%
10285 \write\glswrite{(markup-locref-list
10286 :sep \string"\string\delimN\space\string")}%
10287 \write\glswrite{(markup-range
10288 :sep \string"\string\delimR\space\string")}%
10289 \@onelevel@sanitize\gls@suffixF
10290 \@onelevel@sanitize\gls@suffixFF
10291 \ifx\gls@suffixF\@empty
10292 \else
10293 \write\glswrite{(markup-range
10294 :close "\gls@suffixF" :length 1 :ignore-end)}%
10295 \fi
10296 \ifx\gls@suffixFF\@empty
10297 \else
10298 \write\glswrite{(markup-range
10299 :close "\gls@suffixFF" :length 2 :ignore-end)}%
10300 \fi
10301 \write\glswrite{^^J; define format to use for locations^^J}%
10302 \write\glswrite{\@xdylocref}%
10303 \write\glswrite{^^J; define letter group list format^^J}%
10304 \write\glswrite{(markup-letter-group-list
10305 :sep \string"\string\glsgroupskip\string~n\string")}%

```

```

10306 \write\glswrite{^^J; letter group headings^^J}%
10307 \write\glswrite{(markup-letter-group
10308   :open-head \string"\string\glsgroupheading
10309   \glsoopenbrace\string"^^J\space\space\space
10310   :close-head \string"\glsclosebrace\string")}%
10311 \write\glswrite{^^J; additional letter groups^^J}%
10312 \write\glswrite{\@xdylettergroups}%
10313 \write\glswrite{^^J; additional sort rules^^J}
10314 \write\glswrite{\@xdysortrules}%
10315 \noist}
10316 \else
10317 \edef\@gls@actualchar{\string?}
10318 \edef\@gls@encapchar{\string|}
10319 \edef\@gls@levelchar{\string!}
10320 \edef\@gls@quotechar{\string"}
10321 \def\writeist{\relax
10322   \openout\glswrite=\istfilename
10323   \write\glswrite{\expandafter\@gobble\string\% makeindex style file
10324     created by the glossaries package}
10325   \write\glswrite{\expandafter\@gobble\string\% for document
10326     '\jobname' on \the\year-\the\month-\the\day}
10327   \write\glswrite{actual '\@gls@actualchar'}
10328   \write\glswrite{encap '\@gls@encapchar'}
10329   \write\glswrite{level '\@gls@levelchar'}
10330   \write\glswrite{quote '\@gls@quotechar'}
10331   \write\glswrite{keyword \string"\string\glossaryentry\string"}
10332   \write\glswrite{preamble \string"\string\glossarysection[\string
10333     \glossarytoctitle]{\string\glossarytitle}\string
10334     \glossarypreamble\string\n\string\begin{theglossary}\string
10335     \glossaryheader\string\n\string"}
10336   \write\glswrite{postamble \string"\string%\string\n\string
10337     \end{theglossary}\string\glossarypostamble\string\n
10338     \string"}
10339   \write\glswrite{group_skip \string"\string\glsgroupskip\string\n
10340     \string"}
10341   \write\glswrite{item_0 \string"\string%\string\n\string"}
10342   \write\glswrite{item_1 \string"\string%\string\n\string"}
10343   \write\glswrite{item_2 \string"\string%\string\n\string"}
10344   \write\glswrite{item_01 \string"\string%\string\n\string"}
10345   \write\glswrite{item_x1
10346     \string"\string\relax \string\glsresetentrylist\string\n
10347     \string"}
10348   \write\glswrite{item_12 \string"\string%\string\n\string"}
10349   \write\glswrite{item_x2
10350     \string"\string\relax \string\glsresetentrylist\string\n
10351     \string"}
10352   \write\glswrite{delim_0 \string"\string\{\string
10353     \glossaryentrynumbers\string\{\string\relax \string"}
10354   \write\glswrite{delim_1 \string"\string\{\string

```

```

10355     \glossaryentrynumbers\string\{\string\relax \string}
10356     \write\glswrite{delim_2 \string"\string\{\string
10357     \glossaryentrynumbers\string\{\string\relax \string}
10358     \write\glswrite{delim_t \string"\string\}\string\}\string}
10359     \write\glswrite{delim_n \string"\string\delimN \string}
10360     \write\glswrite{delim_r \string"\string\delimR \string}
10361     \write\glswrite{headings_flag 1}
10362     \write\glswrite{heading_prefix
10363         \string"\string\glsgroupheading\string\{\string}
10364     \write\glswrite{heading_suffix
10365         \string"\string\}\string\relax
10366         \string\glresetentrylist \string}
10367     \write\glswrite{symhead_positive \string"glssymbols\string}
10368     \write\glswrite{numhead_positive \string"glslnumbers\string}
10369     \write\glswrite{page_compositor \string"glscpositor\string}
10370     \@gls@escbsdq\gls@suffixF
10371     \@gls@escbsdq\gls@suffixFF
10372     \ifx\gls@suffixF\@empty
10373     \else
10374         \write\glswrite{suffix_2p \string"\gls@suffixF\string}
10375     \fi
10376     \ifx\gls@suffixFF\@empty
10377     \else
10378         \write\glswrite{suffix_3p \string"\gls@suffixFF\string}
10379     \fi
10380     \noist
10381 }
10382 \fi

```

\noist

```
10383 \renewcommand*{\noist}{\let\writeist\relax}
```

## 4.2 glossaries-compatible-307

```

10384 \NeedsTeXFormat{LaTeX2e}
10385 \ProvidesPackage{glossaries-compatible-307}[2021/11/01 v4.49 (NLCT)]

```

Compatibility macros for predefined glossary styles:

`\atglossarystyle` Defines a compatibility glossary style.

```

10386 \newcommand{\compatglossarystyle}[2]{%
10387     \ifcsundef{@glscompstyle@#1}%
10388     {%
10389         \csdef{@glscompstyle@#1}{#2}%
10390     }%
10391     {%
10392         \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
10393     }%
10394 }

```

Backward compatible inline style.

```

10395 \compatglossarystyle{inline}{%
10396   \renewcommand{\glossaryentryfield}[5]{%
10397     \glsinlinedopostchild
10398     \gls@inlinesep
10399     \def\glo@desc{##3}%
10400     \def\@no@post@desc{\nopostdesc}%
10401     \glstentryitem{##1}\glsinlinenameformat{##1}{##2}%
10402     \ifx\glo@desc\@no@post@desc
10403       \glsinlineemptydescformat{##4}{##5}%
10404     \else
10405       \ifstrepty{##3}%
10406         {\glsinlineemptydescformat{##4}{##5}}%
10407         {\glsinlinedescformat{##3}{##4}{##5}}%
10408     \fi
10409     \ifglshaschildren{##1}%
10410     {%
10411       \glsresetsubentrycounter
10412       \glsinlineparentchildseparator
10413       \def\gls@inlinesubsep{}%
10414       \def\gls@inlinepostchild{\glsinlinepostchild}%
10415     }%
10416     {}%
10417     \def\gls@inlinesep{\glsinlineseparator}%
10418   }%

```

Sub-entries display description:

```

10419   \renewcommand{\glossarysubentryfield}[6]{%
10420     \gls@inlinesubsep%
10421     \glsinlinesubnameformat{##2}{##3}%
10422     \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
10423     \def\gls@inlinesubsep{\glsinlinesubseparator}%
10424   }%
10425 }

```

Backward compatible list style.

```

10426 \compatglossarystyle{list}{%
10427   \renewcommand*{\glossaryentryfield}[5]{%
10428     \item[\glstentryitem{##1}\glstarget{##1}{##2}]
10429       ##3\glspostdescription\space ##5}%

```

Sub-entries continue on the same line:

```

10430   \renewcommand*{\glossarysubentryfield}[6]{%
10431     \glssubentryitem{##2}%
10432     \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
10433 }

```

Backward compatible listgroup style.

```

10434 \compatglossarystyle{listgroup}{%
10435   \csuse{@glscompstyle@list}%
10436 }%

```

Backward compatible listhypergroup style.

```
10437 \compatglossarystyle{listhypergroup}{%
10438   \csuse{@glscompstyle@list}%
10439 }%
```

Backward compatible altlist style.

```
10440 \compatglossarystyle{altlist}{%
10441   \renewcommand*{\glossaryentryfield}[5]{%
10442     \item[\glstentryitem{##1}\glstarget{##1}{##2}]%
10443     \mbox{}\par\nobreak\@afterheading
10444     ##3\glspostdescription\space ##5}%
10445   \renewcommand{\glossarysubentryfield}[6]{%
10446     \par
10447     \glssubentryitem{##2}%
10448     \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
10449 }%
```

Backward compatible altlistgroup style.

```
10450 \compatglossarystyle{altlistgroup}{%
10451   \csuse{@glscompstyle@altlist}%
10452 }%
```

Backward compatible altlisthypergroup style.

```
10453 \compatglossarystyle{altlisthypergroup}{%
10454   \csuse{@glscompstyle@altlist}%
10455 }%
```

Backward compatible listdotted style.

```
10456 \compatglossarystyle{listdotted}{%
10457   \renewcommand*{\glossaryentryfield}[5]{%
10458     \item[\makebox[\glslistdottedwidth][l]{%
10459       \glstentryitem{##1}\glstarget{##1}{##2}%
10460       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
10461   \renewcommand*{\glossarysubentryfield}[6]{%
10462     \item[\makebox[\glslistdottedwidth][l]{%
10463       \glssubentryitem{##2}%
10464       \glstarget{##2}{##3}%
10465       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
10466 }%
```

Backward compatible sublistdotted style.

```
10467 \compatglossarystyle{sublistdotted}{%
10468   \csuse{@glscompstyle@listdotted}%
10469   \renewcommand*{\glossaryentryfield}[5]{%
10470     \item[\glstentryitem{##1}\glstarget{##1}{##2}]}%
10471 }%
```

Backward compatible long style.

```
10472 \compatglossarystyle{long}{%
10473   \renewcommand*{\glossaryentryfield}[5]{%
10474     \glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
10475   \renewcommand*{\glossarysubentryfield}[6]{%
10476     \glssubentryitem{##2}\glstarget{##2}{##3} & ##4\glspostdescription\space ##5\\}%
10477 }%
```

```

10476      &
10477      \glssubentryitem{##2}%
10478      \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
10479 }%

```

Backward compatible longborder style.

```

10480 \compatglossarystyle{longborder}{%
10481   \csuse{@glscmpstyle@long}%
10482 }%

```

Backward compatible longheader style.

```

10483 \compatglossarystyle{longheader}{%
10484   \csuse{@glscmpstyle@long}%
10485 }%

```

Backward compatible longheaderborder style.

```

10486 \compatglossarystyle{longheaderborder}{%
10487   \csuse{@glscmpstyle@long}%
10488 }%

```

Backward compatible long3col style.

```

10489 \compatglossarystyle{long3col}{%
10490   \renewcommand*{\glossaryentryfield}[5]{%
10491     \glstentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
10492   \renewcommand*{\glossarysubentryfield}[6]{%
10493     &
10494     \glssubentryitem{##2}%
10495     \glstarget{##2}{\strut}##4 & ##6\\}%
10496 }%

```

Backward compatible long3colborder style.

```

10497 \compatglossarystyle{long3colborder}{%
10498   \csuse{@glscmpstyle@long3col}%
10499 }%

```

Backward compatible long3colheader style.

```

10500 \compatglossarystyle{long3colheader}{%
10501   \csuse{@glscmpstyle@long3col}%
10502 }%

```

Backward compatible long3colheaderborder style.

```

10503 \compatglossarystyle{long3colheaderborder}{%
10504   \csuse{@glscmpstyle@long3col}%
10505 }%

```

Backward compatible long4col style.

```

10506 \compatglossarystyle{long4col}{%
10507   \renewcommand*{\glossaryentryfield}[5]{%
10508     \glstentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
10509   \renewcommand*{\glossarysubentryfield}[6]{%
10510     &
10511     \glssubentryitem{##2}%

```



10512        \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%  
10513 }%

Backward compatible long4colheader style.

10514 \compatglossarystyle{long4colheader}{%  
10515    \csuse{@glscompstyle@long4col}%  
10516 }%

Backward compatible long4colborder style.

10517 \compatglossarystyle{long4colborder}{%  
10518    \csuse{@glscompstyle@long4col}%  
10519 }%

Backward compatible long4colheaderborder style.

10520 \compatglossarystyle{long4colheaderborder}{%  
10521    \csuse{@glscompstyle@long4col}%  
10522 }%

Backward compatible altlong4col style.

10523 \compatglossarystyle{altlong4col}{%  
10524    \csuse{@glscompstyle@long4col}%  
10525 }%

Backward compatible altlong4colheader style.

10526 \compatglossarystyle{altlong4colheader}{%  
10527    \csuse{@glscompstyle@long4col}%  
10528 }%

Backward compatible altlong4colborder style.

10529 \compatglossarystyle{altlong4colborder}{%  
10530    \csuse{@glscompstyle@long4col}%  
10531 }%

Backward compatible altlong4colheaderborder style.

10532 \compatglossarystyle{altlong4colheaderborder}{%  
10533    \csuse{@glscompstyle@long4col}%  
10534 }%

Backward compatible long style.

10535 \compatglossarystyle{longragged}{%  
10536    \renewcommand\*{\glossaryentryfield}[5]{%  
10537      \glssentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%  
10538      \tabularnewline}%  
10539    \renewcommand\*{\glossarysubentryfield}[6]{%  
10540      &  
10541      \glssubentryitem{##2}%  
10542      \glstarget{##2}{\strut}##4\glspostdescription\space ##6%  
10543      \tabularnewline}%  
10544 }%

Backward compatible longraggedborder style.

10545 \compatglossarystyle{longraggedborder}{%  
10546    \csuse{@glscompstyle@longragged}%  
10547 }%

Backward compatible longraggedheader style.

```
10548 \compatglossarystyle{longraggedheader}{%
10549 \csuse{@glscmpstyle@longragged}%
10550 }%
```

Backward compatible longraggedheaderborder style.

```
10551 \compatglossarystyle{longraggedheaderborder}{%
10552 \csuse{@glscmpstyle@longragged}%
10553 }%
```

Backward compatible longragged3col style.

```
10554 \compatglossarystyle{longragged3col}{%
10555 \renewcommand*{\glossaryentryfield}[5]{%
10556 \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
10557 \renewcommand*{\glossarysubentryfield}[6]{%
10558 &
10559 \glssubentryitem{##2}%
10560 \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
10561 }%
```

Backward compatible longragged3colborder style.

```
10562 \compatglossarystyle{longragged3colborder}{%
10563 \csuse{@glscmpstyle@longragged3col}%
10564 }%
```

Backward compatible longragged3colheader style.

```
10565 \compatglossarystyle{longragged3colheader}{%
10566 \csuse{@glscmpstyle@longragged3col}%
10567 }%
```

Backward compatible longragged3colheaderborder style.

```
10568 \compatglossarystyle{longragged3colheaderborder}{%
10569 \csuse{@glscmpstyle@longragged3col}%
10570 }%
```

Backward compatible altlongragged4col style.

```
10571 \compatglossarystyle{altlongragged4col}{%
10572 \renewcommand*{\glossaryentryfield}[5]{%
10573 \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
10574 \renewcommand*{\glossarysubentryfield}[6]{%
10575 &
10576 \glssubentryitem{##2}%
10577 \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
10578 }%
```

Backward compatible altlongragged4colheader style.

```
10579 \compatglossarystyle{altlongragged4colheader}{%
10580 \csuse{@glscmpstyle@altlong4col}%
10581 }%
```

Backward compatible altlongragged4colborder style.

```
10582 \compatglossarystyle{altlongragged4colborder}{%
```

```

10583 \csuse{@glscompstyle@altlong4col}%
10584 }%

```

Backward compatible altlongragged4colheaderborder style.

```

10585 \compatglossarystyle{altlongragged4colheaderborder}{%
10586 \csuse{@glscompstyle@altlong4col}%
10587 }%

```

Backward compatible index style.

```

10588 \compatglossarystyle{index}{%
10589 \renewcommand*{\glossaryentryfield}[5]{%
10590 \item\glsentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10591 \ifx\relax##4\relax
10592 \else
10593 \space{##4}%
10594 \fi
10595 \space ##3\glspostdescription \space ##5}%
10596 \renewcommand*{\glossarysubentryfield}[6]{%
10597 \ifcase##1\relax
10598 % level 0
10599 \item
10600 \or
10601 % level 1
10602 \subitem
10603 \glssubentryitem{##2}%
10604 \else
10605 % all other levels
10606 \subsubitem
10607 \fi
10608 \textbf{\glstarget{##2}{##3}}%
10609 \ifx\relax##5\relax
10610 \else
10611 \space{##5}%
10612 \fi
10613 \space##4\glspostdescription\space ##6}%
10614 }%

```

Backward compatible indexgroup style.

```

10615 \compatglossarystyle{indexgroup}{%
10616 \csuse{@glscompstyle@index}%
10617 }%

```

Backward compatible indexhypergroup style.

```

10618 \compatglossarystyle{indexhypergroup}{%
10619 \csuse{@glscompstyle@index}%
10620 }%

```

Backward compatible tree style.

```

10621 \compatglossarystyle{tree}{%
10622 \renewcommand{\glossaryentryfield}[5]{%
10623 \hangindent0pt\relax

```

```

10624 \parindent0pt\relax
10625 \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10626 \ifx\relax##4\relax
10627 \else
10628 \space(##4)%
10629 \fi
10630 \space ##3\glspostdescription \space ##5\par}%
10631 \renewcommand{\glossarysubentryfield}[6]{%
10632 \hangindent##1\glstreeindent\relax
10633 \parindent##1\glstreeindent\relax
10634 \ifnum##1=1\relax
10635 \glssubentryitem{##2}%
10636 \fi
10637 \textbf{\glstarget{##2}{##3}}%
10638 \ifx\relax##5\relax
10639 \else
10640 \space(##5)%
10641 \fi
10642 \space##4\glspostdescription\space ##6\par}%
10643 }%

```

Backward compatible treegroup style.

```

10644 \compatglossarystyle{treegroup}{%
10645 \csuse{@glscmpstyle@tree}%
10646 }%

```

Backward compatible treehypergroup style.

```

10647 \compatglossarystyle{treehypergroup}{%
10648 \csuse{@glscmpstyle@tree}%
10649 }%

```

Backward compatible treenoname style.

```

10650 \compatglossarystyle{treenoname}{%
10651 \renewcommand{\glossaryentryfield}[5]{%
10652 \hangindent0pt\relax
10653 \parindent0pt\relax
10654 \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10655 \ifx\relax##4\relax
10656 \else
10657 \space(##4)%
10658 \fi
10659 \space ##3\glspostdescription \space ##5\par}%
10660 \renewcommand{\glossarysubentryfield}[6]{%
10661 \hangindent##1\glstreeindent\relax
10662 \parindent##1\glstreeindent\relax
10663 \ifnum##1=1\relax
10664 \glssubentryitem{##2}%
10665 \fi
10666 \glstarget{##2}{\strut}%
10667 ##4\glspostdescription\space ##6\par}%
10668 }%

```

Backward compatible treenonamegroup style.

```
10669 \compatglossarystyle{treenonamegroup}{%
10670   \csuse{@glscompstyle@treenoname}%
10671 }%
```

Backward compatible treenonamehypergroup style.

```
10672 \compatglossarystyle{treenonamehypergroup}{%
10673   \csuse{@glscompstyle@treenoname}%
10674 }%
```

Backward compatible alttree style.

```
10675 \compatglossarystyle{alttree}{%
10676   \renewcommand{\glossaryentryfield}[5]{%
10677     \ifnum\@gls@prevlevel=0\relax
10678     \else
10679       \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
10680       \hangindent\glstreeindent
10681       \parindent\glstreeindent
10682     \fi
10683     \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
10684       \glssentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
10685     \ifx\relax##4\relax
10686     \else
10687       (##4)\space
10688     \fi
10689     ##3\glspostdescription \space ##5\par
10690     \def\@gls@prevlevel{0}%
10691   }%
10692   \renewcommand{\glossarysubentryfield}[6]{%
10693     \ifnum##1=1\relax
10694       \glssubentryitem{##2}%
10695     \fi
10696     \ifnum\@gls@prevlevel=##1\relax
10697     \else
10698       \@ifundefined{@glswidestname\romannumeral##1}{%
10699         \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
10700         \settowidth{\gls@tmplen}{\textbf{%
10701           \csname @glswidestname\romannumeral##1\endcsname\space}}}%
10702         \ifnum\@gls@prevlevel<##1\relax
10703           \setlength\glstreeindent\gls@tmplen
10704           \addtolength\glstreeindent\parindent
10705           \parindent\glstreeindent
10706         \else
10707           \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
10708             \settowidth{\glstreeindent}{\textbf{%
10709               \@glswidestname\space}}{%
10710             \settowidth{\glstreeindent}{\textbf{%
10711               \csname @glswidestname\romannumeral\@gls@prevlevel
10712               \endcsname\space}}}%
10713             \addtolength\parindent{-\glstreeindent}%

```

```

10714         \setlength\glstreeindent\parindent
10715     \fi
10716 \fi
10717 \hangindent\glstreeindent
10718 \makebox[0pt][r]{\makebox[\glstemplen][l]{%
10719     \textbf{\glstarget{##2}{##3}}}%
10720 \ifx##5\relax\relax
10721 \else
10722     (##5)\space
10723 \fi
10724 ##4\glspostdescription\space ##6\par
10725 \def\@gls@prevlevel{##1}%
10726 }%
10727 }%

```

Backward compatible alttreegroup style.

```

10728 \compatglossarystyle{alttreegroup}{%
10729 \csuse{@glscompstyle@alttree}%
10730 }%

```

Backward compatible alttreehypergroup style.

```

10731 \compatglossarystyle{alttreehypergroup}{%
10732 \csuse{@glscompstyle@alttree}%
10733 }%

```

Backward compatible mcolindex style.

```

10734 \compatglossarystyle{mcolindex}{%
10735 \csuse{@glscompstyle@index}%
10736 }%

```

Backward compatible mcolindexgroup style.

```

10737 \compatglossarystyle{mcolindexgroup}{%
10738 \csuse{@glscompstyle@index}%
10739 }%

```

Backward compatible mcolindexhypergroup style.

```

10740 \compatglossarystyle{mcolindexhypergroup}{%
10741 \csuse{@glscompstyle@index}%
10742 }%

```

Backward compatible mcoltree style.

```

10743 \compatglossarystyle{mcoltree}{%
10744 \csuse{@glscompstyle@tree}%
10745 }%

```

Backward compatible mcoltreegroup style.

```

10746 \compatglossarystyle{mcolindextreegroup}{%
10747 \csuse{@glscompstyle@tree}%
10748 }%

```

Backward compatible mcoltreehypergroup style.

```

10749 \compatglossarystyle{mcolindextreehypergroup}{%

```

10750 \csuse{@glscompstyle@tree}%  
 10751 }%

Backward compatible mcoltreenoname style.

10752 \compatglossarystyle{mcoltreenoname}{%  
 10753 \csuse{@glscompstyle@tree}%  
 10754 }%

Backward compatible mcoltreenonamegroup style.

10755 \compatglossarystyle{mcoltreenonamegroup}{%  
 10756 \csuse{@glscompstyle@tree}%  
 10757 }%

Backward compatible mcoltreenonamehypergroup style.

10758 \compatglossarystyle{mcoltreenonamehypergroup}{%  
 10759 \csuse{@glscompstyle@tree}%  
 10760 }%

Backward compatible mcolalmtree style.

10761 \compatglossarystyle{mcolalmtree}{%  
 10762 \csuse{@glscompstyle@almtree}%  
 10763 }%

Backward compatible mcolalmtreegroup style.

10764 \compatglossarystyle{mcolalmtreegroup}{%  
 10765 \csuse{@glscompstyle@almtree}%  
 10766 }%

Backward compatible mcolalmtreehypergroup style.

10767 \compatglossarystyle{mcolalmtreehypergroup}{%  
 10768 \csuse{@glscompstyle@almtree}%  
 10769 }%

Backward compatible superragged style.

10770 \compatglossarystyle{superragged}{%  
 10771 \renewcommand\*{\glossaryentryfield}[5]{%  
 10772 \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%  
 10773 \tabularnewline}%  
 10774 \renewcommand\*{\glossarysubentryfield}[6]{%  
 10775 &  
 10776 \glssubentryitem{##2}%  
 10777 \glstarget{##2}{\strut}##4\glspostdescription\space ##6%  
 10778 \tabularnewline}%  
 10779 }%

Backward compatible superraggedborder style.

10780 \compatglossarystyle{superraggedborder}{%  
 10781 \csuse{@glscompstyle@superragged}%  
 10782 }%

Backward compatible superraggedheader style.

10783 \compatglossarystyle{superraggedheader}{%  
 10784 \csuse{@glscompstyle@superragged}%  
 10785 }%

Backward compatible superraggedheaderborder style.

```
10786 \compatglossarystyle{superraggedheaderborder}{%
10787   \csuse{@glscompstyle@superragged}%
10788 }%
```

Backward compatible superragged3col style.

```
10789 \compatglossarystyle{superragged3col}{%
10790   \renewcommand*{\glossaryentryfield}[5]{%
10791     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
10792   \renewcommand*{\glossarysubentryfield}[6]{%
10793     &
10794     \glssubentryitem{##2}%
10795     \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
10796 }%
```

Backward compatible superragged3colborder style.

```
10797 \compatglossarystyle{superragged3colborder}{%
10798   \csuse{@glscompstyle@superragged3col}%
10799 }%
```

Backward compatible superragged3colheader style.

```
10800 \compatglossarystyle{superragged3colheader}{%
10801   \csuse{@glscompstyle@superragged3col}%
10802 }%
```

Backward compatible superragged3colheaderborder style.

```
10803 \compatglossarystyle{superragged3colheaderborder}{%
10804   \csuse{@glscompstyle@superragged3col}%
10805 }%
```

Backward compatible altsuperragged4col style.

```
10806 \compatglossarystyle{altsuperragged4col}{%
10807   \renewcommand*{\glossaryentryfield}[5]{%
10808     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
10809   \renewcommand*{\glossarysubentryfield}[6]{%
10810     &
10811     \glssubentryitem{##2}%
10812     \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
10813 }%
```

Backward compatible altsuperragged4colheader style.

```
10814 \compatglossarystyle{altsuperragged4colheader}{%
10815   \csuse{@glscompstyle@altsuperragged4col}%
10816 }%
```

Backward compatible altsuperragged4colborder style.

```
10817 \compatglossarystyle{altsuperragged4colborder}{%
10818   \csuse{@glscompstyle@altsuperragged4col}%
10819 }%
```

Backward compatible altsuperragged4colheaderborder style.

```
10820 \compatglossarystyle{altsuperragged4colheaderborder}{%
```



```
10821 \csuse{@glscompstyle@altsuperragged4col}%
10822 }%
```

Backward compatible super style.

```
10823 \compatglossarystyle{super}{%
10824   \renewcommand*{\glossaryentryfield}[5]{%
10825     \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
10826   \renewcommand*{\glossarysubentryfield}[6]{%
10827     &
10828     \glssubentryitem{##2}%
10829     \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
10830 }%
```

Backward compatible superborder style.

```
10831 \compatglossarystyle{superborder}{%
10832   \csuse{@glscompstyle@super}%
10833 }%
```

Backward compatible superheader style.

```
10834 \compatglossarystyle{superheader}{%
10835   \csuse{@glscompstyle@super}%
10836 }%
```

Backward compatible superheaderborder style.

```
10837 \compatglossarystyle{superheaderborder}{%
10838   \csuse{@glscompstyle@super}%
10839 }%
```

Backward compatible super3col style.

```
10840 \compatglossarystyle{super3col}{%
10841   \renewcommand*{\glossaryentryfield}[5]{%
10842     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
10843   \renewcommand*{\glossarysubentryfield}[6]{%
10844     &
10845     \glssubentryitem{##2}%
10846     \glstarget{##2}{\strut}##4 & ##6\\}%
10847 }%
```

Backward compatible super3colborder style.

```
10848 \compatglossarystyle{super3colborder}{%
10849   \csuse{@glscompstyle@super3col}%
10850 }%
```

Backward compatible super3colheader style.

```
10851 \compatglossarystyle{super3colheader}{%
10852   \csuse{@glscompstyle@super3col}%
10853 }%
```

Backward compatible super3colheaderborder style.

```
10854 \compatglossarystyle{super3colheaderborder}{%
10855   \csuse{@glscompstyle@super3col}%
10856 }%
```

Backward compatible super4col style.

```
10857 \compatglossarystyle{super4col}{%
10858   \renewcommand*{\glossaryentryfield}[5]{%
10859     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
10860   \renewcommand*{\glossarysubentryfield}[6]{%
10861     &
10862     \glssubentryitem{##2}%
10863     \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
10864 }%
```

Backward compatible super4colheader style.

```
10865 \compatglossarystyle{super4colheader}{%
10866   \csuse{@glscompstyle@super4col}%
10867 }%
```

Backward compatible super4colborder style.

```
10868 \compatglossarystyle{super4colborder}{%
10869   \csuse{@glscompstyle@super4col}%
10870 }%
```

Backward compatible super4colheaderborder style.

```
10871 \compatglossarystyle{super4colheaderborder}{%
10872   \csuse{@glscompstyle@super4col}%
10873 }%
```

Backward compatible altsuper4col style.

```
10874 \compatglossarystyle{altsuper4col}{%
10875   \csuse{@glscompstyle@super4col}%
10876 }%
```

Backward compatible altsuper4colheader style.

```
10877 \compatglossarystyle{altsuper4colheader}{%
10878   \csuse{@glscompstyle@super4col}%
10879 }%
```

Backward compatible altsuper4colborder style.

```
10880 \compatglossarystyle{altsuper4colborder}{%
10881   \csuse{@glscompstyle@super4col}%
10882 }%
```

Backward compatible altsuper4colheaderborder style.

```
10883 \compatglossarystyle{altsuper4colheaderborder}{%
10884   \csuse{@glscompstyle@super4col}%
10885 }%
```

## 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.

```
10886 \NeedsTeXFormat{LaTeX2e}
```

Package version number now in line with main glossaries package number.

```
10887 \ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)]
```

```
10888 Experimental glossaries accessibility]
```

Pass all options to glossaries:

```
10889 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
10890 \ProcessOptions
```

This package should be loaded before glossaries-extra, so complain if that has already been loaded.

```
10891 \@ifpackageloaded{glossaries-extra}
```

```
10892 {%
```

If the accsupp option was used, `\@glsxtr@doaccsupp` will have been set, otherwise it will be empty.

```
10893 \ifx\@glsxtr@doaccsupp\empty
```

```
10894 \GlossariesWarning{The ‘glossaries-accsupp’
```

```
10895 package has been loaded\MessageBreak
```

```
10896 after the ‘glossaries-extra’ package. This\MessageBreak
```

```
10897 can cause a failure to integrate both packages. \MessageBreak
```

```
10898 Either use the ‘accsupp’ option when you load\MessageBreak
```

```
10899 ‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
```

```
10900 before loading ‘glossaries-extra’}%
```

```
10901 \fi
```

```
10902 }
```

```
10903 {}
```

`\tumbleglossentry` Override style compatibility macros:

```
10904 \def\compatibleglossentry#1#2{%
```

```
10905 \toks@{#2}%
```

```
10906 \protected@edef\do@glossentry{%
```

```
10907 \noexpand\accsuppglossaryentryfield{#1}%
```

```
10908 {\noexpand\glsnamefont
```

```
10909 {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
```

```

10910    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
10911    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
10912    {\the\toks@}%
10913  }%
10914  \@do@glossentry
10915 }

```

lesubglossentry

```

10916 \def\compatiblesubglossentry#1#2#3{%
10917   \toks@{#3}%
10918   \protected@edef\@do@subglossentry{%
10919     \noexpand\accsuppglossarysubentryfield{\number#1}%
10920     {#2}%
10921     {\noexpand\glsnamefont
10922      {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
10923     {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
10924     {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
10925     {\the\toks@}%
10926   }%
10927   \@do@subglossentry
10928 }

```

Required packages:

```

10929 \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.

```

10930 \providecommand{\gls@accsupp@engine}{accsupp}

```

s@accessibility `\gls@accessibility[\<options>]{\<PDF element>}{\<value>}{\<content>}`

```

10931 \providecommand{\gls@accessibility}[4]{#4}
10932 \ifdefstring\gls@accsupp@engine{accsupp}
10933 {
10934   \RequirePackage{accsupp}
10935   \renewcommand{\gls@accessibility}[4]{%
10936     \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp}%
10937   }
10938 }
10939 {}

```

lsaccessibility `\glsaccessibility[\<options>]{\<PDF element>}{\<value>}{\<content>}`

User-level command that includes debug info if required.

```

10940 \newcommand{\glsaccessibility}[4] [] {%
10941   \@glsshowaccsupp{#1}{#2}{#3}%
10942   \gls@accessibility{#1}{#2}{#3}{#4}%
10943 }

```

## 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:

```

10944 \define@key{glossentry}{access}{%
10945   \def\@glo@access{#1}%
10946 }

```

**textaccess** The replacement text corresponding to the text key:

```

10947 \define@key{glossentry}{textaccess}{%
10948   \def\@glo@textaccess{#1}%
10949 }

```

**firstaccess** The replacement text corresponding to the first key:

```

10950 \define@key{glossentry}{firstaccess}{%
10951   \def\@glo@firstaccess{#1}%
10952 }

```

**pluralaccess** The replacement text corresponding to the plural key:

```

10953 \define@key{glossentry}{pluralaccess}{%
10954   \def\@glo@pluralaccess{#1}%
10955 }

```

**firstpluralaccess** The replacement text corresponding to the firstplural key:

```

10956 \define@key{glossentry}{firstpluralaccess}{%
10957   \def\@glo@firstpluralaccess{#1}%
10958 }

```

**symbolaccess** The replacement text corresponding to the symbol key:

```

10959 \define@key{glossentry}{symbolaccess}{%
10960   \def\@glo@symbolaccess{#1}%
10961 }

```

**symbolpluralaccess** The replacement text corresponding to the symbolplural key:

```

10962 \define@key{glossentry}{symbolpluralaccess}{%
10963   \def\@glo@symbolpluralaccess{#1}%
10964 }

```

descriptionaccess The replacement text corresponding to the description key:

```
10965 \define@key{glossentry}{descriptionaccess}{%
10966   \def\@glo@descaccess{#1}%
10967 }
```

ionpluralaccess The replacement text corresponding to the descriptionplural key:

```
10968 \define@key{glossentry}{descriptionpluralaccess}{%
10969   \def\@glo@descpluralaccess{#1}%
10970 }
```

shortaccess The replacement text corresponding to the short key:

```
10971 \define@key{glossentry}{shortaccess}{%
10972   \def\@glo@shortaccess{#1}%
10973 }
```

ortpluralaccess The replacement text corresponding to the shortplural key:

```
10974 \define@key{glossentry}{shortpluralaccess}{%
10975   \def\@glo@shortpluralaccess{#1}%
10976 }
```

longaccess The replacement text corresponding to the long key:

```
10977 \define@key{glossentry}{longaccess}{%
10978   \def\@glo@longaccess{#1}%
10979 }
```

ongpluralaccess The replacement text corresponding to the longplural key:

```
10980 \define@key{glossentry}{longpluralaccess}{%
10981   \def\@glo@longpluralaccess{#1}%
10982 }
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
10983 \define@key{glossentry}{user1access}{%
10984   \def\@glo@useriaccess{#1}%
10985 }
```

user2access The replacement text corresponding to the user2 key:

```
10986 \define@key{glossentry}{user2access}{%
10987   \def\@glo@useriiaccess{#1}%
10988 }
```

user3access The replacement text corresponding to the user3 key:

```
10989 \define@key{glossentry}{user3access}{%
10990   \def\@glo@useriiiaccess{#1}%
10991 }
```

user4access The replacement text corresponding to the user4 key:

```
10992 \define@key{glossentry}{user4access}{%
10993   \def\@glo@userivaccess{#1}%
10994 }
```

user5access The replacement text corresponding to the user5 key:

```
10995 \define@key{glossentry}{user5access}{%
10996   \def\@glo@uservaccess{#1}%
10997 }
```

user6access The replacement text corresponding to the user6 key:

```
10998 \define@key{glossentry}{user6access}{%
10999   \def\@glo@userviaccess{#1}%
11000 }
```

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:

```
11001 \appto\@gls@keymap{,%
11002   {access}{access},%
11003   {textaccess}{textaccess},%
11004   {firstaccess}{firstaccess},%
11005   {pluralaccess}{pluralaccess},%
11006   {firstpluralaccess}{firstpluralaccess},%
11007   {symbolaccess}{symbolaccess},%
11008   {symbolpluralaccess}{symbolpluralaccess},%
11009   {descaccess}{descaccess},%
11010   {descpluralaccess}{descpluralaccess},%
11011   {shortaccess}{shortaccess},%
11012   {shortpluralaccess}{shortpluralaccess},%
11013   {longaccess}{longaccess},%
11014   {longpluralaccess}{longpluralaccess},%
11015   {user1access}{useriaccess},%
11016   {user2access}{useriiaccess},%
11017   {user3access}{useriiiaccess},%
11018   {user4access}{userivaccess},%
11019   {user5access}{uservaccess},%
11020   {user6access}{userviaccess}%
11021 }
```

\@gls@noaccess Indicates that no replacement text has been provided.

```
11022 \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.

```
11023 \let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
11024 \renewcommand*{\@newglossaryentryprehook}{%
```

```

11025 \@gls@oldnewglossaryentryprehook
11026 \def\@glo@access{\relax}%

```

Initialise the other keys:

```

11027 \def\@glo@textaccess{\@glo@access}%
11028 \def\@glo@firstaccess{\@glo@access}%
11029 \def\@glo@pluralaccess{\@glo@textaccess}%
11030 \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
11031 \def\@glo@symbolaccess{\relax}%
11032 \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
11033 \def\@glo@descaccess{\relax}%
11034 \def\@glo@descpluralaccess{\@glo@descaccess}%
11035 \def\@glo@shortaccess{\relax}%
11036 \def\@glo@shortpluralaccess{\@glo@shortaccess}%
11037 \def\@glo@longaccess{\relax}%
11038 \def\@glo@longpluralaccess{\@glo@longaccess}%
11039 \def\@glo@useriaccess{\relax}%
11040 \def\@glo@useriiaaccess{\relax}%
11041 \def\@glo@useriiiaaccess{\relax}%
11042 \def\@glo@userivaccess{\relax}%
11043 \def\@glo@uservaccess{\relax}%
11044 \def\@glo@userviaccess{\relax}%
11045 }

```

Add to the end hook:

```

11046 \let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
11047 \renewcommand*{\@newglossaryentryposthook}{%
11048   \@gls@oldnewglossaryentryposthook

```

Store the access information:

```

11049 \expandafter
11050   \protected@xdef\csname glo@\@glo@label @access\endcsname{%
11051     \@glo@access}%
11052 \expandafter
11053   \protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
11054     \@glo@textaccess}%
11055 \expandafter
11056   \protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
11057     \@glo@firstaccess}%
11058 \expandafter
11059   \protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
11060     \@glo@pluralaccess}%
11061 \expandafter
11062   \protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
11063     \@glo@firstpluralaccess}%
11064 \expandafter
11065   \protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
11066     \@glo@symbolaccess}%
11067 \expandafter
11068   \protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
11069     \@glo@symbolpluralaccess}%

```



```

11070 \expandafter
11071   \protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
11072     \@glo@descaccess}%
11073 \expandafter
11074   \protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
11075     \@glo@descpluralaccess}%
11076 \expandafter
11077   \protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
11078     \@glo@shortaccess}%
11079 \expandafter
11080   \protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
11081     \@glo@shortpluralaccess}%
11082 \expandafter
11083   \protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
11084     \@glo@longaccess}%
11085 \expandafter
11086   \protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
11087     \@glo@longpluralaccess}%
11088 \expandafter
11089   \protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
11090     \@glo@useriaccess}%
11091 \expandafter
11092   \protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
11093     \@glo@useriiaccess}%
11094 \expandafter
11095   \protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
11096     \@glo@useriiiaccess}%
11097 \expandafter
11098   \protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
11099     \@glo@userivaccess}%
11100 \expandafter
11101   \protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
11102     \@glo@uservaccess}%
11103 \expandafter
11104   \protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
11105     \@glo@userviaccess}%
11106 }

```

## 5.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

11107 \newcommand*{\glsentryaccess}[1]{%
11108   \@gls@entry@field{#1}{access}%
11109 }

```

`entrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

11110 \newcommand*{\glsentrytextaccess}[1]{%
11111   \@gls@entry@field{#1}{textaccess}%

```

```

11112 }

entryfirstaccess  Get the value of the firstaccess key for the entry with the given label:
11113 \newcommand*{\glentryfirstaccess}[1]{%
11114   \@gls@entry@field{#1}{firstaccess}%
11115 }

entrypluralaccess  Get the value of the pluralaccess key for the entry with the given label:
11116 \newcommand*{\glentrypluralaccess}[1]{%
11117   \@gls@entry@field{#1}{pluralaccess}%
11118 }

firstpluralaccess  Get the value of the firstpluralaccess key for the entry with the given label:
11119 \newcommand*{\glentryfirstpluralaccess}[1]{%
11120   \@gls@entry@field{#1}{firstpluralaccess}%
11121 }

entrysymbolaccess  Get the value of the symbolaccess key for the entry with the given label:
11122 \newcommand*{\glentrysymbolaccess}[1]{%
11123   \@gls@entry@field{#1}{symbolaccess}%
11124 }

symbolpluralaccess  Get the value of the symbolpluralaccess key for the entry with the given label:
11125 \newcommand*{\glentrysymbolpluralaccess}[1]{%
11126   \@gls@entry@field{#1}{symbolpluralaccess}%
11127 }

entrydescaccess  Get the value of the descriptionaccess key for the entry with the given label:
11128 \newcommand*{\glentrydescaccess}[1]{%
11129   \@gls@entry@field{#1}{descaccess}%
11130 }

descpluralaccess  Get the value of the descriptionpluralaccess key for the entry with the given label:
11131 \newcommand*{\glentrydescpluralaccess}[1]{%
11132   \@gls@entry@field{#1}{descpluralaccess}%
11133 }

entryshortaccess  Get the value of the shortaccess key for the entry with the given label:
11134 \newcommand*{\glentryshortaccess}[1]{%
11135   \@gls@entry@field{#1}{shortaccess}%
11136 }

shortpluralaccess  Get the value of the shortpluralaccess key for the entry with the given label:
11137 \newcommand*{\glentryshortpluralaccess}[1]{%
11138   \@gls@entry@field{#1}{shortpluralaccess}%
11139 }

```

entrylongaccess Get the value of the longaccess key for the entry with the given label:

```
11140 \newcommand*{\glsentrylongaccess}[1]{%
11141   \@gls@entry@field{#1}{longaccess}%
11142 }
```

ongpluralaccess Get the value of the longpluralaccess key for the entry with the given label:

```
11143 \newcommand*{\glsentrylongpluralaccess}[1]{%
11144   \@gls@entry@field{#1}{longpluralaccess}%
11145 }
```

entryuseriaccess Get the value of the user1access key for the entry with the given label:

```
11146 \newcommand*{\glsentryuseriaccess}[1]{%
11147   \@gls@entry@field{#1}{useriaccess}%
11148 }
```

tryuseriiaccess Get the value of the user2access key for the entry with the given label:

```
11149 \newcommand*{\glsentryuseriiaccess}[1]{%
11150   \@gls@entry@field{#1}{useriiaccess}%
11151 }
```

ryuseriiiaccess Get the value of the user3access key for the entry with the given label:

```
11152 \newcommand*{\glsentryuseriiiaccess}[1]{%
11153   \@gls@entry@field{#1}{useriiiaccess}%
11154 }
```

tryuserivaccess Get the value of the user4access key for the entry with the given label:

```
11155 \newcommand*{\glsentryuserivaccess}[1]{%
11156   \@gls@entry@field{#1}{userivaccess}%
11157 }
```

entryuservaccess Get the value of the user5access key for the entry with the given label:

```
11158 \newcommand*{\glsentryuservaccess}[1]{%
11159   \@gls@entry@field{#1}{uservaccess}%
11160 }
```

tryuserviaccess Get the value of the user6access key for the entry with the given label:

```
11161 \newcommand*{\glsentryuserviaccess}[1]{%
11162   \@gls@entry@field{#1}{userviaccess}%
11163 }
```

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`.

```
11164 \newcommand{\glsfieldaccsupp}[4]{%
11165   \ifdef\glscategory
11166   {%
11167     \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
11168     {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
11169     {%
11170       \ifcsdef{glsxtr\glscategory{#4}accsupp}%
11171       {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
11172       {%
11173         \ifcsdef{gls#3accsupp}%
11174         {\csname gls#3accsupp\endcsname{#1}{#2}}%
11175         {\glsaccsupp{#1}{#2}}%
11176       }%
11177     }%
11178   }%
11179   {%
11180     \ifcsdef{gls#3accsupp}%
11181     {\csname gls#3accsupp\endcsname{#1}{#2}}%
11182     {\glsaccsupp{#1}{#2}}%
11183   }%
11184 }
```

glsfieldaccsupp

`\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}`

As `\glsfieldaccsupp` but fully expand replacement text.

```
11185 \newcommand{\xglsfieldaccsupp}[1]{%
11186   \protected@edef\@gls@replacementtext{#1}%
11187   \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
11188 }
```

glsshortaccsupp

`\glsshortaccsupp{<replacement text>}{<text>}`

```
11189 \newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

`\glsshortplaccsupp` `\glsshortplaccsupp{<replacement text>}{<text>}`

```
11190 \newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

`\glsaccsupp` `\glsaccsupp{<replacement text>}{<text>}`

```
11191 \newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
```

`\xglsaccsupp` Fully expands replacement text before calling `\glsaccsupp`

```
11192 \newcommand*{\xglsaccsupp}[2]{%
11193   \protected@edef\@gls@replacementtext{#1}%
11194   \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
11195 }
```

`@access@display` Deprecated. Use `\@gls@fieldaccess@display` instead.

```
11196 \newcommand*{\@gls@access@display}[2]{%
11197   \protected@edef\@glo@access{#2}%
11198   \ifx\@glo@access\@gls@noaccess
11199     #1%
11200   \else
11201     \xglsaccsupp{\@glo@access}{#1}%
11202   \fi
11203 }
```

`daccess@display` `\@gls@fieldaccess@display{<label>}{<field>}{<content>}{<replacement>}`

```
11204 \newcommand*{\@gls@fieldaccess@display}[4]{%
11205   \protected@edef\@glo@access{#4}%
11206   \ifdefequal\@glo@access\@gls@noaccess
11207     {#3}%
11208     {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
11209 }
```

`meaccessdisplay` Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```
11210 \newrobustcmd*{\glsnameaccessdisplay}[2]{%
11211   \ifcsundef{glo@glsdetoklabel{#2}@access}%
11212     {#1}%
11213     {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
11214 }
```

`xtaccessdisplay` As above but for the textaccess replacement text.

```
11215 \newrobustcmd*{\glsxtaccessdisplay}[2]{%
11216   \ifcsundef{glo@glsdetoklabel{#2}@textaccess}%
11217     {#1}%
11218     {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentrytextaccess{#2}}}%
11219 }
```

```

11218 {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
11219 }

alaccessdisplay As above but for the pluralaccess replacement text.
11220 \newrobustcmd*{\glspluralaccessdisplay}[2]{%
11221 \ifcsundef{glo@glstetoklabel{#2}@pluralaccess}%
11222 {#1}%
11223 {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
11224 }

staccessdisplay As above but for the firstaccess replacement text.
11225 \newrobustcmd*{\glsfirstaccessdisplay}[2]{%
11226 \ifcsundef{glo@glstetoklabel{#2}@firstaccess}%
11227 {#1}%
11228 {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
11229 }

alaccessdisplay As above but for the firstpluralaccess replacement text.
11230 \newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
11231 \ifcsundef{glo@glstetoklabel{#2}@firstpluralaccess}%
11232 {#1}%
11233 {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
11234 }

olaccessdisplay As above but for the symbolaccess replacement text.
11235 \newrobustcmd*{\glsymbolaccessdisplay}[2]{%
11236 \ifcsundef{glo@glstetoklabel{#2}@symbolaccess}%
11237 {#1}%
11238 {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
11239 }

alaccessdisplay As above but for the symbolpluralaccess replacement text.
11240 \newrobustcmd*{\glsymbolpluralaccessdisplay}[2]{%
11241 \ifcsundef{glo@glstetoklabel{#2}@symbolpluralaccess}%
11242 {#1}%
11243 {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
11244 }

onaccessdisplay As above but for the descriptionaccess replacement text.
11245 \newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
11246 \ifcsundef{glo@glstetoklabel{#2}@descaccess}%
11247 {#1}%
11248 {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
11249 }

alaccessdisplay As above but for the descriptionpluralaccess replacement text.
11250 \newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
11251 \ifcsundef{glo@glstetoklabel{#2}@descpluralaccess}%

```

```

11252 {#1}%
11253 {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glentrydescpluralaccess{#2}}}%
11254 }

```

rtaccessdisplay As above but for the shortaccess replacement text.

```

11255 \newrobustcmd*{\glsshortaccessdisplay}[2]{%
11256 \ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
11257 {#1}%
11258 {\@gls@fieldaccess@display{#2}{short}{#1}{\glentryshortaccess{#2}}}%
11259 }

```

alaccessdisplay As above but for the shortpluralaccess replacement text.

```

11260 \newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
11261 \ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
11262 {#1}%
11263 {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glentryshortpluralaccess{#2}}}%
11264 }

```

ngaccessdisplay As above but for the longaccess replacement text.

```

11265 \newrobustcmd*{\glslongaccessdisplay}[2]{%
11266 \ifcsundef{glo@glsdetoklabel{#2}@longaccess}%
11267 {#1}%
11268 {\@gls@fieldaccess@display{#2}{long}{#1}{\glentrylongaccess{#2}}}%
11269 }

```

alaccessdisplay As above but for the longpluralaccess replacement text.

```

11270 \newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
11271 \ifcsundef{glo@glsdetoklabel{#2}@longpluralaccess}%
11272 {#1}%
11273 {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glentrylongpluralaccess{#2}}}%
11274 }

```

riaccessdisplay As above but for the user1access replacement text.

```

11275 \newrobustcmd*{\gluseriaccessdisplay}[2]{%
11276 \ifcsundef{glo@glsdetoklabel{#2}@useriaccess}%
11277 {#1}%
11278 {\@gls@fieldaccess@display{#2}{useri}{#1}{\glentryuseriaccess{#2}}}%
11279 }

```

iiaccessdisplay As above but for the user2access replacement text.

```

11280 \newrobustcmd*{\gluseriiaccessdisplay}[2]{%
11281 \ifcsundef{glo@glsdetoklabel{#2}@useriiaccess}%
11282 {#1}%
11283 {\@gls@fieldaccess@display{#2}{userii}{#1}{\glentryuseriiaccess{#2}}}%
11284 }

```

iiiaccessdisplay As above but for the user3access replacement text.

```

11285 \newrobustcmd*{\gluseriiiaccessdisplay}[2]{%

```

```

11286 \ifcsundef{glo@glsdetoklabel{#2}@useriiiaccess}%
11287 {#1}%
11288 {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glentryuseriiiaccess{#2}}}%
11289 }

```

**ivaccessdisplay** As above but for the user4access replacement text.

```

11290 \newrobustcmd*{\gluserivaccessdisplay}[2]{%
11291 \ifcsundef{glo@glsdetoklabel{#2}@userivaccess}%
11292 {#1}%
11293 {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glentryuserivaccess{#2}}}%
11294 }

```

**rvaccessdisplay** As above but for the user5access replacement text.

```

11295 \newrobustcmd*{\glsuservaccessdisplay}[2]{%
11296 \ifcsundef{glo@glsdetoklabel{#2}@uservaccess}%
11297 {#1}%
11298 {\@gls@fieldaccess@display{#2}{userv}{#1}{\glentryuservaccess{#2}}}%
11299 }

```

**viaccessdisplay** As above but for the user6access replacement text.

```

11300 \newrobustcmd*{\glsuserviaccessdisplay}[2]{%
11301 \ifcsundef{glo@glsdetoklabel{#2}@userviaccess}%
11302 {#1}%
11303 {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glentryuserviaccess{#2}}}%
11304 }

```

**lsaccessdisplay** Gets the replacement text corresponding to the named key given by the first argument and calls the appropriate command defined above.

```

11305 \newrobustcmd*{\glsaccessdisplay}[3]{%
11306 \ifcsundef{gls#1accessdisplay}%
11307 {%
11308 \PackageError{glossaries-accsupp}{No accessibility support
11309 for key ‘#1’}{}%
11310 }%
11311 {%
11312 \csname gls#1accessdisplay\endcsname{#2}{#3}%
11313 }%
11314 }

```

**default@entryfmt** Redefine the default entry format to use accessibility information

```

11315 \renewcommand*{\@gls@default@entryfmt}[2]{%
11316 \ifdefempty\glscustomtext
11317 {%
11318 \glsifplural
11319 {%

```

Plural form

```

11320 \glscapscase
11321 {%

```



Don't adjust case

```
11322      \ifglused\glslabel
11323      {%
```

Subsequent use

```
11324      #2{\glsppluralaccessdisplay
11325          {\glentryplural{\glslabel}}{\glslabel}}%
11326          {\glsdescriptionpluralaccessdisplay
11327              {\glentrydescplural{\glslabel}}{\glslabel}}%
11328              {\glssymbolpluralaccessdisplay
11329                  {\glentrysymbolplural{\glslabel}}{\glslabel}}
11330              {\glinsert}%
11331      }%
11332      {%
```

First use

```
11333      #1{\glfirstpluralaccessdisplay
11334          {\glentryfirstplural{\glslabel}}{\glslabel}}%
11335          {\glsdescriptionpluralaccessdisplay
11336              {\glentrydescplural{\glslabel}}{\glslabel}}%
11337              {\glssymbolpluralaccessdisplay
11338                  {\glentrysymbolplural{\glslabel}}{\glslabel}}%
11339              {\glinsert}%
11340      }%
11341      }%
11342      {%
```

Make first letter upper case

```
11343      \ifglused\glslabel
11344      {%
```

Subsequent use.

```
11345      #2{\glsppluralaccessdisplay
11346          {\Glentryplural{\glslabel}}{\glslabel}}%
11347          {\glsdescriptionpluralaccessdisplay
11348              {\glentrydescplural{\glslabel}}{\glslabel}}%
11349              {\glssymbolpluralaccessdisplay
11350                  {\glentrysymbolplural{\glslabel}}{\glslabel}}%
11351              {\glinsert}%
11352      }%
11353      {%
```

First use

```
11354      #1{\glfirstpluralaccessdisplay
11355          {\Glentryfirstplural{\glslabel}}{\glslabel}}%
11356          {\glsdescriptionpluralaccessdisplay
11357              {\glentrydescplural{\glslabel}}{\glslabel}}%
11358              {\glssymbolpluralaccessdisplay
11359                  {\glentrysymbolplural{\glslabel}}{\glslabel}}%
11360              {\glinsert}%
11361      }%
```

11362 }%  
 11363 {%

Make all upper case

11364 \ifglused\glslabel  
 11365 {%

Subsequent use

11366 \MakeUppercase{%  
 11367 #2{\glsppluralaccessdisplay  
 11368 {\glentryplural{\glslabel}}{\glslabel}}%  
 11369 {\gldescriptionpluralaccessdisplay  
 11370 {\glentrydescplural{\glslabel}}{\glslabel}}%  
 11371 {\glssymbolpluralaccessdisplay  
 11372 {\glentrysymbolplural{\glslabel}}{\glslabel}}%  
 11373 {\glinsert}}%  
 11374 }%  
 11375 {%

First use

11376 \MakeUppercase{%  
 11377 #1{\glfirstpluralaccessdisplay  
 11378 {\glentryfirstplural{\glslabel}}{\glslabel}}%  
 11379 {\gldescriptionpluralaccessdisplay  
 11380 {\glentrydescplural{\glslabel}}{\glslabel}}%  
 11381 {\glssymbolpluralaccessdisplay  
 11382 {\glentrysymbolplural{\glslabel}}{\glslabel}}%  
 11383 {\glinsert}}%  
 11384 }%  
 11385 }%  
 11386 }%  
 11387 {%

Singular form

11388 \glscapscase  
 11389 {%

Don't adjust case

11390 \ifglused\glslabel  
 11391 {%

Subsequent use

11392 #2{\glstextaccessdisplay  
 11393 {\glentrytext{\glslabel}}{\glslabel}}%  
 11394 {\gldescriptionaccessdisplay  
 11395 {\glentrydesc{\glslabel}}{\glslabel}}%  
 11396 {\glssymbolaccessdisplay  
 11397 {\glentrysymbol{\glslabel}}{\glslabel}}%  
 11398 {\glinsert}}%  
 11399 }%  
 11400 {%

#### First use

```

11401      #1{\glsfirstaccessdisplay
11402          {\glsentryfirst{\glslabel}}{\glslabel}}%
11403          {\glsdescriptionaccessdisplay
11404              {\glsentrydesc{\glslabel}}{\glslabel}}%
11405          {\glsymbolaccessdisplay
11406              {\glsentrysymbol{\glslabel}}{\glslabel}}%
11407          {\glsinsert}}%
11408      }%
11409  }%
11410  {%

```

#### Make first letter upper case

```

11411      \ifglsused\glslabel
11412      {%

```

#### Subsequent use

```

11413      #2{\glstextaccessdisplay
11414          {\Glsentrytext{\glslabel}}{\glslabel}}%
11415          {\glsdescriptionaccessdisplay
11416              {\glsentrydesc{\glslabel}}{\glslabel}}%
11417          {\glsymbolaccessdisplay
11418              {\glsentrysymbol{\glslabel}}{\glslabel}}%
11419          {\glsinsert}}%
11420      }%
11421      {%

```

#### First use

```

11422      #1{\glsfirstaccessdisplay
11423          {\Glsentryfirst{\glslabel}}{\glslabel}}%
11424          {\glsdescriptionaccessdisplay
11425              {\glsentrydesc{\glslabel}}{\glslabel}}%
11426          {\glsymbolaccessdisplay
11427              {\glsentrysymbol{\glslabel}}{\glslabel}}%
11428          {\glsinsert}}%
11429      }%
11430  }%
11431  {%

```

#### Make all upper case

```

11432      \ifglsused\glslabel
11433      {%

```

#### Subsequent use

```

11434      \MakeUppercase{%
11435          #2{\glstextaccessdisplay
11436              {\glsentrytext{\glslabel}}{\glslabel}}%
11437              {\glsdescriptionaccessdisplay
11438                  {\glsentrydesc{\glslabel}}{\glslabel}}%
11439                  {\glsymbolaccessdisplay
11440                      {\glsentrysymbol{\glslabel}}{\glslabel}}%

```

```

11441         {\glsinsert}}}%
11442     }%
11443     {%

```

First use

```

11444     \MakeUppercase{%
11445         #1{\glsfirstaccessdisplay
11446             {\glsentryfirst{\glslabel}}{\glslabel}}}%
11447         {\glsdescriptionaccessdisplay
11448             {\glsentrydesc{\glslabel}}{\glslabel}}}%
11449         {\glsymbolaccessdisplay
11450             {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11451         {\glsinsert}}}%
11452     }%
11453 }%
11454 }%
11455 }%
11456 {%

```

Custom text provided in \glsdisp

```

11457     \ifglsused{\glslabel}%
11458     {%

```

Subsequent use

```

11459         #2{\glscustomtext}%
11460         {\glsdescriptionaccessdisplay
11461             {\glsentrydesc{\glslabel}}{\glslabel}}}%
11462         {\glsymbolaccessdisplay
11463             {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11464         {\glsinsert}}}%
11465     }%
11466     {%

```

First use

```

11467         #1{\glscustomtext}%
11468         {\glsdescriptionaccessdisplay
11469             {\glsentrydesc{\glslabel}}{\glslabel}}}%
11470         {\glsymbolaccessdisplay
11471             {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11472         {\glsinsert}}}%
11473     }%
11474 }%
11475 }

```

`\glsenentryfmt` Redefine to use accessibility information.

```

11476 \renewcommand*{\glsenentryfmt}{%
11477     \ifdefempty\glscustomtext
11478     {%
11479         \glsifplural
11480         {%

```

### Plural form

```
11481      \glscapscase
11482      {%
```

### Don't adjust case

```
11483      \ifglused\glslabel
11484      {%
```

### Subsequent use

```
11485      \glspluralaccessdisplay
11486      {\glentryplural{\glslabel}}{\glslabel}%
11487      \glinsert
11488      }%
11489      {%
```

### First use

```
11490      \glfirstpluralaccessdisplay
11491      {\glentryfirstplural{\glslabel}}{\glslabel}%
11492      \glinsert
11493      }%
11494      }%
11495      {%
```

### Make first letter upper case

```
11496      \ifglused\glslabel
11497      {%
```

### Subsequent use.

```
11498      \glspluralaccessdisplay
11499      {\Glentryplural{\glslabel}}{\glslabel}%
11500      \glinsert
11501      }%
11502      {%
```

### First use

```
11503      \glfirstpluralaccessdisplay
11504      {\Glentryfirstplural{\glslabel}}{\glslabel}%
11505      \glinsert
11506      }%
11507      }%
11508      {%
```

### Make all upper case

```
11509      \ifglused\glslabel
11510      {%
```

### Subsequent use

```
11511      \glspluralaccessdisplay
11512      {\mfirstucMakeUppercase{\glentryplural{\glslabel}}}%
11513      {\glslabel}%
11514      \mfirstucMakeUppercase{\glinsert}%
11515      }%
11516      {%
```

#### First use

```

11517      \glsfirstpluralaccessdisplay
11518      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
11519      {\glslabel}}%
11520      \mfirstucMakeUppercase{\glsinsert}%
11521      }%
11522      }%
11523      }%
11524      {%

```

#### Singular form

```

11525      \glscapscase
11526      {%

```

#### Don't adjust case

```

11527      \ifglused\glslabel
11528      {%

```

#### Subsequent use

```

11529      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
11530      \glsinsert
11531      }%
11532      {%

```

#### First use

```

11533      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
11534      \glsinsert
11535      }%
11536      }%
11537      {%

```

#### Make first letter upper case

```

11538      \ifglused\glslabel
11539      {%

```

#### Subsequent use

```

11540      \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
11541      \glsinsert
11542      }%
11543      {%

```

#### First use

```

11544      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
11545      \glsinsert
11546      }%
11547      }%
11548      {%

```

#### Make all upper case

```

11549      \ifglused\glslabel
11550      {%

```

#### Subsequent use

```

11551      \glstextaccessdisplay
11552      {\mfirstucMakeUppercase{\glentrytext{\glslabel}}}{\glslabel}%
11553      \mfirstucMakeUppercase{\glsinsert}%
11554      }%
11555      {%

```

#### First use

```

11556      \glsfirstaccessdisplay
11557      {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
11558      \mfirstucMakeUppercase{\glsinsert}%
11559      }%
11560      }%
11561      }%
11562      }%
11563      {%

```

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.

```

11564      \glscustomtext\glsinsert
11565      }%
11566 }

```

`\glsgenacfmt`   Redefine to include accessibility information.

```

11567 \renewcommand*{\glsgenacfmt}{%
11568   \ifdefempty\glscustomtext
11569   {%
11570     \ifglsused\glslabel
11571     {%

```

#### Subsequent use:

```

11572      \glsifplural
11573      {%

```

#### Subsequent plural form:

```

11574      \glscapscase
11575      {%

```

#### Subsequent plural form, don't adjust case:

```

11576      \acronymfont
11577      {\glsshortpluralaccessdisplay
11578       {\glentryshortpl{\glslabel}}{\glslabel}}%
11579      \glsinsert
11580      }%
11581      {%

```

#### Subsequent plural form, make first letter upper case:

```

11582      \acronymfont
11583      {\glsshortpluralaccessdisplay
11584       {\Glsentryshortpl{\glslabel}}{\glslabel}}%
11585      \glsinsert

```

11586 }%

11587 {%

Subsequent plural form, all caps:

11588 \mfirstucMakeUppercase

11589 {\acronymfont

11590 {\glsshortpluralaccessdisplay

11591 {\glentryshortpl{\glslabel}}{\glslabel}}%

11592 \glsinsert}%

11593 }%

11594 }%

11595 {%

Subsequent singular form

11596 \glscapscase

11597 {%

Subsequent singular form, don't adjust case:

11598 \acronymfont

11599 {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%

11600 \glsinsert

11601 }%

11602 {%

Subsequent singular form, make first letter upper case:

11603 \acronymfont

11604 {\glsshortaccessdisplay{\Glentryshort{\glslabel}}{\glslabel}}%

11605 \glsinsert

11606 }%

11607 {%

Subsequent singular form, all caps:

11608 \mfirstucMakeUppercase

11609 {\acronymfont{%

11610 \glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%

11611 \glsinsert}%

11612 }%

11613 }%

11614 }%

11615 {%

First use:

11616 \glsifplural

11617 {%

First use plural form:

11618 \glscapscase

11619 {%

First use plural form, don't adjust case:

11620 \genplacrformat{\glslabel}{\glsinsert}%

11621 }%

11622 {%



First use plural form, make first letter upper case:

```
11623      \Genplacrfullformat{\glslabel}{\glsinsert}%
11624      }%
11625      {%
```

First use plural form, all caps:

```
11626      \mfirstucMakeUppercase
11627      {\genplacrfullformat{\glslabel}{\glsinsert}}%
11628      }%
11629      }%
11630      {%
```

First use singular form

```
11631      \glscapscase
11632      {%
```

First use singular form, don't adjust case:

```
11633      \genacrfullformat{\glslabel}{\glsinsert}%
11634      }%
11635      {%
```

First use singular form, make first letter upper case:

```
11636      \Genacrfullformat{\glslabel}{\glsinsert}%
11637      }%
11638      {%
```

First use singular form, all caps:

```
11639      \mfirstucMakeUppercase
11640      {\genacrfullformat{\glslabel}{\glsinsert}}%
11641      }%
11642      }%
11643      }%
11644      }%
11645      {%
```

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.

```
11646      \glscustomtext
11647      }%
11648 }
```

`\genacrfullformat` Redefine to include accessibility information.

```
11649 \renewcommand*{\genacrfullformat}[2]{%
11650   \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
11651   (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}{#1}})%
11652 }
```

`\genacrfullformat` Redefine to include accessibility information.

```
11653 \renewcommand*{\Genacrfullformat}[2]{%
11654   \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
11655   (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}{#1}})%
11656 }
```

placrfullformat   Redefine to include accessibility information.

```
11657 \renewcommand*{\genplacrfullformat}[2]{%
11658   \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
11659   (\glsshortpluralaccessdisplay
11660     {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
11661 }
```

placrfullformat   Redefine to include accessibility information.

```
11662 \renewcommand*{\Genplacrfullformat}[2]{%
11663   \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
11664   (\glsshortpluralaccessdisplay
11665     {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
11666 }
```

\@acrshort

```
11667 \def\@acrshort#1#2[#3]{%
11668   \glsdoifexists{#2}%
11669   {%
11670     \let\do@gls@link@checkfirsthyper\relax

11671     \let\glsifplural\@secondoftwo
11672     \let\glsupcase\@firstofthree
11673     \let\glsinsert\@empty
11674     \def\glscustomtext{%
11675       \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
11676     }%

    Call \@gls@link

11677     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11678   }%

11679   \glspostlinkhook
11680 }
```

\@Acrshort

```
11681 \def\@Acrshort#1#2[#3]{%
11682   \glsdoifexists{#2}%
11683   {%
11684     \let\do@gls@link@checkfirsthyper\relax

11685     \let\glsifplural\@secondoftwo
11686     \let\glsupcase\@secondofthree
11687     \let\glsinsert\@empty
11688     \def\glscustomtext{%
11689       \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
11690     }%

    Call \@gls@link

11691     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11692   }%
```

```

11693 \glspostlinkhook
11694 }

```

\@ACRshort

```

11695 \def\@ACRshort#1#2[#3]{%
11696   \glstoifexists{#2}%
11697   {%
11698     \let\do@gl@link@checkfirsthyper\relax

11699     \let\gl@sifplural\@secondoftwo
11700     \let\gl@scapscase\@thirdofthree
11701     \let\gl@insert\@empty
11702     \def\glscustomtext{%
11703       \acronymfont{\glsshortaccessdisplay
11704         {\MakeUppercase{\glentryshort{#2}}}{#2}}#3%
11705     }%

    Call\@gl@link

11706     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11707     }%

11708   \glspostlinkhook
11709 }

```

\@acrlong

```

11710 \def\@acrlong#1#2[#3]{%
11711   \glstoifexists{#2}%
11712   {%
11713     \let\do@gl@link@checkfirsthyper\relax

11714     \let\gl@sifplural\@secondoftwo
11715     \let\gl@scapscase\@firstofthree
11716     \let\gl@insert\@empty
11717     \def\glscustomtext{%
11718       \acronymfont{\glslongaccessdisplay{\glentrylong{#2}}{#2}}#3%
11719     }%

    Call\@gl@link

11720     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11721     }%

11722   \glspostlinkhook
11723 }

```

\@Acrlong

```

11724 \def\@Acrlong#1#2[#3]{%
11725   \glstoifexists{#2}%
11726   {%
11727     \let\do@gl@link@checkfirsthyper\relax

```

```

11728 \let\glsifplural\@secondoftwo
11729 \let\glsifcaps\@firstofthree
11730 \let\glsinsert\@empty
11731 \def\glsfont{\font{\glsfontaccessdisplay{\glsfontlong{#2}}{#2}}#3}%
11732 \def\glsfont{\font{\glsfontaccessdisplay{\glsfontlong{#2}}{#2}}#3}%
11733 }%

```

Call \@gls@link

```

11734 \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
11735 }%

11736 \glspostlinkhook
11737 }

```

\@ACRlong

```

11738 \def\@ACRlong#1#2[#3]{%
11739 \glsdoifexists{#2}%
11740 {%
11741 \let\do@gls@link@checkfirsthyper\relax

11742 \let\glsifplural\@secondoftwo
11743 \let\glsifcaps\@firstofthree
11744 \let\glsinsert\@empty
11745 \def\glsfont{\font{\glsfontaccessdisplay{\glsfontlong{#2}}{#2}}#3}%
11746 \def\glsfont{\font{\glsfontaccessdisplay{\glsfontlong{#2}}{#2}}#3}%
11747 \MakeUppercase{\glsfontlong{#2}}{#2}}#3}%
11748 }%

```

Call \@gls@link

```

11749 \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
11750 }%

11751 \glspostlinkhook
11752 }

```

\@gls@text@

```

11753 \def\@gls@text@#1#2[#3]{%
11754 \@gls@field@link{#1}{#2}{\glsfontaccessdisplay{\glsfonttext{#2}}{#2}}#3}%
11755 }

```

\@Gls@text@

```

11756 \def\@Gls@text@#1#2[#3]{%
11757 \@gls@field@link{#1}{#2}{\glsfontaccessdisplay{\Glsfonttext{#2}}{#2}}#3}%
11758 }

```

\@GLStext@

```

11759 \def\@GLStext@#1#2[#3]{%
11760 \@gls@field@link{#1}{#2}%
11761 {\glsfontaccessdisplay{\mfirstucMakeUppercase{\glsfonttext{#2}}{#2}}#3}%
11762 \mfirstucMakeUppercase{#3}}#3}%
11763 }

```

```

\@glsfirst@
11764 \def \@glsfirst@#1#2[#3]{%
11765   \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glentryfirst{#2}}{#2}#3}%
11766 }

\@Glsfirst@
11767 \def \@Glsfirst@#1#2[#3]{%
11768   \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
11769 }

\@GLSfirst@
11770 \def \@GLSfirst@#1#2[#3]{%
11771   \@gls@field@link{#1}{#2}%
11772   {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glentryfirst{#2}}}{#2}%
11773    \mfirstucMakeUppercase{#3}}%
11774 }

\@glsplural@
11775 \def \@glsplural@#1#2[#3]{%
11776   \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glentryplural{#2}}{#2}#3}%
11777 }

\@Glsplural@
11778 \def \@Glsplural@#1#2[#3]{%
11779   \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
11780 }

\@GLSplural@
11781 \def \@GLSplural@#1#2[#3]{%
11782   \@gls@field@link{#1}{#2}%
11783   {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glentryplural{#2}}}{#2}%
11784    \mfirstucMakeUppercase{#3}}%
11785 }

glsfirstplural@
11786 \def \@glsfirstplural@#1#2[#3]{%
11787   \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glentryfirstplural{#2}}{#2}#3}%
11788 }

Glsfirstplural@
11789 \def \@glsfirstplural@#1#2[#3]{%
11790   \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
11791 }

GLSfirstplural@
11792 \def \@GLSfirstplural@#1#2[#3]{%
11793   \@gls@field@link{#1}{#2}%
11794   {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glentryfirstplural{#2}}}{#2}%

```

```

11795     \mfirstucMakeUppercase{#3}}}%
11796 }

\@glsname@
11797 \def\@glsname@#1#2[#3]{%
11798   \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
11799 }

\@Glsname@
11800 \def\@Glsname@#1#2[#3]{%
11801   \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
11802 }

\@GLSname@
11803 \def\@GLSname@#1#2[#3]{%
11804   \@gls@field@link{#1}{#2}%
11805   {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
11806     \mfirstucMakeUppercase{#3}}}%
11807 }

\@glsdesc@
11808 \def\@glsdesc@#1#2[#3]{%
11809   \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
11810 }

\@Glsdesc@
11811 \def\@Glsdesc@#1#2[#3]{%
11812   \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
11813 }

\@GLSdesc@
11814 \def\@GLSdesc@#1#2[#3]{%
11815   \@gls@field@link{#1}{#2}%
11816   {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
11817     \mfirstucMakeUppercase{#3}}}%
11818 }

@glsdescplural@
11819 \def\@glsdescplural@#1#2[#3]{%
11820   \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
11821 }

@Glsdescplural@
11822 \def\@Glsdescplural@#1#2[#3]{%
11823   \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
11824 }

```

@GLSdescplural@

```
11825 \def\@GLSdescplural@#1#2[#3]{%
11826   \@gls@field@link{#1}{#2}%
11827   {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
11828     \mfirstucMakeUppercase{#3}}%
11829 }
```

\@glssymbol@

```
11830 \def\@glssymbol@#1#2[#3]{%
11831   \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
11832 }
```

\@GLssymbol@

```
11833 \def\@GLssymbol@#1#2[#3]{%
11834   \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
11835 }
```

\@GLSsymbol@

```
11836 \def\@GLSsymbol@#1#2[#3]{%
11837   \@gls@field@link{#1}{#2}%
11838   {\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
11839     \mfirstucMakeUppercase{#3}}%
11840 }
```

lssymbolplural@

```
11841 \def\@glssymbolplural@#1#2[#3]{%
11842   \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
11843 }
```

lssymbolplural@

```
11844 \def\@Glssymbolplural@#1#2[#3]{%
11845   \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
11846 }
```

Lssymbolplural@

```
11847 \def\@GLSsymbolplural@#1#2[#3]{%
11848   \@gls@field@link{#1}{#2}%
11849   {\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
11850     \mfirstucMakeUppercase{#3}}%
11851 }
```

\@glsuseri@

```
11852 \def\@glsuseri@#1#2[#3]{%
11853   \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
11854 }
```

\@Glsuseri@

```
11855 \def \@Glsuser@i#1#2[#3]{%
11856   \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
11857 }
```

\@GLSuseri@

```
11858 \def \@GLSuseri@#1#2[#3]{%
11859   \@gls@field@link{#1}{#2}%
11860   {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
11861     \mfirstucMakeUppercase{#3}}%
11862 }
```

\@glsuserii@

```
11863 \def \@glsuserii@#1#2[#3]{%
11864   \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
11865 }
```

\@Glsuserii@

```
11866 \def \@Glsuser@i#1#2[#3]{%
11867   \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
11868 }
```

\@GLSuserii@

```
11869 \def \@GLSuserii@#1#2[#3]{%
11870   \@gls@field@link{#1}{#2}%
11871   {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
11872     \mfirstucMakeUppercase{#3}}%
11873 }
```

\@glsuseriii@

```
11874 \def \@glsuseriii@#1#2[#3]{%
11875   \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
11876 }
```

\@Glsuseriii@

```
11877 \def \@Glsuser@i#1#2[#3]{%
11878   \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
11879 }
```

\@GLSuseriii@

```
11880 \def \@GLSuseriii@#1#2[#3]{%
11881   \@gls@field@link{#1}{#2}%
11882   {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
11883     \mfirstucMakeUppercase{#3}}%
11884 }
```



```

\@glsuseriv@
11885 \def \@glsuseriv@#1#2[#3]{%
11886   \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
11887 }

\@Glsuseriv@
11888 \def \@Glsuser@i#1#2[#3]{%
11889   \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
11890 }

\@GLSuseriv@
11891 \def \@GLSuseriv@#1#2[#3]{%
11892   \@gls@field@link{#1}{#2}%
11893   {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}{#2}%
11894     \mfirstucMakeUppercase{#3}}}%
11895 }

\@glsuserv@
11896 \def \@glsuserv@#1#2[#3]{%
11897   \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
11898 }

\@Glsuserv@
11899 \def \@Glsuser@i#1#2[#3]{%
11900   \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
11901 }

\@GLSuserv@
11902 \def \@GLSuserv@#1#2[#3]{%
11903   \@gls@field@link{#1}{#2}%
11904   {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}{#2}%
11905     \mfirstucMakeUppercase{#3}}}%
11906 }

\@glsuservi@
11907 \def \@glsuservi@#1#2[#3]{%
11908   \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
11909 }

\@Glsuservi@
11910 \def \@Glsuser@i#1#2[#3]{%
11911   \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
11912 }

\@GLSuservi@
11913 \def \@GLSuservi@#1#2[#3]{%
11914   \@gls@field@link{#1}{#2}%
11915   {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#2}}{#2}%
11916     \mfirstucMakeUppercase{#3}}}%
11917 }

```

## 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.

```
11918 \renewcommand*{\glossentryname}[1]{%
11919   \glsdoifexists{#1}%
11920   {%
11921     \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
11922   }%
11923 }

11924 \renewcommand*{\glossentryname}[1]{%
11925   \glsdoifexists{#1}%
11926   {%
11927     \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
11928   }%
11929 }

11930 \renewcommand*{\glossentrydesc}[1]{%
11931   \glsdoifexists{#1}%
11932   {%
11933     \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
11934   }%
11935 }

11936 \renewcommand*{\Glossentrydesc}[1]{%
11937   \glsdoifexists{#1}%
11938   {%
11939     \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
11940   }%
11941 }

11942 \renewcommand*{\glossentrysymbol}[1]{%
11943   \glsdoifexists{#1}%
11944   {%
11945     \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
11946   }%
11947 }

11948 \renewcommand*{\Glossentrysymbol}[1]{%
11949   \glsdoifexists{#1}%
11950   {%
11951     \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
11952   }%
11953 }
```

ssaryentryfield

```

11954 \newcommand*{\accsuppglossaryentryfield}[5]{%
11955   \glossaryentryfield{#1}%
11956   {\glslnameaccessdisplay{#2}{#1}}%
11957   {\glsldescriptionaccessdisplay{#3}{#1}}%
11958   {\glssymbolaccessdisplay{#4}{#1}}{#5}%
11959 }

```

rysubentryfield

```

11960 \newcommand*{\accsuppglossarysubentryfield}[6]{%
11961   \glossarysubentryfield{#1}{#2}%
11962   {\glslnameaccessdisplay{#3}{#2}}%
11963   {\glsldescriptionaccessdisplay{#4}{#2}}%
11964   {\glssymbolaccessdisplay{#5}{#2}}{#6}%
11965 }

```

## 5.4 Acronyms

Redefine acronym styles provided by glossaries:

long-short    *<long>* (*<short>*) acronym style.

```

11966 \renewacronymstyle{long-short}%
11967 {%

```

Check for long form in case this is a mixed glossary.

```

11968   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
11969 }%
11970 {%
11971   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
11972   \renewcommand*{\genacrfullformat}[2]{%
11973     \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
11974     (\glsshortaccessdisplay
11975       {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
11976   }%
11977   \renewcommand*{\Genacrfullformat}[2]{%
11978     \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
11979     (\glsshortaccessdisplay
11980       {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
11981   }%
11982   \renewcommand*{\genplacrfullformat}[2]{%
11983     \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
11984     (\glsshortpluralaccessdisplay
11985       {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
11986   }%
11987   \renewcommand*{\Genplacrfullformat}[2]{%
11988     \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
11989     (\glsshortpluralaccessdisplay
11990       {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
11991   }%
11992   \renewcommand*{\acronymentry}[1]{%

```

```

11993 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}%
11994 \renewcommand*{\acronymsort}[2]{##1}%
11995 \renewcommand*{\acronymfont}[1]{##1}%
11996 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
11997 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
11998 }

```

short-long <short> (<long>) acronym style.

```

11999 \renewacronymstyle{short-long}%
12000 {%

```

Check for long form in case this is a mixed glossary.

```

12001 \ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
12002 }%
12003 {%
12004 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
12005 \renewcommand*{\genacrfullformat}[2]{%
12006 \glsshortaccessdisplay
12007 {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
12008 (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
12009 }%
12010 \renewcommand*{\Genacrfullformat}[2]{%
12011 \glsshortaccessdisplay
12012 {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
12013 (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
12014 }%
12015 \renewcommand*{\genplacrfullformat}[2]{%
12016 \glsshortpluralaccessdisplay
12017 {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
12018 (\glslongpluralaccessdisplay
12019 {\glentrylongpl{##1}}{##1})%
12020 }%
12021 \renewcommand*{\Genplacrfullformat}[2]{%
12022 \glsshortpluralaccessdisplay
12023 {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
12024 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
12025 }%
12026 \renewcommand*{\acronymentry}[1]{%
12027 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}%
12028 \renewcommand*{\acronymsort}[2]{##1}%
12029 \renewcommand*{\acronymfont}[1]{##1}%
12030 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
12031 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
12032 }

```

long-short-desc <long> ({<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12033 \renewacronymstyle{long-short-desc}%
12034 {%

```

```

12035 \GlsUseAcrEntryDispStyle{long-short}%
12036 }%
12037 {%
12038 \GlsUseAcrStyleDefs{long-short}%
12039 \renewcommand*{\GenericAcronymFields}{}%
12040 \renewcommand*{\acronymsort}[2]{##2}%
12041 \renewcommand*{\acronymentry}[1]{%
12042 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
12043 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1))}%
12044 }

```

g-sc-short-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12045 \renewacronymstyle{long-sc-short-desc}%
12046 {%
12047 \GlsUseAcrEntryDispStyle{long-sc-short}%
12048 }%
12049 {%
12050 \GlsUseAcrStyleDefs{long-sc-short}%
12051 \renewcommand*{\GenericAcronymFields}{}%
12052 \renewcommand*{\acronymsort}[2]{##2}%
12053 \renewcommand*{\acronymentry}[1]{%
12054 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
12055 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1))}%
12056 }

```

g-sm-short-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12057 \renewacronymstyle{long-sm-short-desc}%
12058 {%
12059 \GlsUseAcrEntryDispStyle{long-sm-short}%
12060 }%
12061 {%
12062 \GlsUseAcrStyleDefs{long-sm-short}%
12063 \renewcommand*{\GenericAcronymFields}{}%
12064 \renewcommand*{\acronymsort}[2]{##2}%
12065 \renewcommand*{\acronymentry}[1]{%
12066 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
12067 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1))}%
12068 }

```

short-long-desc *<short>* ({<long>}) acronym style that has an accompanying description (which the user needs to supply).

```

12069 \renewacronymstyle{short-long-desc}%
12070 {%
12071 \GlsUseAcrEntryDispStyle{short-long}%
12072 }%
12073 {%
12074 \GlsUseAcrStyleDefs{short-long}%

```

```

12075 \renewcommand*{\GenericAcronymFields}{}%
12076 \renewcommand*{\acronymsort}[2]{##2}%
12077 \renewcommand*{\acronymentry}[1]{%
12078     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12079     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
12080 }

```

short-long-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12081 \renewacronymstyle{sc-short-long-desc}%
12082 {%
12083     \GlsUseAcrEntryDispStyle{sc-short-long}%
12084 }%
12085 {%
12086     \GlsUseAcrStyleDefs{sc-short-long}%
12087     \renewcommand*{\GenericAcronymFields}{}%
12088     \renewcommand*{\acronymsort}[2]{##2}%
12089     \renewcommand*{\acronymentry}[1]{%
12090         \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12091         (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
12092     }

```

short-long-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12093 \renewacronymstyle{sm-short-long-desc}%
12094 {%
12095     \GlsUseAcrEntryDispStyle{sm-short-long}%
12096 }%
12097 {%
12098     \GlsUseAcrStyleDefs{sm-short-long}%
12099     \renewcommand*{\GenericAcronymFields}{}%
12100     \renewcommand*{\acronymsort}[2]{##2}%
12101     \renewcommand*{\acronymentry}[1]{%
12102         \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12103         (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
12104     }

```

dua *<long>* only acronym style.

```

12105 \renewacronymstyle{dua}%
12106 {%

```

Check for long form in case this is a mixed glossary.

```

12107 \ifdefempty\glscustomtext
12108 {%
12109     \ifglshaslong{\glslabel}%
12110     {%
12111         \glsifplural
12112         {%

```

Plural form:

12113           \glscapscase  
12114           {%

Plural form, don't adjust case:

12115           \glslongpluralaccessdisplay{\glentrylongpl{\glslabel}}{\glslabel}%  
12116           \glsinsert  
12117           }%  
12118           {%

Plural form, make first letter upper case:

12119           \glslongpluralaccessdisplay{\Glentrylongpl{\glslabel}}{\glslabel}%  
12120           \glsinsert  
12121           }%  
12122           {%

Plural form, all caps:

12123           \glslongpluralaccessdisplay  
12124           {\mfirstucMakeUppercase{\glentrylongpl{\glslabel}}}{\glslabel}%  
12125           \mfirstucMakeUppercase{\glsinsert}%  
12126           }%  
12127           }%  
12128           {%

Singular form

12129           \glscapscase  
12130           {%

Singular form, don't adjust case:

12131           \glslongaccessdisplay{\glentrylong{\glslabel}}{\glslabel}\glsinsert  
12132           }%  
12133           {%

Subsequent singular form, make first letter upper case:

12134           \glslongaccessdisplay{\Glentrylong{\glslabel}}{\glslabel}\glsinsert  
12135           }%  
12136           {%

Subsequent singular form, all caps:

12137           \glslongaccessdisplay  
12138           {\mfirstucMakeUppercase  
12139           {\glentrylong{\glslabel}\glsinsert}}{\glslabel}%  
12140           \mfirstucMakeUppercase{\glsinsert}%  
12141           }%  
12142           }%  
12143           }%  
12144           {%

Not an acronym:

12145           \glsgenentryfmt  
12146           }%  
12147           }%

```

12148 {\glscustomtext\glsinsert}%
12149 }%
12150 {%
12151 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
12152 \renewcommand*{\acrfullfmt}[3]{%
12153   \glslink[##1]{##2}{%
12154     \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
12155     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}}}%
12156 \renewcommand*{\Acrfullfmt}[3]{%
12157   \glslink[##1]{##2}{%
12158     \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
12159     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}}}%
12160 \renewcommand*{\ACRfullfmt}[3]{%
12161   \glslink[##1]{##2}{%
12162     \glslongaccessdisplay
12163     {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
12164     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}}}%
12165 \renewcommand*{\acrfullplfmt}[3]{%
12166   \glslink[##1]{##2}{%
12167     \glslongpluralaccessdisplay
12168     {\glsentrylongpl{##2}}{##2}##3\space
12169     (\glsshortpluralaccessdisplay
12170     {\acronymfont{\glsentryshortpl{##2}}}{##2}}}%
12171 \renewcommand*{\Acrfullplfmt}[3]{%
12172   \glslink[##1]{##2}{%
12173     \glslongpluralaccessdisplay
12174     {\Glsentrylongpl{##2}}{##2}##3\space
12175     (\glsshortpluralaccessdisplay
12176     {\acronymfont{\glsentryshortpl{##2}}}{##2}}}%
12177 \renewcommand*{\ACRfullplfmt}[3]{%
12178   \glslink[##1]{##2}{%
12179     \glslongpluralaccessdisplay
12180     {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
12181     (\glsshortpluralaccessdisplay
12182     {\acronymfont{\glsentryshortpl{##2}}}{##2}}}%
12183 \renewcommand*{\glsentryfull}[1]{%
12184   \glslongaccessdisplay{\glsentrylong{##1}}\space
12185   (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
12186 }%
12187 \renewcommand*{\Glsentryfull}[1]{%
12188   \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
12189   (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
12190 }%
12191 \renewcommand*{\glsentryfullpl}[1]{%
12192   \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
12193   (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
12194 }%
12195 \renewcommand*{\Glsentryfullpl}[1]{%
12196   \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space

```



```

12197 (\glsshortpluralaccessdisplay{\acronymfont{\glentryshortpl{##1}}}{##1})%
12198 }%
12199 \renewcommand*{\acronymentry}[1]{%
12200 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
12201 \renewcommand*{\acronymsort}[2]{##1}%
12202 \renewcommand*{\acronymfont}[1]{##1}%
12203 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
12204 }

```

dua-desc *<long>* only acronym style with user-supplied description.

```

12205 \renewacronymstyle{dua-desc}%
12206 {%
12207 \GlsUseAcrEntryDispStyle{dua}%
12208 }%
12209 {%
12210 \GlsUseAcrStyleDefs{dua}%
12211 \renewcommand*{\GenericAcronymFields}{}%
12212 \renewcommand*{\acronymentry}[1]{%
12213 \glslongaccessdisplay{\acronymfont{\glentrylong{##1}}}{##1}}%
12214 \renewcommand*{\acronymsort}[2]{##2}%
12215 }%

```

footnote *<short>*\footnote{*<long>*} acronym style.

```

12216 \renewacronymstyle{footnote}%
12217 {%
12218 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
12219 }%
12220 {%
12221 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

```

Need to ensure hyperlinks are switched off on first use:

```

12222 \glshyperfirstfalse
12223 \renewcommand*{\genacrfullformat}[2]{%
12224 \glsshortaccessdisplay
12225 {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2%
12226 \protect\footnote{\glslongaccessdisplay{\glentrylong{##1}}}{##1}}%
12227 }%
12228 \renewcommand*{\Genacrfullformat}[2]{%
12229 \glsshortaccessdisplay
12230 {\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
12231 \protect\footnote{\glslongaccessdisplay{\glentrylong{##1}}}{##1}}%
12232 }%
12233 \renewcommand*{\genplacrfullformat}[2]{%
12234 \glsshortpluralaccessdisplay
12235 {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2%
12236 \protect\footnote{\glslongpluralaccessdisplay{\glentrylongpl{##1}}}{##1}}%
12237 }%
12238 \renewcommand*{\Genplacrfullformat}[2]{%

```

```

12239 \glsshortpluralaccessdisplay
12240 {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
12241 \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
12242 }%
12243 \renewcommand*{\acronymentry}[1]{%
12244 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
12245 \renewcommand*{\acronymsort}[2]{##1}%
12246 \renewcommand*{\acronymfont}[1]{##1}%
12247 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%

```

Don't use footnotes for \acrfull:

```

12248 \renewcommand*{\acrfullfmt}[3]{%
12249 \glslink[##1]{##2}{%
12250 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
12251 (\glslongaccessdisplay{\glsentrylong{##2}}{##2}}}%
12252 \renewcommand*{\Acrfullfmt}[3]{%
12253 \glslink[##1]{##2}{%
12254 \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
12255 (\glslongaccessdisplay{\glsentrylong{##2}}{##2}}}%
12256 \renewcommand*{\ACRfullfmt}[3]{%
12257 \glslink[##1]{##2}{%
12258 \glsshortaccessdisplay
12259 {\mfirstucMakeUppercase
12260 {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
12261 (\glslongaccessdisplay{\glsentrylong{##2}}{##2}}}%
12262 \renewcommand*{\acrfullplfmt}[3]{%
12263 \glslink[##1]{##2}{%
12264 \glsshortpluralaccessdisplay
12265 {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
12266 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}}}%
12267 \renewcommand*{\ACRfullplfmt}[3]{%
12268 \glslink[##1]{##2}{%
12269 \glsshortpluralaccessdisplay
12270 {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
12271 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}}}%
12272 \renewcommand*{\ACRfullplplfmt}[3]{%
12273 \glslink[##1]{##2}{%
12274 \glsshortpluralaccessdisplay
12275 {\mfirstucMakeUppercase
12276 {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
12277 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}}}%

```

Similarly for \glsentryfull etc:

```

12278 \renewcommand*{\glsentryfull}[1]{%
12279 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
12280 (\glslongaccessdisplay{\glsentrylong{##1}}{##1}}}%
12281 \renewcommand*{\Glsentryfull}[1]{%
12282 \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
12283 (\glslongaccessdisplay{\glsentrylong{##1}}{##1}}}%
12284 \renewcommand*{\glsentryfullpl}[1]{%

```

```

12285 \glsshortpluralaccessdisplay
12286 {\acronymfont{\glentryshortpl{##1}}{##1}\space
12287 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}}%
12288 \renewcommand*{\Glsentryfullpl}[1]{%
12289 \glsshortpluralaccessdisplay
12290 {\acronymfont{\Glsentryshortpl{##1}}{##1}\space
12291 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}}%
12292 }

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

12293 \renewacronymstyle{footnote-sc}%
12294 {%
12295 \GlsUseAcrEntryDisplayStyle{footnote}%
12296 }%
12297 {%
12298 \GlsUseAcrStyleDefs{footnote}%
12299 \renewcommand{\acronymentry}[1]{%
12300 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}
12301 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
12302 \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
12303 }%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```

12304 \renewacronymstyle{footnote-sm}%
12305 {%
12306 \GlsUseAcrEntryDisplayStyle{footnote}%
12307 }%
12308 {%
12309 \GlsUseAcrStyleDefs{footnote}%
12310 \renewcommand{\acronymentry}[1]{%
12311 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}
12312 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
12313 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
12314 }%

```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

12315 \renewacronymstyle{footnote-desc}%
12316 {%
12317 \GlsUseAcrEntryDisplayStyle{footnote}%
12318 }%
12319 {%
12320 \GlsUseAcrStyleDefs{footnote}%
12321 \renewcommand*{\GenericAcronymFields}{}%
12322 \renewcommand*{\acronymsort}[2]{##2}%
12323 \renewcommand*{\acronymentry}[1]{%
12324 \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
12325 (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}}%
12326 }

```

ootnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
12327 \renewacronymstyle{footnote-sc-desc}%
12328 {%
12329   \GlsUseAcrEntryDispStyle{footnote-sc}%
12330 }%
12331 {%
12332   \GlsUseAcrStyleDefs{footnote-sc}%
12333   \renewcommand*{\GenericAcronymFields}{}%
12334   \renewcommand*{\acronymsort}[2]{##2}%
12335   \renewcommand*{\acronymentry}[1]{%
12336     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12337     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1)}}%
12338 }
```

ootnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
12339 \renewacronymstyle{footnote-sm-desc}%
12340 {%
12341   \GlsUseAcrEntryDispStyle{footnote-sm}%
12342 }%
12343 {%
12344   \GlsUseAcrStyleDefs{footnote-sm}%
12345   \renewcommand*{\GenericAcronymFields}{}%
12346   \renewcommand*{\acronymsort}[2]{##2}%
12347   \renewcommand*{\acronymentry}[1]{%
12348     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12349     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1)}}%
12350 }
```

aultshortaccess \glsdefaultshortaccess{<long>}{<short>}

Default shortaccess value.

```
12351 \newcommand*{\glsdefaultshortaccess}[2]{#1}
```

Use \newacronymhook to modify the key list to set the access text to the long version by default.

```
12352 \renewcommand*{\newacronymhook}{%
12353   \protected@edef\@gls@keylist{%
12354     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12355     shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12356     \the\glskeylisttok}%
12357   \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
12358 }
```

ltNewAcronymDef Modify default style to use access text:

```
12359 \renewcommand*{\DefaultNewAcronymDef}{}%
```

```

12360 \edef\@do@newglossaryentry{%
12361   \noexpand\newglossaryentry{\the\glslabeltok}%
12362   {%
12363     type=\acronymtype,%
12364     name={\the\glsshorttok},%
12365     description={\the\glslongtok},%
12366     descriptionaccess=\relax,
12367     text={\the\glsshorttok},%
12368     access={\noexpand\@glo@textaccess},%
12369     sort={\the\glsshorttok},%
12370     short={\the\glsshorttok},%
12371     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12372     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12373     long={\the\glslongtok},%
12374     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12375     descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12376     first={\noexpand\glslongaccessdisplay
12377       {\the\glslongtok}{\the\glslabeltok}\space
12378       (\noexpand\glsshortaccessdisplay
12379         {\the\glsshorttok}{\the\glslabeltok})},%
12380     plural={\the\glsshorttok\acrpluralsuffix},%
12381     firstplural={\noexpand\glslongpluralaccessdisplay
12382       {\noexpand\@glo@longpl}{\the\glslabeltok}\space
12383       (\noexpand\glsshortpluralaccessdisplay
12384         {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
12385     firstaccess=\relax,
12386     firstpluralaccess=\relax,
12387     textaccess={\noexpand\@glo@shortaccess},%
12388     \the\glskeylisttok
12389   }%
12390 }%
12391 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12392 \let\@org@gls@assign@plural\gls@assign@plural
12393 \let\@org@gls@assign@descplural\gls@assign@descplural
12394 \def\gls@assign@firstpl##1##2{%
12395   \@gls@expand@field{##1}{firstpl}{##2}%
12396 }%
12397 \def\gls@assign@plural##1##2{%
12398   \@gls@expand@field{##1}{plural}{##2}%
12399 }%
12400 \def\gls@assign@descplural##1##2{%
12401   \@gls@expand@field{##1}{descplural}{##2}%
12402 }%
12403 \@do@newglossaryentry
12404 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12405 \let\gls@assign@plural\@org@gls@assign@plural
12406 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12407 \@gls@noexpand@assign@default@plurals
12408 }

```

teNewAcronymDef

```
12409 \renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
12410   \edef\@do@newglossaryentry{%
12411     \noexpand\newglossaryentry{\the\glslabeltok}%
12412     {%
12413       type=\acronymtype,%
12414       name={\noexpand\acronymfont{\the\glsshorttok}},%
12415       sort={\the\glsshorttok},%
12416       text={\the\glsshorttok},%
12417       short={\the\glsshorttok},%
12418       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12419       shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12420       long={\the\glslongtok},%
12421       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12422       access={\noexpand\@glo@textaccess},%
12423       plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12424       symbol={\the\glslongtok},%
12425       symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12426       firstpluralaccess=\relax,
12427       textaccess={\noexpand\@glo@shortaccess},%
12428       \the\glskeylisttok
12429     }%
12430   }%
12431   \let\@org@gls@assign@firstpl\gls@assign@firstpl
12432   \let\@org@gls@assign@plural\gls@assign@plural
12433   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12434   \def\gls@assign@firstpl##1##2{%
12435     \@@gls@expand@field{##1}{firstpl}{##2}%
12436   }%
12437   \def\gls@assign@plural##1##2{%
12438     \@@gls@expand@field{##1}{plural}{##2}%
12439   }%
12440   \def\gls@assign@symbolplural##1##2{%
12441     \@@gls@expand@field{##1}{symbolplural}{##2}%
12442   }%
12443   \@do@newglossaryentry
12444   \let\gls@assign@plural\@org@gls@assign@plural
12445   \let\gls@assign@firstpl\@org@gls@assign@firstpl
12446   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12447 }
```

onNewAcronymDef

```
12448 \renewcommand*{\DescriptionNewAcronymDef}{%
12449   \edef\@do@newglossaryentry{%
12450     \noexpand\newglossaryentry{\the\glslabeltok}%
12451     {%
12452       type=\acronymtype,%
12453       name={\noexpand
12454         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
```

```

12455     access={\noexpand\@glo@textaccess},%
12456     sort={\the\glsshorttok},%
12457     short={\the\glsshorttok},%
12458     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12459     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12460     long={\the\glslongtok},%
12461     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12462     first={\the\glslongtok},%
12463     firstaccess=\relax,
12464     firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12465     text={\the\glsshorttok},%
12466     textaccess={\the\glslongtok},%
12467     plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12468     symbol={\noexpand\@glo@text},%
12469     symbolaccess={\noexpand\@glo@textaccess},%
12470     symbolplural={\noexpand\@glo@plural},%
12471     firstpluralaccess=\relax,
12472     textaccess={\noexpand\@glo@shortaccess},%
12473     \the\glskeylisttok}%
12474 }%
12475 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12476 \let\@org@gls@assign@plural\gls@assign@plural
12477 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12478 \def\gls@assign@firstpl##1##2{%
12479   \@gls@expand@field{##1}{firstpl}{##2}%
12480 }%
12481 \def\gls@assign@plural##1##2{%
12482   \@gls@expand@field{##1}{plural}{##2}%
12483 }%
12484 \def\gls@assign@symbolplural##1##2{%
12485   \@gls@expand@field{##1}{symbolplural}{##2}%
12486 }%
12487 \do@newglossaryentry
12488 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12489 \let\gls@assign@plural\@org@gls@assign@plural
12490 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12491 }

```

teNewAcronymDef

```

12492 \renewcommand*{\FootnoteNewAcronymDef}{%
12493   \edef\@do@newglossaryentry{%
12494     \noexpand\newglossaryentry{\the\glslabeltok}%
12495     {%
12496       type=\acronymtype,%
12497       name={\noexpand\acronymfont{\the\glsshorttok}},%
12498       sort={\the\glsshorttok},%
12499       text={\the\glsshorttok},%
12500       textaccess={\the\glslongtok},%
12501       access={\noexpand\@glo@textaccess},%

```

```

12502 plural={\the\glssshorttok\noexpand\acrpluralsuffix},%
12503 short={\the\glssshorttok},%
12504 shortplural={\the\glssshorttok\noexpand\acrpluralsuffix},%
12505 long={\the\glslongtok},%
12506 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12507 description={\the\glslongtok},%
12508 descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12509 \the\glskeylisttok
12510 }%
12511 }%
12512 \let\@org@gls@assign@plural\gls@assign@plural
12513 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12514 \let\@org@gls@assign@descplural\gls@assign@descplural
12515 \def\gls@assign@firstpl##1##2{%
12516   \@gls@expand@field{##1}{firstpl}{##2}%
12517 }%
12518 \def\gls@assign@plural##1##2{%
12519   \@gls@expand@field{##1}{plural}{##2}%
12520 }%
12521 \def\gls@assign@descplural##1##2{%
12522   \@gls@expand@field{##1}{descplural}{##2}%
12523 }%
12524 \do@newglossaryentry
12525 \let\gls@assign@plural\@org@gls@assign@plural
12526 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12527 \let\gls@assign@descplural\@org@gls@assign@descplural
12528 }

```

# 11NewAcronymDef

```

12529 \renewcommand*{\SmallNewAcronymDef}{%
12530   \edef\@do@newglossaryentry{%
12531     \noexpand\newglossaryentry{\the\glslabeltok}%
12532     {%
12533       type=\acronymtype,%
12534       name={\noexpand\acronymfont{\the\glssshorttok}},%
12535       access={\noexpand\@glo@symbolaccess},%
12536       sort={\the\glssshorttok},%
12537       short={\the\glssshorttok},%
12538       shortplural={\the\glssshorttok\noexpand\acrpluralsuffix},%
12539       shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glssshorttok}},%
12540       long={\the\glslongtok},%
12541       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12542       text={\noexpand\@glo@short},%
12543       textaccess={\noexpand\@glo@shortaccess},%
12544       plural={\noexpand\@glo@shortpl},%
12545       first={\the\glslongtok},%
12546       firstaccess=\relax,%
12547       firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12548       description={\noexpand\@glo@first},%

```



```

12549     descriptionplural={\noexpand\@glo@firstplural},%
12550     symbol={\the\glsshorttok},%
12551     symbolaccess={\the\glslongtok},%
12552     symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12553     \the\glskeylisttok
12554   }%
12555 }%
12556 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12557 \let\@org@gls@assign@plural\gls@assign@plural
12558 \let\@org@gls@assign@descplural\gls@assign@descplural
12559 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12560 \def\gls@assign@firstpl##1##2{%
12561   \@@gls@expand@field{##1}{firstpl}{##2}%
12562 }%
12563 \def\gls@assign@plural##1##2{%
12564   \@@gls@expand@field{##1}{plural}{##2}%
12565 }%
12566 \def\gls@assign@descplural##1##2{%
12567   \@@gls@expand@field{##1}{descplural}{##2}%
12568 }%
12569 \def\gls@assign@symbolplural##1##2{%
12570   \@@gls@expand@field{##1}{symbolplural}{##2}%
12571 }%
12572 \do@newglossaryentry
12573 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12574 \let\gls@assign@plural\@org@gls@assign@plural
12575 \let\gls@assign@descplural\@org@gls@assign@descplural
12576 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12577 }

```

The following are kept for compatibility with versions before 3.0:

sshortaccesskey

```
12578 \newcommand*{\glsshortaccesskey}{\glsshortkey access}%
```

pluralaccesskey

```
12579 \newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
```

lslongaccesskey

```
12580 \newcommand*{\glslongaccesskey}{\glslongkey access}%
```

pluralaccesskey

```
12581 \newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

## 5.5 Debugging Commands

owglongnameaccess

```
12582 \newcommand*{\showglongnameaccess}[1]{%
```

```

12583 \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
12584 }

```

owglotextaccess

```

12585 \newcommand*{\showglotextaccess}[1]{%
12586 \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
12587 }

```

glopluralaccess

```

12588 \newcommand*{\showglopluralaccess}[1]{%
12589 \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
12590 }

```

wglofirstaccess

```

12591 \newcommand*{\showglofirstaccess}[1]{%
12592 \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
12593 }

```

rstpluralaccess

```

12594 \newcommand*{\showglofirstpluralaccess}[1]{%
12595 \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
12596 }

```

glosymbolaccess

```

12597 \newcommand*{\showglosymbolaccess}[1]{%
12598 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
12599 }

```

bolpluralaccess

```

12600 \newcommand*{\showglosymbolpluralaccess}[1]{%
12601 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
12602 }

```

owglodescaccess

```

12603 \newcommand*{\showglodescaccess}[1]{%
12604 \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
12605 }

```

escpluralaccess

```

12606 \newcommand*{\showglodescpluralaccess}[1]{%
12607 \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
12608 }

```

wgloshortaccess

```

12609 \newcommand*{\showgloshortaccess}[1]{%
12610 \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
12611 }

```

ortpluralaccess

```
12612 \newcommand*{\showgloshortpluralaccess}[1]{%  
12613   \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname  
12614 }
```

owglolongaccess

```
12615 \newcommand*{\showglolongaccess}[1]{%  
12616   \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname  
12617 }
```

ongpluralaccess

```
12618 \newcommand*{\showglolongpluralaccess}[1]{%  
12619   \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname  
12620 }
```

## 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.

```
12621 \NeedsTeXFormat{LaTeX2e}
12622 \ProvidesPackage{glossaries-babel}[2021/11/01 v4.49 (NLCT)]
```

Load tracklang to obtain language settings.

```
12623 \RequirePackage{tracklang}
12624 \let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
12625 \AnyTrackedLanguages
12626 {%
12627   \ForEachTrackedDialect{\this@dialect}{%
12628     \IfTrackedLanguageFileExists{\this@dialect}%
12629     {glossaries-}% prefix
12630     {.ldf}%
12631     {%
12632       \RequireGlossariesLang{\CurrentTrackedTag}%
12633     }%
12634     {%
12635       \PackageWarningNoLine{glossaries}%
12636       {No language module detected for ‘\this@dialect’.\MessageBreak
12637       Language modules need to be installed separately.\MessageBreak
12638       Please check on CTAN for a bundle called\MessageBreak
12639       ‘glossaries-\CurrentTrackedLanguage’ or similar}%
12640     }%
12641   }%
12642 }%
12643 {}%
```

### 6.1 Polyglossia Captions

Language support has now been split off into independent language modules.

```
12644 \NeedsTeXFormat{LaTeX2e}
12645 \ProvidesPackage{glossaries-polyglossia}[2021/11/01 v4.49 (NLCT)]
```

Load tracklang to obtain language settings.

```
12646 \RequirePackage{tracklang}
12647 \let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
12648 \AnyTrackedLanguages
```

```

12649 {%
12650   \ForEachTrackedDialect{\this@dialect}{%
12651     \IfTrackedLanguageFileExists{\this@dialect}%
12652     {glossaries-}% prefix
12653     {.ldf}%
12654     {%
12655       \RequireGlossariesLang{\CurrentTrackedTag}%
12656     }%
12657     {%
12658       \PackageWarningNoLine{glossaries}%
12659       {No language module detected for ‘\this@dialect’.\MessageBreak
12660       Language modules need to be installed separately.\MessageBreak
12661       Please check on CTAN for a bundle called\MessageBreak
12662       ‘glossaries-\CurrentTrackedLanguage’ or similar}%
12663     }%
12664   }%
12665 }%
12666 {}%

```

## 6.2 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

12667 \NeedsTeXFormat{LaTeX2e}
12668 \ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]
12669 \RequirePackage{ifthen}
12670 \RequirePackage{xkeyval}[2006/11/18]
12671 \RequirePackage{mfirstuc}
12672 \RequirePackage{textcase}
12673 \renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
12674 \RequirePackage{xfor}
12675 \RequirePackage{datatool-base}
12676 \RequirePackage{amsgen}
12677 \RequirePackage{etoolbox}
12678 \newif\if@gls@docloaded
12679 \@ifpackageloaded{doc}%
12680 {%
12681   \@gls@docloadedtrue
12682 }%
12683 {%
12684   \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
12685 }
12686 \if@gls@docloaded
12687   \let\glsorg@theglossary\theglossary
12688   \let\glsorg@endtheglossary\endtheglossary
12689   \let\glsorg@PrintChanges\PrintChanges
12690   \renewcommand{\PrintChanges}{%
12691     \begingroup
12692     \let\theglossary\glsorg@theglossary

```

```

12693     \let\endtheglossary\glsorg@endtheglossary
12694     \glsorg@PrintChanges
12695 \endgroup
12696 }
12697 \fi
12698 \newif\if@gls@debug
12699 \define@choicekey{glossaries.sty}{debug}{\@gls@debug@val\@gls@debug@nr}%
12700 {true,false,showtargets,showaccsupp}[true]{%
12701   \ifcase\@gls@debug@nr\relax
12702     % debug=true
12703     \@gls@debugtrue
12704     \renewcommand*\GlossariesWarning[1]{%
12705       \PackageWarning{glossaries}{##1}%
12706     }%
12707     \renewcommand*\GlossariesWarningNoLine[1]{%
12708       \PackageWarningNoLine{glossaries}{##1}%
12709     }%
12710     \let\@glsshowtarget\@gobble
12711     \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12712   \or
12713     % debug=false
12714     \@gls@debugfalse
12715     \let\@glsshowtarget\@gobble
12716     \let\@glsshowaccsupp\@gobblethree
12717     \PackageInfo{glossaries}{debug mode OFF}%
12718   \or
12719     % debug=showtargets
12720     \@gls@debugtrue
12721     \renewcommand*\GlossariesWarning[1]{%
12722       \PackageWarning{glossaries}{##1}%
12723     }%
12724     \renewcommand*\GlossariesWarningNoLine[1]{%
12725       \PackageWarningNoLine{glossaries}{##1}%
12726     }%
12727     \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12728     \renewcommand{\@glsshowtarget}{\@glsshowtarget}%
12729   \or
12730     % debug=showaccsupp
12731     \@gls@debugtrue
12732     \renewcommand*\GlossariesWarning[1]{%
12733       \PackageWarning{glossaries}{##1}%
12734     }%
12735     \renewcommand*\GlossariesWarningNoLine[1]{%
12736       \PackageWarningNoLine{glossaries}{##1}%
12737     }%
12738     \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12739     \renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
12740   \fi
12741 }

```

```

12742 \newcommand*{\glsshowtarget}[1]{%
12743   \ifmode
12744     \nfss@text{\glsshowtargetfont [#1]}%
12745   \else
12746     \ifinner
12747       {\glsshowtargetfont [#1]}%
12748     \else
12749       \glsshowtargetouter{#1}%
12750     \fi
12751   \fi
12752 }
12753 \newcommand*{\glsshowtargetouter}[1]{%
12754   \glsshowtargetsymbols\marginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
12755 \newcommand*{\glsshowtargetsymbols}{\tiny$\triangleright$}
12756 \newcommand*{\glsshowtargetfont}{\ttfamily\footnotesize}
12757 \newcommand*{\@glsshowtarget}[1]{%
12758   \newrobustcmd*{\@glsshowtarget}[1]{%
12759     \begingroup
12760     \protected@edef\@gls@tmp{#1}%
12761     \@onelevel@sanitize\@gls@tmp
12762     \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
12763   \endgroup
12764 }
12765 \newcommand*{\@glsshowaccsupp}[3]{%
12766   \newrobustcmd*{\glsshowaccsupp}[3]{%
12767     \ifstrepty{#1}%
12768     {\@glsshowtarget{/#2 (#3)}}%
12769     {\@glsshowtarget{/#2 (#3) [#1]}}%
12770 }
12771 \newcommand*{\@gls@see@noindex}{%
12772   \PackageError{glossaries}%
12773   {'\gls@xr@key' key may only be used after \string\makeglossaries\space
12774   or \string\makenoidxglossaries\space (or move
12775   \string\newglossaryentry\space
12776   definitions into the preamble)}}%
12777   {You must use \string\makeglossaries\space
12778   or \string\makenoidxglossaries\space before defining
12779   any entries that have a '\gls@xr@key' key. It may
12780   be that the 'see' key has been written to the .glsdefs
12781   file from the previous run, in which case you need to
12782   move your definitions
12783   to the preamble if you don't want to use
12784   \string\makeglossaries\space
12785   or \string\makenoidxglossaries}%
12786 }
12787 \define@choicekey{glossaries.sty}{seenindex}%
12788   [\gls@seenindex@val\gls@seenindex@nr]{error,warn,ignore}{%
12789   \ifcase\gls@seenindex@nr
12790     \renewcommand*{\@gls@see@noindex}{%

```

```

12791 \PackageError{glossaries}%
12792 {'\gls@xr@key' key may only be used after \string\makeglossaries\space
12793 or \string\makenoidxglossaries}%
12794 {You must use \string\makeglossaries\space
12795 or \string\makenoidxglossaries\space before defining
12796 any entries that have a '\gls@xr@key' key}%
12797 }%
12798 \or
12799 \renewcommand*{\@gls@see@noindex}{%
12800 \GlossariesWarning{'\gls@xr@key' key ignored}%
12801 }%
12802 \or
12803 \renewcommand*{\@gls@see@noindex}{}%
12804 \fi
12805 }
12806 \define@boolkey{glossaries.sty}[gls]{toc}[true]{}
12807 \define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
12808 \ifcsundef{chapter}%
12809 {\newcommand*{\@glossarysec}{section}}%
12810 {\newcommand*{\@glossarysec}{chapter}}
12811 \define@choicekey{glossaries.sty}{section}{part,chapter,section,%
12812 subsection,subsubsection,paragraph,subparagraph}[section]{}
12813 \renewcommand*{\@glossarysec}{#1}}
12814 \newcommand*{\@glossarysecstar}{*}%
12815 \newcommand*{\@glossaryseclabel}{}%
12816 \newcommand*{\glsautoprefix}{}%
12817 \define@choicekey{glossaries.sty}{numberedsection}%
12818 [\gls@numberedsection@val\gls@numberedsection@nr]{}%
12819 false,nolabel,autolabel,nameref}[nolabel]{}%
12820 \ifcase\gls@numberedsection@nr\relax
12821 \renewcommand*{\@glossarysecstar}{*}%
12822 \renewcommand*{\@glossaryseclabel}{}%
12823 \or
12824 \renewcommand*{\@glossarysecstar}{}%
12825 \renewcommand*{\@glossaryseclabel}{}%
12826 \or
12827 \renewcommand*{\@glossarysecstar}{}%
12828 \renewcommand*{\@glossaryseclabel}{%
12829 \label{\glsautoprefix\@glo@type}}%
12830 \or
12831 \renewcommand*{\@glossarysecstar}{*}%
12832 \renewcommand*{\@glossaryseclabel}{%
12833 \protected@edef\@currentlabelname{\glossarytoctitle}%
12834 \label{\glsautoprefix\@glo@type}}%
12835 \fi
12836 }
12837 \ifpackageloaded{classicthesis}
12838 {\newcommand*{\@glossary@default@style}{index}}
12839 {\newcommand*{\@glossary@default@style}{list}}

```



```

12840 \define@key{glossaries.sty}{style}{%
12841   \def\@glossary@default@style{#1}%
12842 }
12843 \newcommand*\@gls@declareoption}[2]{%
12844   \DeclareOptionX{#1}{#2}%
12845   \DeclareOption{#1}{#2}%
12846 }
12847 \newcommand*\@glossaryentrynumbers}[1]{#1\@gls@save@numberlist{#1}}
12848 \@gls@declareoption{nonumberlist}{%
12849   \renewcommand*\@glossaryentrynumbers}[1]{\@gls@save@numberlist{#1}}%
12850 }
12851 \define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
12852 \glssavenumberlistfalse
12853 \newcommand*\@glo@seeautonumberlist{}
12854 \@gls@declareoption{seeautonumberlist}{%
12855   \renewcommand*\@glo@seeautonumberlist{}%
12856   \def\@glo@prefix{\glsnextpages}%
12857 }%
12858 }
12859 \define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
12860 \glsclocationstrue
12861 \newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
12862 \@gls@declareoption{nolong}{\renewcommand*\@gls@loadlong{}}
12863 \IfFileExists{supertabular.sty}{%
12864   \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}%
12865   \newcommand*\@gls@loadsuper{}%
12866 \@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}
12867 \newcommand*\@gls@loadlist{\RequirePackage{glossary-list}[=v4.46]}
12868 \@gls@declareoption{nolist}{%
12869   \renewcommand*\@gls@loadlist{}%
12870   \ifdefstring{\@glossary@default@style}{list}%
12871     {\let\@glossary@default@style\relax}%
12872   }%
12873 }%
12874 }
12875 \newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}}
12876 \@gls@declareoption{notree}{\renewcommand*\@gls@loadtree{}}
12877 \@gls@declareoption{nostyles}{%
12878   \renewcommand*\@gls@loadlong{}%
12879   \renewcommand*\@gls@loadsuper{}%
12880   \renewcommand*\@gls@loadlist{}%
12881   \renewcommand*\@gls@loadtree{}%
12882   \let\@glossary@default@style\relax
12883 }
12884 \newcommand*\@gls@postdescription{%
12885   \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
12886 }
12887 \define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
12888 \glsnopostdotfalse

```

```

12889 \define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
12890 \glsnogroupskipfalse
12891 \define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
12892 \@ifclassloaded{memoir}
12893 {%
12894   \glsucmarktrue
12895 }%
12896 {%
12897   \glsucmarkfalse
12898 }
12899 \newcommand*{\@gls@define@glossaryentrycounter}{%
12900   \ifglsentrycounter
12901     \ifundef\c@glossaryentry
12902     {%
12903       \ifx\@gls@counterwithin\@empty
12904         \newcounter{glossaryentry}%
12905       \else
12906         \newcounter{glossaryentry}[\@gls@counterwithin]%
12907       \fi
12908       \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
12909     }%
12910   {}%
12911 \fi
12912 }
12913 \define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
12914 \glsentrycounterfalse
12915 \define@key{glossaries.sty}{counterwithin}{%
12916   \renewcommand*{\@gls@counterwithin}{#1}%
12917   \glsentrycountertrue
12918   \@gls@define@glossaryentrycounter
12919 }
12920 \newcommand*{\@gls@counterwithin}{}
12921 \newcommand{\@gls@define@glossarysubentrycounter}{%
12922   \ifundef\c@glossarysubentry
12923   {%
12924     \ifglssubentrycounter
12925     \ifglsentrycounter
12926       \newcounter{glossarysubentry}[glossaryentry]%
12927     \else
12928       \newcounter{glossarysubentry}%
12929     \fi
12930     \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
12931   \fi
12932 }%
12933 {}%
12934 }
12935 \define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
12936 \glssubentrycounterfalse
12937 \newcommand*{\@gls@default@sorttype}{standard}

```

```

12938 \define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
12939   \renewcommand*{\@glo@default@sorttype}{#1}%
12940   \csname @gls@setupsort@#1\endcsname
12941 }
12942 \newcommand*{\glsprestandardsort}[3]{%
12943   \glsdosanitizesort
12944 }
12945 \newcommand*{\@glo@check@sortallowed}[1]{%
12946 \newcommand*{\@gls@setupsort@standard}{%
12947   \def\do@glo@storeentry{\@glo@storeentry}%
12948   \def\@gls@defsortcount##1{%
12949     \def\@gls@defsort##1##2{%
12950       \ifx\@glo@sort\@glsdefaultsort
12951         \let\@glo@sort\@glo@name
12952       \fi
12953       \let\glsdosanitizesort\@gls@sanitizesort
12954       \glsprestandardsort{\@glo@sort}{##1}{##2}%
12955       \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
12956     }%
12957     \def\@gls@setsort##1{%
12958       \let\@glo@check@sortallowed\@gobble
12959     }
12960 \@gls@setupsort@standard
12961 \newcommand*{\glssortnumberfmt}[1]{%
12962   \ifnum#1<100000 0\fi
12963   \ifnum#1<10000 0\fi
12964   \ifnum#1<1000 0\fi
12965   \ifnum#1<100 0\fi
12966   \ifnum#1<10 0\fi
12967   \number#1%
12968 }
12969 \newcommand*{\@gls@setupsort@def}{%
12970   \def\do@glo@storeentry{\@glo@storeentry}%
12971   \def\@gls@defsortcount##1{%
12972     \expandafter\global
12973     \expandafter\newcount\csname glossary@##1@sortcount\endcsname
12974   }%
12975   \def\@gls@defsort##1##2{%
12976     \ifcsundef{glossary@##1@sortcount}%
12977     {\@gls@defsortcount{##1}}%
12978     {}%
12979     \expandafter\global\expandafter
12980     \advance\csname glossary@##1@sortcount\endcsname by 1\relax
12981     \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
12982       \expandafter\glssortnumberfmt
12983       {\csname glossary@##1@sortcount\endcsname}}%
12984   }%
12985   \def\@gls@setsort##1{%
12986     \let\@glo@check@sortallowed\@gobble

```

```

12987 }
12988 \newcommand*{\@gls@setupsort@use}{%
12989   \let\do@glo@storeentry\@gobble
12990   \def\@gls@defsortcount##1{%
12991     \expandafter\global
12992     \expandafter\newcount\csname glossary@##1@sortcount\endcsname
12993   }%
12994   \def\@gls@defsort##1##2{%
12995     \expandafter\gdef\csname glo@##2@sort\endcsname{%}%
12996   }%
12997   \def\@gls@setsort##1{%
12998     \edef\@glo@parent{\csname glo@##1@parent\endcsname}%
12999     \ifx\@glo@parent\@empty
13000     \else
13001       \expandafter\@gls@setsort\expandafter{\@glo@parent}%
13002     \fi
13003     \edef\@glo@type{\csname glo@##1@type\endcsname}%
13004     \edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
13005     \ifx\@gls@tmp\@empty
13006       \expandafter\global\expandafter
13007       \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
13008       \expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
13009         \expandafter\glssortnumberfmt
13010         {\csname glossary@\@glo@type @sortcount\endcsname}}%
13011       \@glo@storeentry{##1}%
13012     \fi
13013   }%
13014   \let\@glo@check@sortallowed\@gobble
13015 }
13016 \newcommand*{\@gls@setupsort@none}{%
13017   \def\do@glo@storeentry##1{%}%
13018   \def\@gls@defsortcount##1{%}%
13019   \def\@gls@defsort##1##2{%
13020     \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
13021   }%
13022   \def\@gls@setsort##1{%
13023     \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
13024       {Option sort=none not allowed with \string##1}%
13025       {(Use sort=def instead)}}%
13026 }
13027 \newcommand*{\glsdefmain}{%
13028   \if@gls@docloaded
13029     \newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
13030   \else
13031     \newglossary{main}{gls}{glo}{\glossaryname}%
13032   \fi
13033   \newcommand*{\gls@tr@set@main@toctitle}{%
13034     \translatelet{\glossarytoctitle}{Glossary}%
13035   }%

```

```

13036 }
13037 \newcommand*{\glsdefaulttype}{main}
13038 \newcommand*{\acronymtype}{\glsdefaulttype}
13039 \@gls@declareoption{nomain}{%
13040   \let\glsdefaulttype\relax
13041   \renewcommand*{\glsdefmain}{}}%
13042 }
13043 \define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
13044   \ifglsacronym
13045     \renewcommand{\@gls@do@acronymsdef}{%
13046       \DeclareAcronymList{acronym}%
13047       \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
13048       \renewcommand*{\acronymtype}{acronym}%
13049       \newcommand*{\gls@tr@set@acronym@toctitle}{%
13050         \translatelet{\glossarytoctitle}{Acronyms}%
13051       }%
13052     }%
13053   \else
13054     \let\@gls@do@acronymsdef\relax
13055   \fi
13056 }
13057 \AtBeginDocument{%
13058   \ifglsacronym
13059     \ifbool{glscompatible-3.07}{%
13060       {}%
13061     }{%
13062       \providecommand*{\printacronyms}[1][1]{%
13063         \printglossary[type=\acronymtype,#1]}%
13064     }%
13065   \fi
13066 }
13067 \newcommand*{\@gls@do@acronymsdef}{}
13068 \@gls@declareoption{acronyms}{%
13069   \glsacronymtrue
13070   \def\@gls@do@acronymsdef{%
13071     \DeclareAcronymList{acronym}%
13072     \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
13073     \renewcommand*{\acronymtype}{acronym}%
13074     \newcommand*{\gls@tr@set@acronym@toctitle}{%
13075       \translatelet{\glossarytoctitle}{Acronyms}%
13076     }%
13077   }%
13078 }
13079 \newcommand*{\@glsacronymlists}{}
13080 \newcommand*{\@addtoacronymlists}[1]{%
13081   \ifx\@glsacronymlists\@empty
13082     \protected@xdef\@glsacronymlists{#1}%
13083   \else
13084     \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%

```

```

13085 \fi
13086 }
13087 \newcommand*{\DeclareAcronymList}[1]{%
13088   \glsIfListOfAcronyms{#1}{}{\@addtoacronymlists{#1}}%
13089 }
13090 \newcommand{\glsIfListOfAcronyms}[1]{%
13091   \edef\@do@gl@s@islistofacronyms{%
13092     \noexpand\@gl@s@islistofacronyms{#1}{\@gl@s@acronymlists}}%
13093   \@do@gl@s@islistofacronyms
13094 }
13095 \newcommand{\@gl@s@islistofacronyms}[4]{%
13096   \def\gl@s@islistofacronyms##1,#1,##2\end@gl@s@islistofacronyms{%
13097     \def\@gl@s@before{##1}\def\@gl@s@after{##2}}%
13098   \gl@s@islistofacronyms,#2,#1,\@nil\end@gl@s@islistofacronyms
13099   \ifx\@gl@s@after\@nnil
13100     #4%
13101   \else
13102     #3%
13103   \fi
13104 }
13105 \newif\if@gl@sisacronymlist
13106 \newcommand*{\gl@s@checkisacronymlist}[1]{%
13107   \glsIfListOfAcronyms{#1}%
13108   {\@gl@sisacronymlisttrue}{\@gl@sisacronymlistfalse}%
13109 }
13110 \newcommand*{\SetAcronymLists}[1]{%
13111   \renewcommand*{\@gl@s@acronymlists}{#1}%
13112 }
13113 \define@key{glossaries.sty}{acronymlists}{%
13114   \DeclareAcronymList{#1}%
13115 }
13116 \newcommand{\glscounter}{page}
13117 \define@key{glossaries.sty}{counter}{%
13118   \renewcommand*{\glscounter}{#1}%
13119 }
13120 \newcommand*{\@gl@s@nohyperlist}{}
13121 \newcommand*{\GlsDeclareNoHyperList}[1]{%
13122   \ifdefempty\@gl@s@nohyperlist
13123     {%
13124       \renewcommand*{\@gl@s@nohyperlist}{#1}%
13125     }%
13126     {%
13127       \appto\@gl@s@nohyperlist{,#1}%
13128     }%
13129 }
13130 \define@key{glossaries.sty}{nohypertypes}{%
13131   \GlsDeclareNoHyperList{#1}%
13132 }
13133 \newcommand*{\GlossariesWarning}[1]{%

```

```

13134 \PackageWarning{glossaries}{#1}%
13135 }
13136 \newcommand*\GlossariesWarningNoLine}[1]{%
13137 \PackageWarningNoLine{glossaries}{#1}%
13138 }
13139 \newcommand{\glosortentrieswarning}{%
13140 \typeout{Using TeX to sort glossary entries---this may
13141 take a while}%
13142 }
13143 \@gls@declareoption{nowarn}{%
13144 \if@gls@debug
13145 \GlossariesWarning{Warnings can't be suppressed in debug mode}%
13146 \else
13147 \renewcommand*\GlossariesWarning}[1]{}%
13148 \renewcommand*\GlossariesWarningNoLine}[1]{}%
13149 \renewcommand*\glosortentrieswarning}{}%
13150 \renewcommand*\@gls@missinglang@warn}[2]{}%
13151 \fi
13152 }
13153 \newcommand*\@gls@missinglang@warn}[2]{%
13154 \PackageWarningNoLine{glossaries}%
13155 {No language module detected for '#1'.\MessageBreak
13156 Language modules need to be installed separately.\MessageBreak
13157 Please check on CTAN for a bundle called\MessageBreak
13158 'glossaries-#2' or similar}%
13159 }
13160 \@gls@declareoption{nolangwarn}{%
13161 \renewcommand*\@gls@missinglang@warn}[2]{}%
13162 }
13163 \newcommand*\@gls@warnonglossdefined}{%
13164 \GlossariesWarning{Overriding \string\printglossary}%
13165 }
13166 \newcommand*\@gls@warnontheglossdefined}{%
13167 \GlossariesWarning{Overriding 'theglossary' environment}%
13168 }
13169 \@gls@declareoption{noredefwarn}{%
13170 \renewcommand*\@gls@warnonglossdefined}{}%
13171 \renewcommand*\@gls@warnontheglossdefined}{}%
13172 }
13173 \newcommand*\@gls@sanitizedesc}{%
13174 }
13175 \newcommand*\glssetexpandfield}[1]{%
13176 \csdef{gls@assign@#1@field}##1##2{%
13177 \@@gls@expand@field{##1}{#1}{##2}%
13178 }%
13179 }
13180 \newcommand*\glssetnoexpandfield}[1]{%
13181 \csdef{gls@assign@#1@field}##1##2{%
13182 \@@gls@noexpand@field{##1}{#1}{##2}%

```

```

13183 }%
13184 }
13185 \glssetexpandfield{type}
13186 \glssetnoexpandfield{desc}
13187 \glssetnoexpandfield{descplural}
13188 \newcommand*{\@gls@sanitizename}{%
13189 \glssetnoexpandfield{name}
13190 \newcommand*{\@gls@sanitizesymbol}{%
13191 \glssetnoexpandfield{symbol}
13192 \glssetnoexpandfield{symbolplural}
13193 \newcommand*{\@gls@sanitizesort}{%
13194 \ifglssanitizesort
13195 \@@gls@sanitizesort
13196 \else
13197 \@@gls@nosanitizesort
13198 \fi
13199 }
13200 \newcommand*{\@@gls@sanitizesort{%
13201 \@onelevel@sanitize\@glo@sort
13202 }
13203 \newcommand*{\@@gls@nosanitizesort}{%
13204 \newcommand*{\@gls@noidx@sanitizesort{%
13205 \ifdefvoid\@glo@sort
13206 }%
13207 {%
13208 \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
13209 }%
13210 }
13211 \def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
13212 \def\@glo@sort{#1#2}%
13213 \@onelevel@sanitize\@glo@sort
13214 }
13215 \newcommand*{\@@gls@noidx@nosanitizesort}{%
13216 \ifdefvoid\@glo@sort
13217 }%
13218 {%
13219 \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
13220 }%
13221 }
13222 \def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
13223 \bgroup
13224 \glsnoidxstripaccents
13225 \protected@xdef\@glo@sort{#1#2}%
13226 \egroup
13227 \let\@glo@sort\@glo@sort
13228 }
13229 \newcommand*\glsnoidxstripaccents{%
13230 \let\IeC\@firstofone
13231 \let\add@accent@\@secondoftwo

```



```

13232 \let\@text@composite@x\@secondoftwo
13233 \let\@tabacckludge\@secondoftwo
13234 \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
13235 \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
13236 \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
13237 \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
13238 \let\'\'@firstofone
13239 \let\'\'@firstofone
13240 \let\~\@firstofone
13241 \let\"\"@firstofone
13242 \let\u\@firstofone
13243 \let\t\@firstofone
13244 \let\d\@firstofone
13245 \let\r\@firstofone
13246 \let\=\@firstofone
13247 \let\.\@firstofone
13248 \let\~\@firstofone
13249 \let\v\@firstofone
13250 \let\H\@firstofone
13251 \let\c\@firstofone
13252 \let\b\@firstofone
13253 \let\a\@secondoftwo
13254 \def\AE{AE}%
13255 \def\ae{ae}%
13256 \def\OE{OE}%
13257 \def\oe{oe}%
13258 \def\AA{AA}%
13259 \def\aa{aa}%
13260 \def\L{L}%
13261 \def\l{l}%
13262 \def\O{O}%
13263 \def\o{o}%
13264 \def\SS{SS}%
13265 \def\ss{ss}%
13266 \def\th{th}%
13267 \def\TH{TH}%
13268 \def\dh{dh}%
13269 \def\DH{DH}%
13270 }
13271 \@ifl@t@r\fmtversion{2019/10/01}
13272 {%
13273 \appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
13274 }
13275 {}
13276 \define@boolkey[glS]{sanitize}{description}[true]{%
13277 \GlossariesWarning{sanitize={description} package option deprecated}%
13278 \ifglS@sanitize@description
13279 \glSsetnoexpandfield{desc}%
13280 \glSsetnoexpandfield{descplural}%

```

```

13281 \else
13282   \glsssetexpandfield{desc}%
13283   \glsssetexpandfield{descplural}%
13284 \fi
13285 }
13286 \define@boolkey[glS]{sanitize}{name}[true]{%
13287   \GlossariesWarning{sanitize={name} package option deprecated}%
13288   \ifglS@sanitize@name
13289     \glsssetnoexpandfield{name}%
13290   \else
13291     \glsssetexpandfield{name}%
13292   \fi
13293 }
13294 \define@boolkey[glS]{sanitize}{symbol}[true]{%
13295   \GlossariesWarning{sanitize={symbol} package option deprecated}%
13296   \ifglS@sanitize@symbol
13297     \glsssetnoexpandfield{symbol}%
13298     \glsssetnoexpandfield{symbolplural}%
13299   \else
13300     \glsssetexpandfield{symbol}%
13301     \glsssetexpandfield{symbolplural}%
13302   \fi
13303 }
13304 \define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
13305   \ifglsssanitizesort
13306     \glsssetnoexpandfield{sortvalue}%
13307     \renewcommand*{\@glS@noidx@setsanitizesort}{%
13308       \glsssanitizesorttrue
13309       \glsssetnoexpandfield{sortvalue}%
13310     }%
13311   \else
13312     \glsssetexpandfield{sortvalue}%
13313     \renewcommand*{\@glS@noidx@setsanitizesort}{%
13314       \glsssanitizesortfalse
13315       \glsssetexpandfield{sortvalue}%
13316     }%
13317   \fi
13318 }
13319 \glsssanitizesorttrue
13320 \glsssetnoexpandfield{sortvalue}%
13321 \newcommand*{\@glS@noidx@setsanitizesort}{%
13322   \glsssanitizesortfalse
13323   \glsssetexpandfield{sortvalue}%
13324 }
13325 \define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
13326   \setbool{glsssanitizesort}{#1}%
13327   \ifglsssanitizesort
13328     \glsssetnoexpandfield{sortvalue}%
13329   \else

```

```

13330 \glssetexpandfield{sortvalue}%
13331 \fi
13332 \GlossariesWarning{sanitize={sort} package option
13333 deprecated. Use sanitizesort instead}%
13334 }
13335 \define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
13336 \ifthenelse{\equal{#1}{none}}{%
13337 {%
13338 \GlossariesWarning{sanitize package option deprecated}%
13339 \glssetexpandfield{name}%
13340 \glssetexpandfield{symbol}%
13341 \glssetexpandfield{symbolplural}%
13342 \glssetexpandfield{desc}%
13343 \glssetexpandfield{descplural}%
13344 }%
13345 {%
13346 \setkeys{gls}{sanitize}{#1}%
13347 }%
13348 }
13349 \newif\ifglstranslate
13350 \newcommand*\@gls@usetranslator{%
13351 \@ifpackageloaded{polyglossia}%
13352 {%
13353 \let\glsifusetranslator\@secondoftwo
13354 }%
13355 {%
13356 \@ifpackageloaded{babel}%
13357 {%
13358 \IfFileExists{translator.sty}%
13359 {%
13360 \RequirePackage{translator}%
13361 \let\glsifusetranslator\@firstoftwo
13362 }%
13363 }%
13364 }%
13365 {}%
13366 }%
13367 }
13368 \newcommand{\glsifusedtranslatordict}[3]{%
13369 \glsifusetranslator
13370 {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
13371 {#3}%
13372 }
13373 \@gls@declareoption{nottranslate}{%
13374 \glstranslatefalse
13375 \let\@gls@usetranslator\relax
13376 \let\glsifusetranslator\@secondoftwo
13377 }
13378 \define@choicekey{glossaries.sty}{translate}%

```

```

13379 [\gls@translate@val\gls@translate@nr]%
13380 {true,false,babel}[true]%
13381 {%
13382   \ifcase\gls@translate@nr\relax
13383     \glstranslatetrue
13384     \renewcommand*\@gls@usetranslator{%
13385       \@ifpackageloaded{polyglossia}%
13386       {%
13387         \let\glsifusetranslator\@secondoftwo
13388       }%
13389       {%
13390         \@ifpackageloaded{babel}%
13391         {%
13392           \IfFileExists{translator.sty}%
13393           {%
13394             \RequirePackage{translator}%
13395             \let\glsifusetranslator\@firstoftwo
13396           }%
13397           {}%
13398         }%
13399         {}%
13400       }%
13401     }%
13402   \or
13403     \glstranslatefalse
13404     \let\@gls@usetranslator\relax
13405     \let\glsifusetranslator\@secondoftwo
13406   \or
13407     \glstranslatetrue
13408     \let\@gls@usetranslator\relax
13409     \let\glsifusetranslator\@secondoftwo
13410   \fi
13411 }
13412 \glstranslatefalse
13413 \let\glsifusetranslator\@secondoftwo
13414 \@ifpackageloaded{translator}%
13415 {%
13416   \glstranslatetrue
13417   \let\glsifusetranslator\@firstoftwo
13418 }%
13419 {%
13420   \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
13421   {
13422     \@ifpackageloaded{\gls@thissty}%
13423     {%
13424       \glstranslatetrue
13425       \@endfortrue
13426     }%
13427     {}%

```

```

13428 }
13429 }
13430 \define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
13431 \glsindexonlyfirstfalse
13432 \define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
13433 \glshyperfirsttrue
13434 \newcommand*{\@gls@setacrstyle}{}
13435 \define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
13436 \ifbool{glsacrdescription}%
13437 {}%
13438 {}%
13439 \renewcommand*{\@gls@sanitizedesc}{}%
13440 }%
13441 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13442 }
13443 \define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
13444 \renewcommand*{\@gls@sanitizesymbol}{}%
13445 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13446 }
13447 \define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
13448 \renewcommand*{\@gls@sanitizesymbol}{}%
13449 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13450 }
13451 \define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
13452 \renewcommand*{\@gls@sanitizesymbol}{}%
13453 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13454 }
13455 \define@boolkey{glossaries.sty}[glsacr]{dua}[true]{}
13456 \renewcommand*{\@gls@sanitizesymbol}{}%
13457 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13458 }
13459 \define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
13460 \newcommand*{\glsorder}{word}
13461 \newcommand*{\@glsorder}[1]{}
13462 \define@choicekey{glossaries.sty}{order}{word,letter}{}
13463 \def\glsorder{#1}
13464 \newif\ifglxindy
13465 \glxindyfalse
13466 \@gls@declareoption{makeindex}{\glxindyfalse}
13467 \define@boolkey[gls]{xindy}{glsnumbers}[true]{}
13468 \gls@xindy@glsnumberstrue
13469 \def\@xdy@main@language{\language}%
13470 \define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
13471 \ifcsundef{inputencodingname}{}%
13472 \def\gls@codepage{}{}%
13473 \def\gls@codepage{\inputencodingname}
13474 }
13475 \define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
13476 \define@key{glossaries.sty}{xindy}[]{}

```

```

13477 \glxindytrue
13478 \setkeys[glx]{indy}{#1}%
13479 }
13480 \@glx@declareoption{xindygloss}{%
13481 \glxindytrue
13482 }
13483 \@glx@declareoption{xindynoglsnumbers}{%
13484 \glxindytrue
13485 \glx@xindy@glxnumbersfalse
13486 }
13487 \providecommand{\@domakeglossaries}[1]{#1}
13488 \@glx@declareoption{disablemakegloss}{%
13489 \ifdefequal\makeglossaries\@no@makeglossaries
13490 {%
13491 \GlossariesWarning{Option ‘disablemakegloss’ has no effect
13492 (\string\makenoidxglossaries\space has already been used)}%
13493 }%
13494 {%
13495 \ifdefequal\@makeglossary\@gobble
13496 {%
13497 \GlossariesWarning{Option ‘disablemakegloss’ has no effect
13498 (\string\makeglossaries\space has already been used)}%
13499 }%
13500 {%
13501 \renewcommand{\@domakeglossaries}[1]{%
13502 \PackageInfo{glossaries}{\string\makeglossaries\space and
13503 \string\makenoidxglossaries\space have been disabled}%
13504 }%
13505 }%
13506 }%
13507 }
13508 \@glx@declareoption{restoremakegloss}{%
13509 \ifdefequal\makeglossaries\@no@makeglossaries
13510 {%
13511 \GlossariesWarning{Option ‘restoremakegloss’ has no effect
13512 (\string\makenoidxglossaries\space has already been used)}%
13513 }%
13514 {%
13515 \ifdefequal\@makeglossary\@gobble
13516 {%
13517 \GlossariesWarning{Option ‘restoremakegloss’ has no effect
13518 (\string\makeglossaries\space has already been used)}%
13519 }%
13520 {%
13521 \PackageInfo{glossaries}{\string\makeglossaries\space and
13522 \string\makenoidxglossaries\space have been restored}%
13523 \let\@domakeglossaries\@firstofone
13524 }%
13525 }%

```

```

13526 }
13527 \newcommand*{\@do@write@glslabels}{%
13528   \AtEndDocument{\@do@write@glslabels}%
13529   \let\@do@write@glslabels\relax
13530 }
13531 \newcommand*{\@do@do@write@glslabels}{%
13532   \newwrite\@glsl@labelsfile
13533   \immediate\openout\@glsl@labelsfile=\jobname.glslabels
13534   \forallglstentries[\@glo@types,\@ignored@glossaries]{\@glstentry}%
13535   {\ifdefempty{\@glstentry}{\immediate\write\@glsl@labelsfile{\@glstentry}}}%
13536   \immediate\closeout\@glsl@labelsfile
13537 }
13538 \@glsl@declareoption{writeglslabels}{\@do@write@glslabels}
13539 \newif\ifglslautomake
13540 \newcommand{\glsl@automake@nr}{1}
13541 \define@choicekey{glossaries.sty}{automake}%
13542   [\glsl@automake@val\glsl@automake@nr]{true,false,immediate}[true]{%
13543   \ifnum\glsl@automake@nr=1\relax
13544     \glslautomakefalse
13545   \else
13546     \glslautomaketrue
13547   \fi
13548   \ifglslautomake
13549     \renewcommand*{\@glsl@doautomake}{%
13550       \PackageError{glossaries}{You must use
13551       \string\makeglossaries\space with automake=true}
13552       {%
13553         Either remove the automake=true setting or
13554         add \string\makeglossaries\space to your document preamble.%
13555       }%
13556     }%
13557   \else
13558     \renewcommand*{\@glsl@doautomake}{}%
13559   \fi
13560 }
13561 \glslautomakefalse
13562 \newcommand*{\@glsl@doautomake}{}
13563 \AtEndDocument{\@glsl@doautomake}
13564 \define@boolkey{glossaries.sty}[glsl]{savewrites}[true]{%
13565   \ifglssavewrites
13566     \renewcommand*{\glslwritefiles}{\@glslwritefiles}%
13567   \else
13568     \let\glslwritefiles\@empty
13569   \fi
13570 }
13571 \glssavewritesfalse
13572 \let\glslwritefiles\@empty
13573 \define@boolkey{glossaries.sty}[glsl]{compatible-3.07}[true]{%
13574 \boolfalse{glslcompatible-3.07}

```

```

13575 \define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
13576   \ifbool{glscompatible-2.07}%
13577   {%
13578     \booltrue{glscompatible-3.07}%
13579   }%
13580   {}%
13581 }
13582 \boolfalse{glscompatible-2.07}
13583 \let\gls@original@makeglossary\makeglossary
13584 \let\gls@original@glossary\glossary
13585 \def\makeglossary{%
13586   \GlossariesWarning{Use of \string\makeglossary\space with
13587   glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
13588   instead. If you \MessageBreak need the original definition of
13589   \string\makeglossary\space use \MessageBreak the package options
13590   kernelglossredefs=false (to \MessageBreak restore the former definition of
13591   \string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
13592   conflict)}}%
13593   \makeglossaries
13594 }
13595 \newcommand*{\@gls@override@glossary}[1][main]{%
13596   \GlossariesWarning{Use of \string\glossary\space with
13597   glossaries.sty is deprecated. \MessageBreak Indexing should be performed
13598   with the user level \MessageBreak commands, such as \string\gls\space or
13599   \string\glsadd. If you need the \MessageBreak original definition of
13600   \string\glossary\space use the package \MessageBreak options
13601   kernelglossredefs=false (to restore the \MessageBreak former definition of
13602   \string\glossary) and nomain (if the \MessageBreak file extensions cause a
13603   conflict)}}%
13604   \gls@glossary{#1}%
13605 }
13606 \if@gls@docloaded
13607 \else
13608   \def\glossary{\@gls@override@glossary}
13609 \fi
13610 \define@choicekey{glossaries.sty}{kernelglossredefs}%
13611   [\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
13612 {%
13613   \ifcase\gls@debug@nr\relax
13614     \def\glossary{\@gls@override@glossary}%
13615     \def\makeglossary{%
13616       \GlossariesWarning{Use of \string\makeglossary\space with
13617       glossaries.sty is deprecated. Use \string\makeglossaries\space
13618       instead. If you need the original definition of
13619       \string\makeglossary\space use the package options
13620       kernelglossredefs=false (to prevent redefinition of
13621       \string\makeglossary) and nomain (if the file extensions cause a
13622       conflict)}}%
13623       \makeglossaries

```



```

13624 }%
13625 \or
13626 \let\glossary\gls@original@glossary
13627 \let\makeglossary\gls@original@makeglossary
13628 \or
13629 \def\makeglossary{\makeglossaries}%
13630 \renewcommand*{\@gls@override@glossary}[1][main]{%
13631 \gls@glossary{##1}%
13632 }%
13633 \fi
13634 }
13635 \@gls@declareoption{symbols}{%
13636 \let\@gls@do@symbolsdef\@gls@symbolsdef
13637 }
13638 \newcommand*{\@gls@do@symbolsdef}{%
13639 \newcommand*{\@gls@symbolsdef}{%
13640 \newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
13641 \newcommand*{\printsymbols}[1][ ]{\printglossary[type=symbols,##1]}%
13642 \newcommand*{\gls@tr@set@symbols@toctitle}{%
13643 \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
13644 }%
13645 }%
13646 \@gls@declareoption{numbers}{%
13647 \let\@gls@do@numbersdef\@gls@numbersdef
13648 }
13649 \newcommand*{\@gls@do@numbersdef}{%
13650 \newcommand*{\@gls@numbersdef}{%
13651 \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
13652 \newcommand*{\printnumbers}[1][ ]{\printglossary[type=numbers,##1]}%
13653 \newcommand*{\gls@tr@set@numbers@toctitle}{%
13654 \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
13655 }%
13656 }%
13657 \@gls@declareoption{index}{%
13658 \ifx\@gls@do@indexdef\@empty
13659 \let\@gls@do@indexdef\@gls@indexdef
13660 \fi
13661 }
13662 \@gls@declareoption{noglossaryindex}{%
13663 \let\@gls@do@indexdef\relax
13664 }
13665 \newcommand*{\@gls@do@indexdef}{%
13666 \newcommand*{\@gls@indexdef}{%
13667 \newglossary[ilg]{index}{ind}{idx}{\indexname}%
13668 \newcommand*{\printindex}[1][ ]{\printglossary[type=index,##1]}%
13669 \newcommand*{\newterm}[2][ ]{%
13670 \newglossaryentry{##2}%
13671 {type={index},name={##2},description={\nopostdesc},##1}}
13672 \let\@gls@do@indexdef\relax

```

```

13673 }%
13674 \@for\CurrentOption := \@declaredoptions\do{%
13675   \ifx\CurrentOption\@empty
13676   \else
13677     \@expandtwoargs
13678     \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
13679     \ifin@
13680     \@use@ption
13681     \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
13682     \fi
13683   \fi
13684 }
13685 \ProcessOptionsX
13686 \RequirePackage{glossaries-compatible-307}
13687 \disable@keys{glossaries.sty}{compatible-2.07,%
13688 xindy,xindygloss,xindynoglsnumbers,makeindex,%
13689 acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
13690 nomain,noglossaryindex}
13691 \newcommand*{\setupglossaries}[1]{%
13692   \renewcommand*{@gls@setacrstyle}{}%
13693   \ifglsacrshortcuts
13694     \def@gls@setupshortcuts{glsacrshortcutstrue}%
13695   \else
13696     \def@gls@setupshortcuts{%
13697       \ifglsacrshortcuts
13698         \DefineAcronymSynonyms
13699       \fi
13700     }%
13701   \fi
13702   \glsacrshortcutsfalse
13703   \let@gls@do@numbersdef\relax
13704   \let@gls@do@symbolssdef\relax
13705   \let@gls@do@indexdef\relax
13706   \let@gls@do@acronymsdef\relax
13707   \ifgl Sentrycounter
13708     \let@gls@doentrycounterdef\relax
13709   \else
13710     \let@gls@doentrycounterdef@gls@define@glossaryentrycounter
13711   \fi
13712   \ifglssubentrycounter
13713     \let@gls@dosubentrycounterdef\relax
13714   \else
13715     \let@gls@dosubentrycounterdef@gls@define@glossarysubentrycounter
13716   \fi
13717   \setkeys{glossaries.sty}{#1}%
13718   \@gls@setacrstyle
13719   \@gls@setupshortcuts
13720   \@gls@do@acronymsdef
13721   \@gls@do@numbersdef

```

```

13722 \@gls@do@symbolssdef
13723 \@gls@do@indexdef
13724 \@gls@doentrycounterdef
13725 \@gls@dosubentrycounterdef
13726 }
13727 \ifthenelse{\equal{\glscounter}{section}}%
13728 {%
13729   \ifcsundef{chapter}{}%
13730   {%
13731     \let\@gls@old@chapter\@chapter
13732     \def\@chapter[#1]#2{\@gls@old@chapter[#1]{#2}%
13733       \ifcsundef{hyperdef}{}{\hyperdef{section}{\thesection}{}}}%
13734   }%
13735 }%
13736 {}
13737 \newcommand*{\@gls@onlypremakeg}{}
13738 \newcommand*{\@onlypremakeg}[1]{%
13739   \ifx\@gls@onlypremakeg\@empty
13740     \def\@gls@onlypremakeg{#1}%
13741   \else
13742     \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
13743     \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
13744   \fi
13745 }
13746 \newcommand*{\@disable@onlypremakeg}{%
13747 \@for\@thiscs:=\@gls@onlypremakeg\do{%
13748   \expandafter\@disable@premakecs\@thiscs%
13749 }}
13750 \newcommand*{\@disable@premakecs}[1]{%
13751   \def#1{\PackageError{glossaries}{\string#1\space may only be
13752     used before \string\makeglossaries}{You can't use
13753     \string#1\space after \string\makeglossaries}}%
13754 }
13755 \providecommand*{\glossaryname}{Glossary}
13756 \providecommand*{\acronymname}{Acronyms}
13757 \newcommand*{\glssettocitle}[1]{%
13758   \def\glossarytocitle{\csname @glotype@#1@title\endcsname}}
13759 \providecommand*{\entryname}{Notation}
13760 \providecommand*{\descriptionname}{Description}
13761 \providecommand*{\symbolname}{Symbol}
13762 \providecommand*{\pagelistname}{Page List}
13763 \providecommand*{\glssymbolsgroupname}{Symbols}
13764 \providecommand*{\glsnumbersgroupname}{Numbers}
13765 \newcommand*{\glspluralsuffix}{s}
13766 \newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
13767 \newcommand*{\glsupacrpluralsuffix}{\glsacrpluralsuffix}
13768 \providecommand*{\seename}{see}
13769 \providecommand*{\andname}{\&}
13770 \newcommand*{\RequireGlossariesLang}[1]{%

```

```

13771 \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
13772 }
13773 \newcommand*{\ProvidesGlossariesLang}[1]{%
13774   \ProvidesFile{glossaries-#1.ldf}%
13775 }
13776 \newcommand*{\addglossarytocaptions}[1]{%
13777   \ifglstranslate
13778     \RequirePackage{tracklang}
13779     \@gls@usetranslator
13780     \ifpackageloaded{translator}
13781     {%
13782       \ifboolexpr
13783       {
13784         test {\ifdefstring{\trans@languages}{English}}
13785         and not
13786         test {\ifdefstring{bbl@loaded}{english}}
13787       }
13788       {%
13789         \let\glsifusetranslator\@secondoftwo
13790       }%
13791       {%
13792         \usedictionary{glossaries-dictionary}%
13793         \renewcommand*{\addglossarytocaptions}[1]{%
13794           \ifcsundef{captions#1}{}%
13795           {%
13796             \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
13797             \expandafter\toks@\expandafter{\@gls@tmp
13798               \renewcommand*{\glossaryname}{\translate{Glossary}}}%
13799             }%
13800             \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
13801           }%
13802         }%
13803       }%
13804     }%
13805   }{}%
13806   \AnyTrackedLanguages
13807   {%
13808     \ForEachTrackedDialect{\this@dialect}{%
13809       \IfTrackedLanguageFileExists{\this@dialect}%
13810       {glossaries-}% prefix
13811       {.ldf}%
13812     }%
13813     \RequireGlossariesLang{\CurrentTrackedTag}%
13814   }%
13815   {%
13816     \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
13817   }%
13818 }%
13819 }%

```

```

13820 {}%
13821 \glsifusetranslator
13822 {%
13823   \renewcommand*{\glssettoctitle}[1]{%
13824     \ifcsdef{gls@tr@set@#1@toctitle}%
13825     {%
13826       \csuse{gls@tr@set@#1@toctitle}%
13827     }%
13828     {%
13829       \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
13830     }%
13831   }%
13832   \renewcommand*{\glossaryname}{\translate{Glossary}}%
13833   \renewcommand*{\acronymname}{\translate{Acronyms}}%
13834   \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
13835   \renewcommand*{\descriptionname}{%
13836     \translate{Description (glossaries)}}%
13837   \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
13838   \renewcommand*{\pagelistname}{%
13839     \translate{Page List (glossaries)}}%
13840   \renewcommand*{\glssymbolsgroupname}{%
13841     \translate{Symbols (glossaries)}}%
13842   \renewcommand*{\glsnumbersgroupname}{%
13843     \translate{Numbers (glossaries)}}%
13844   }{}%
13845 \fi
13846 \DeclareRobustCommand*{\nopostdesc}{%
13847 \newcommand*{\@nopostdesc}{%
13848   \let\org@glspostdescription\glspostdescription
13849   \def\glspostdescription{%
13850     \let\glspostdescription\org@glspostdescription}%
13851 }
13852 \newcommand*{\@no@post@desc}{\nopostdesc}
13853 \newcommand{\glspar}{\par}
13854 \newcommand{\setStyleFile}[1]{%
13855   \renewcommand*{\gls@istfilebase}{#1}%
13856   \ifglsxindy
13857     \def\istfilename{\gls@istfilebase.xdy}
13858   \else
13859     \def\istfilename{\gls@istfilebase.ist}
13860   \fi
13861 }
13862 \@onlypremakeg\setStyleFile
13863 \ifglsxindy
13864   \def\istfilename{\gls@istfilebase.xdy}
13865 \else
13866   \def\istfilename{\gls@istfilebase.ist}
13867 \fi
13868 \newcommand*{\gls@istfilebase}{\jobname}

```

```

13869 \newcommand*{\@istfilename}[1]{}
13870 \newcommand*{\glscompositor}{.}
13871 \newcommand*{\glsSetCompositor}[1]{%
13872   \renewcommand*{\glscompositor}{#1}}
13873 \@onlypremakeg\glsSetCompositor
13874 \newcommand*{\@glsAlphacompositor}{\glscompositor}
13875 \ifglsexindy
13876   \newcommand*{\glsSetAlphaCompositor}[1]{%
13877     \renewcommand*{\@glsAlphacompositor}{#1}}
13878 \else
13879   \newcommand*{\glsSetAlphaCompositor}[1]{%
13880     \glsnoxywarning\glsSetAlphaCompositor}
13881 \fi
13882 \@onlypremakeg\glsSetAlphaCompositor
13883 \newcommand*{\gls@suffixF}{}
13884 \newcommand*{\glsSetSuffixF}[1]{%
13885   \renewcommand*{\gls@suffixF}{#1}}
13886 \@onlypremakeg\glsSetSuffixF
13887 \newcommand*{\gls@suffixFF}{}
13888 \newcommand*{\glsSetSuffixFF}[1]{%
13889   \renewcommand*{\gls@suffixFF}{#1}%
13890 }
13891 \ifcsundef{hyperlink}%
13892 {%
13893   \newcommand*{\glsnumberformat}[1]{#1}%
13894 }%
13895 {%
13896   \newcommand*{\glsnumberformat}[1]{\glshypernumber{#1}}%
13897 }
13898 \newcommand{\delimN}{, }
13899 \newcommand{\delimR}{--}
13900 \newcommand*{\glossarypreamble}{%
13901   \csuse{@glossarypreamble@\currentglossary}%
13902 }
13903 \newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
13904   \ifglossaryexists*{#1}%
13905     {\csgdef{@glossarypreamble@#1}{#2}}%
13906     {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
13907 }
13908 \newcommand*{\glossarypostamble}{}
13909 \newcommand*{\glossarysection}[2][\@gls@title]{%
13910   \def\@gls@title{#2}%
13911   \ifcsundef{phantomsection}%
13912     {%
13913       \@glossarysection{#1}{#2}%
13914     }%
13915     {%
13916       \@pglossarysection{#1}{#2}%
13917     }%

```

```

13918 \glsglossarymark{\glossarytoctitle}%
13919 }
13920 \ifcsundef{glossarymark}%
13921 {%
13922   \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
13923 }%
13924 {%
13925   \@ifclassloaded{memoir}
13926   {%
13927     \newcommand{\glsglossarymark}[1]{%
13928       \ifglsucmark
13929         \markboth{\memUHead{#1}}{\memUHead{#1}}%
13930       \else
13931         \markboth{#1}{#1}%
13932       \fi
13933     }
13934   }%
13935   {%
13936     \newcommand{\glsglossarymark}[1]{%
13937       \ifglsucmark
13938         \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
13939       \else
13940         \@mkboth{#1}{#1}%
13941       \fi
13942     }
13943   }
13944 }
13945 \providecommand{\glossarymark}[1]{%
13946   \ifglsucmark
13947     \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
13948   \else
13949     \@mkboth{#1}{#1}%
13950   \fi
13951 }
13952 \newcommand*{\setglossarysection}[1]{%
13953 \setkeys{glossaries.sty}{section=#1}}
13954 \newcommand*{\@glossarysection}[2]{%
13955   \ifdefempty\@glossarysecstar
13956   {%
13957     \csname\@glossarysec\endcsname[#1]{#2}%
13958   }%
13959   {%
13960     \csname\@glossarysec\endcsname*{#2}%
13961     \@gls@toc{#1}{\@glossarysec}%
13962   }%
13963   \@glossaryseclabel
13964 }
13965 \newcommand*{\@p@glossarysection}[2]{%
13966   \glsclearpage

```

```

13967 \phantomsection
13968 \ifdefempty\@@glossarysecstar
13969 {%
13970   \csname\@@glossarysec\endcsname{#2}%
13971 }%
13972 {%
13973   \@gls@toc{#1}{\@@glossarysec}%
13974   \csname\@@glossarysec\endcsname*{#2}%
13975 }%
13976 \@@glossaryseclabel
13977 }
13978 \newcommand*\@gls@docclearpage{%
13979   \ifthenelse{\equal{\@@glossarysec}{chapter}}%
13980   {%
13981     \ifcsundef{cleardoublepage}%
13982     {%
13983       \clearpage
13984     }%
13985     {%
13986       \ifcsdef{if@openright}%
13987       {%
13988         \if@openright
13989           \cleardoublepage
13990         \else
13991           \clearpage
13992         \fi
13993       }%
13994       {%
13995         \cleardoublepage
13996       }%
13997     }%
13998   }%
13999 }%
14000 }
14001 \newcommand*\@gls@clearpage{\@gls@docclearpage}
14002 \newcommand*\@gls@toc[2]{%
14003   \ifglstoc
14004     \ifglslnumberline
14005       \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
14006     \else
14007       \addcontentsline{toc}{#2}{#1}%
14008     \fi
14009   \fi
14010 }
14011 \newcommand*\@glsnoxywarning[1]{%
14012   \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
14013 }
14014 \newcommand*\@glsnomakeindexwarning[1]{%
14015   \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%

```



```

14016 }
14017 \ifglxsindy
14018   \edef\xdyattributes{\string"default\string"}%
14019 \fi
14020 \ifglxsindy
14021   \edef\xdyattributelist{}%
14022 \fi
14023 \ifglxsindy
14024   \def\xdylocref{}
14025 \fi
14026 \newcommand*{\@gls@ifinlist}[4]{%
14027   \def\@do@ifinlist##1,##2\end@do@ifinlist{%
14028     \def\@gls@listsuffix{##2}%
14029     \ifx\@gls@listsuffix\@empty
14030       #4%
14031     \else
14032       #3%
14033     \fi
14034   }%
14035   \@do@ifinlist,##1,##2,\end@do@ifinlist
14036 }
14037 \ifglxsindy
14038   \newcommand*{\@xdycounters}{\glscounter}
14039   \newcommand*\GlsAddXdyCounters[1]{%
14040     \@for\@gls@ctr:=#1\do{%
14041       \edef\@do@addcounter{%
14042         \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
14043         {%
14044           \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
14045             \noexpand\@gls@ctr}%
14046         }%
14047       }%
14048       \@do@addcounter
14049     }
14050   }
14051   \@onlypremakeg\GlsAddXdyCounters
14052 \else
14053   \newcommand*\GlsAddXdyCounters[1]{%
14054     \glsnoxywarning\GlsAddXdyAttribute
14055   }
14056 \fi
14057 \newcommand*\@disabled@glsaddxdycounters{%
14058   \PackageError{glossaries}{\string\GlsAddXdyCounters\space
14059     can't be used after \string\GlsAddXdyAttribute}{Move all
14060     occurrences of \string\GlsAddXdyCounters\space before the first
14061     instance of \string\GlsAddXdyAttribute}%
14062 }
14063 \ifglxsindy
14064   \newcommand*\@glsaddxdyattribute[2]{%

```

```

14065 \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
14066 \string"#2#1\string"}%
14067 \expandafter\toks@\expandafter{\@xdylocref}%
14068 \edef\@xdylocref{\the\toks@ ^^J%
14069 (markup-locref
14070 :open \string"\glstildechar n%
14071 \expandafter\string\csname glsX#2X#1\endcsname
14072 \string" ^^J
14073 :close \string"\string" ^^J
14074 :attr \string"#2#1\string")}%
14075 \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
14076 \setentrycounter[##1]{##2}\csname #1\endcsname{##2}%
14077 }%
14078 }
14079 \newcommand*\GlsAddXdyAttribute[1]{%
14080 \ifx\@xdyattributelist\@empty
14081 \edef\@xdyattributelist{#1}%
14082 \else
14083 \edef\@xdyattributelist{\@xdyattributelist,#1}%
14084 \fi
14085 \@for\@this@counter:=\@xdycounters\do{%
14086 \protected@edef\gls@do@addxdyattribute{%
14087 \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
14088 }
14089 \gls@do@addxdyattribute
14090 }%
14091 \let\GlsAddXdyCounters\@disabled@glsaddxdycounters
14092 }
14093 \@onlypremakeg\GlsAddXdyAttribute
14094 \else
14095 \newcommand*\GlsAddXdyAttribute[1]{%
14096 \glsnoxindywarning\GlsAddXdyAttribute}
14097 \fi
14098 \ifglxsindy
14099 \newcommand*{\@gls@addpredefinedattributes}{%
14100 \GlsAddXdyAttribute{glsnumberformat}
14101 \GlsAddXdyAttribute{textrm}
14102 \GlsAddXdyAttribute{textsf}
14103 \GlsAddXdyAttribute{texttt}
14104 \GlsAddXdyAttribute{textbf}
14105 \GlsAddXdyAttribute{textmd}
14106 \GlsAddXdyAttribute{textit}
14107 \GlsAddXdyAttribute{textup}
14108 \GlsAddXdyAttribute{textsl}
14109 \GlsAddXdyAttribute{textsc}
14110 \GlsAddXdyAttribute{emph}
14111 \GlsAddXdyAttribute{glshypernumber}
14112 \GlsAddXdyAttribute{hyperrm}
14113 \GlsAddXdyAttribute{hypersf}

```

```

14114 \GlsAddXdyAttribute{hypertt}
14115 \GlsAddXdyAttribute{hyperbf}
14116 \GlsAddXdyAttribute{hypermd}
14117 \GlsAddXdyAttribute{hyperit}
14118 \GlsAddXdyAttribute{hyperup}
14119 \GlsAddXdyAttribute{hypersl}
14120 \GlsAddXdyAttribute{hypersc}
14121 \GlsAddXdyAttribute{hyperemph}
14122 \GlsAddXdyAttribute{glsignore}
14123 }
14124 \else
14125 \let\@gls@addpredefinedattributes\relax
14126 \fi
14127 \def\@xdyuseralphabets{}
14128 \ifglxindy
14129 \newcommand*{\GlsAddXdyAlphabet}[2]{%
14130 \edef\@xdyuseralphabets{%
14131 \@xdyuseralphabets ^^J
14132 (define-alphabet "#1" (#2))}}
14133 \else
14134 \newcommand*{\GlsAddXdyAlphabet}[2]{%
14135 \glsnoxywarning\GlsAddXdyAlphabet}
14136 \fi
14137 \ifglxindy
14138 \newcommand*{\@gls@xdy@locationlist}{%
14139 roman-page-numbers,%
14140 Roman-page-numbers,%
14141 arabic-page-numbers,%
14142 alpha-page-numbers,%
14143 Alpha-page-numbers,%
14144 Appendix-page-numbers,%
14145 arabic-section-numbers%
14146 }
14147 \protected@edef\@gls@roman{\@roman{0}\string"
14148 \string"roman-numbers-lowercase\string" :sep \string"}}%
14149 \@onelevel@sanitize\@gls@roman
14150 \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
14151 :sep \string"}%
14152 \@onelevel@sanitize\@tmp
14153 \ifx\@tmp\@gls@roman
14154 \expandafter
14155 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
14156 \string"roman-numbers-lowercase\string"%
14157 }%
14158 \else
14159 \expandafter
14160 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
14161 :sep \string"\@gls@roman\string"%
14162 }%

```

```

14163 \fi
14164 \expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
14165   \string"roman-numbers-uppercase\string"%
14166 }%
14167 \expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
14168   \string"arabic-numbers\string"%
14169 }%
14170 \expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
14171   \string"alpha\string"%
14172 }%
14173 \expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
14174   \string"ALPHA\string"%
14175 }%
14176 \expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
14177   \string"ALPHA\string"
14178   :sep \string"@glsAlphacompositor\string"
14179   \string"arabic-numbers\string"%
14180 }
14181 \expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
14182   \string"arabic-numbers\string"
14183   :sep \string"\glscompositor\string"
14184   \string"arabic-numbers\string"%
14185 }%
14186 \def\@xdyuserlocationdefs{}
14187 \def\@xdyuserlocationnames{}
14188 \fi
14189 \ifglsxindy
14190 \newcommand\@xdycrossrefhook{}
14191 \fi
14192 \ifglsxindy
14193 \newcommand*{\GlsAddXdyLocation}[3][[]]{%
14194   \def\@gls@tmp{#1}%
14195   \ifx\@gls@tmp\@empty
14196     \edef\@xdyuserlocationdefs{%
14197       \@xdyuserlocationdefs ^^J%
14198       (define-location-class \string"#2\string"^^J\space\space
14199       \space(:sep \string"{}\glsopenbrace\string" #3
14200       :sep \string"\glsclosebrace\string"))
14201   }%
14202   \else
14203     \edef\@xdyuserlocationdefs{%
14204       \@xdyuserlocationdefs ^^J%
14205       (define-location-class \string"#2\string"^^J\space\space
14206       \space(:sep "\glsopenbrace"
14207       #1
14208       :sep "\glsclosebrace\glsopenbrace" #3
14209       :sep "\glsclosebrace"))
14210   }%
14211 \fi

```

```

14212     \edef\@xdyuserlocationnames{%
14213         \@xdyuserlocationnames^^J\space\space\space
14214         \string"#2\string"}%
14215     }
14216     \@onlypremakeg\GlsAddXdyLocation
14217 \else
14218     \newcommand*\GlsAddXdyLocation[2]{%
14219         \glsnoinxindywarning\GlsAddXdyLocation}
14220 \fi
14221 \ifglxindy
14222     \def\@xdylocationclassorder{^^J\space\space\space
14223         \string"roman-page-numbers\string"^^J\space\space\space
14224         \string"arabic-page-numbers\string"^^J\space\space\space
14225         \string"arabic-section-numbers\string"^^J\space\space\space
14226         \string"alpha-page-numbers\string"^^J\space\space\space
14227         \string"Roman-page-numbers\string"^^J\space\space\space
14228         \string"Alpha-page-numbers\string"^^J\space\space\space
14229         \string"Appendix-page-numbers\string"
14230         \@xdyuserlocationnames^^J\space\space\space
14231         \string"see\string"
14232     }
14233 \fi
14234 \ifglxindy
14235     \newcommand*\GlsSetXdyLocationClassOrder[1]{%
14236         \def\@xdylocationclassorder{#1}}
14237 \else
14238     \newcommand*\GlsSetXdyLocationClassOrder[1]{%
14239         \glsnoinxindywarning\GlsSetXdyLocationClassOrder}
14240 \fi
14241 \ifglxindy
14242     \def\@xdysortrules{}
14243 \fi
14244 \ifglxindy
14245     \newcommand*\GlsAddSortRule[2]{%
14246         \expandafter\toks@\expandafter{\@xdysortrules}%
14247         \protected@edef\@xdysortrules{\the\toks@ ^^J
14248             (sort-rule \string"#1\string" \string"#2\string")}%
14249     }
14250 \else
14251     \newcommand*\GlsAddSortRule[2]{%
14252         \glsnoinxindywarning\GlsAddSortRule}
14253 \fi
14254 \ifglxindy
14255     \def\@xdyrequiredstyles{tex}
14256 \fi
14257 \ifglxindy
14258     \newcommand*\GlsAddXdyStyle[1]{%
14259         \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
14260 \else

```

```

14261 \newcommand*\GlsAddXdyStyle[1]{%
14262   \glsnoxywarning\GlsAddXdyStyle}
14263 \fi
14264 \ifglxindy
14265   \newcommand*\GlsSetXdyStyles[1]{%
14266     \edef\@xdyrequiredstyles{#1}}
14267 \else
14268   \newcommand*\GlsSetXdyStyles[1]{%
14269     \glsnoxywarning\GlsSetXdyStyles}
14270 \fi
14271 \newcommand*\findrootlanguage{}
14272 \def\@xdy@language#1#2{}
14273 \ifglxindy
14274   \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
14275     \ifglossaryexists{#1}{%
14276       \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
14277     }{%
14278       \PackageError{glossaries}{Can't set language type for
14279         glossary type '#1' --- no such glossary}{%
14280         You have specified a glossary type that doesn't exist}}
14281 \else
14282   \newcommand*\GlsSetXdyLanguage[2][]{%
14283     \glsnoxywarning\GlsSetXdyLanguage}
14284 \fi
14285 \def\@gls@codepage#1#2{}
14286 \ifglxindy
14287   \newcommand*\GlsSetXdyCodePage[1]{%
14288     \renewcommand*\@gls@codepage{#1}%
14289   }
14290   \AtBeginDocument{%
14291     \ifx\@gls@codepage\@empty
14292       \@ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
14293     \fi
14294   }
14295 \else
14296   \newcommand*\GlsSetXdyCodePage[1]{%
14297     \glsnoxywarning\GlsSetXdyCodePage}
14298 \fi
14299 \ifglxindy
14300   \ifglxindy@glxnumbers
14301     \def\@xdylettergroups{(define-letter-group
14302       \string"glxnumbers\string"^^J\space\space\space
14303       :prefixes (\string"0\string" \string"1\string"
14304       \string"2\string" \string"3\string" \string"4\string"
14305       \string"5\string" \string"6\string" \string"7\string"
14306       \string"8\string" \string"9\string")^^J\space\space\space
14307       \@xdynumbergrouporder)}
14308   \else
14309     \def\@xdylettergroups{}

```

```

14310 \fi
14311 \fi
14312 \newcommand*\GlsAddLetterGroup[2]{%
14313   \expandafter\toks@\expandafter{\@xdylettergroups}%
14314   \protected@edef\@xdylettergroups{\the\toks@^^J%
14315     (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
14316   }%
14317 \newcommand*\forallglossaries}[3][\@glo@types]{%
14318   \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
14319 }
14320
14321 \newcommand*\forallacronyms}[2]{%
14322   \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
14323 }
14324 \newcommand*\forallglsentries}[3][\glsdefaulttype]{%
14325   \edef\@glo@list{\csname glolist@#1\endcsname}%
14326   \@for#2:=\@glo@list\do
14327   {%
14328     \ifdefempty{#2}{#3}%
14329   }%
14330 }
14331 \newcommand*\forallglsentries}[3][\@glo@types]{%
14332   \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
14333   {%
14334     \forallglsentries[\@this@glo@]{#2}{#3}%
14335   }%
14336 }
14337 \newcommand{\ifglossaryexists}{%
14338   \@ifstar\s@ifglossaryexists\@ifglossaryexists
14339 }
14340 \newcommand{\@ifglossaryexists}[3]{%
14341   \ifcsundef{@glo@type@#1@out}{#3}{#2}%
14342 }
14343 \newcommand{\s@ifglossaryexists}[3]{%
14344   \ifcsundef{glolist@#1}{#3}{#2}%
14345 }
14346 \newcommand*\glsdetoklabel}[1]{#1}
14347 \newcommand{\ifglsentryexists}[3]{%
14348   \ifcsundef{glo@glsdetoklabel{#1}@name}{#3}{#2}%
14349 }
14350 \newcommand*\ifglsused}[3]{%
14351   \ifbool{glo@glsdetoklabel{#1}@flag}{#2}{#3}%
14352 }
14353 \newcommand{\glsdoifexists}[2]{%
14354   \ifglsentryexists{#1}{#2}{%
14355     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
14356       has not been defined}{You need to define a glossary entry before you
14357       can use it.}}%
14358 }

```

```

14359 \newcommand{\glsdoifnoexists}[2]{%
14360   \ifglentryexists{#1}{%
14361     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
14362       been defined}{-}{#2}%
14363 }
14364 \newcommand{\glsdoifexistsorwarn}[2]{%
14365   \ifglentryexists{#1}{#2}{%
14366     \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
14367       has not been defined}%
14368   }%
14369 }
14370 \newcommand{\glsdoifexistsordo}[3]{%
14371   \ifglentryexists{#1}{#2}{%
14372     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
14373       has not been defined}{You need to define a glossary entry before you
14374       can use it.}%
14375     #3%
14376   }%
14377 }
14378 \newcommand{\doifglossarynoexistsordo}[3]{%
14379   \ifglossaryexists*{#1}%
14380   {%
14381     \PackageError{glossaries}{Glossary type ‘#1’ already exists}{-}%
14382     #3%
14383   }%
14384   {#2}%
14385 }
14386 \newrobustcmd{\ifglshaschildren}[3]{%
14387   \glsdoifexists{#1}%
14388   {%
14389     \def\do@glshaschildren{#3}%
14390     \edef\@gls@thislabel{\glsdetoklabel{#1}}%
14391     \expandafter\forlentries\expandafter
14392       [\csname glo@\@gls@thislabel @type\endcsname]
14393     {\glo@label}%
14394     {%
14395       \letcs\glo@parent{glo@\glo@label @parent}%
14396       \ifdefequal\@gls@thislabel\glo@parent
14397       {%
14398         \def\do@glshaschildren{#2}%
14399         \@endfortrue
14400       }%
14401       }%
14402     }%
14403     \do@glshaschildren
14404   }%
14405 }
14406 \newcommand{\ifglshasparent}[3]{%
14407   \glsdoifexists{#1}%

```



```

14408 {%
14409   \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
14410 }%
14411 }
14412 \newcommand*{\ifglshasdesc}[3]{%
14413   \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
14414   {#3}%
14415   {#2}%
14416 }
14417 \newcommand*{\ifglsdescsuppressed}[3]{%
14418   \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
14419   {#2}%
14420   {#3}%
14421 }
14422 \newrobustcmd*{\ifglshassymbol}[3]{%
14423   \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
14424   \ifdefempty\@glo@symbol
14425   {#3}%
14426   {%
14427     \ifdefequal\@glo@symbol\@gls@default@value
14428     {#3}%
14429     {#2}%
14430   }%
14431 }
14432 \newrobustcmd*{\ifglshaslong}[3]{%
14433   \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
14434   \ifdefempty\@glo@long
14435   {#3}%
14436   {%
14437     \ifdefequal\@glo@long\@gls@default@value
14438     {#3}%
14439     {#2}%
14440   }%
14441 }
14442 \newrobustcmd*{\ifglshasshort}[3]{%
14443   \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
14444   \ifdefempty\@glo@short
14445   {#3}%
14446   {%
14447     \ifdefequal\@glo@short\@gls@default@value
14448     {#3}%
14449     {#2}%
14450   }%
14451 }
14452 \newrobustcmd*{\ifglshasfield}[4]{%
14453   \glsdoifexists{#2}%
14454   {%
14455     \letcs{\@glo@thisvalue}{glo@\glsdetoklabel{#2}@#1}%
14456     \ifdef\@glo@thisvalue

```

```

14457 {%
14458   \ifdefempty\@glo@thisvalue
14459   {%
14460     #4%
14461   }%
14462   {%
14463     \ifdefequal\@glo@thisvalue\@gls@default@value
14464     {%
14465       #4%
14466     }%
14467     {%
14468       \let\glscurrentfieldvalue\@glo@thisvalue
14469       #3%
14470     }%
14471   }%
14472 }%
14473 {%
14474   \@gls@fetchfield{\@gls@thisfield}{#1}%
14475   \ifdef\@gls@thisfield
14476   {%
14477     \letcs{\@glo@thisvalue}{glo\@glsdetoklabel{#2}\@gls@thisfield}%
14478     \ifdefempty\@glo@thisvalue
14479     {%
14480       #4%
14481     }%
14482     {%
14483       \ifdefequal\@glo@thisvalue\@gls@default@value
14484       {%
14485         #4%
14486       }%
14487       {%
14488         \let\glscurrentfieldvalue\@glo@thisvalue
14489         #3%
14490       }%
14491     }%
14492   }%
14493   {%
14494     \GlossariesWarning{Unknown entry field '#1'}%
14495     #4%
14496   }%
14497 }%
14498 }%
14499 }
14500 \newcommand*{\glscurrentfieldvalue}{}
14501 \newcommand*{\@glo@types}{,}
14502 \newcommand*{\@gls@provide@newglossary}%
14503   \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
14504   \let\@gls@provide@newglossary\relax
14505 }

```

```

14506 \newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
14507   \csgdef{gls@#1@entryfmt}{#2}%
14508 }
14509 \newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
14510 \newcommand*{\@gls@forbidtext}[1]{%
14511   \ifboolexpr{test {\ifdefstring{#1}{tex}}
14512     or test {\ifdefstring{#1}{TEX}}}
14513   {%
14514     \def#1{nottex}%
14515     \PackageError{glossaries}%
14516       {Forbidden '.tex' extension replaced with '.nottex'}%
14517       {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
14518         Don't use '.tex' as an extension for a temporary file.}%
14519   }%
14520   {%
14521   }%
14522 }
14523 \newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
14524 \def\@gls@gobbleopt[#1]{%
14525 \newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
14526 \newcommand*{\s@newglossary}[2]{%
14527   \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
14528 }
14529 \newcommand*{\ns@newglossary}[5][glg]{%
14530   \doifglossarynoexistsordo{#2}%
14531   {%
14532     \ifundef\glsdefaulttype
14533     {%
14534       \gdef\glsdefaulttype{#2}%
14535     }{}%
14536     \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
14537     \expandafter\gdef\csname glolist@#2\endcsname{,}%
14538     \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
14539     \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
14540     \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
14541     \expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
14542     \expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
14543     \expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname
14544     \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
14545     \@gls@provide@newglossary
14546     \protected@write\auxout{{\string\@newglossary{#2}{#1}{#3}{#4}}}%
14547     \ifcsundef{gls@#2@entryfmt}%
14548     {%
14549       \defglsentryfmt[#2]{\glsentryfmt}%
14550     }%
14551     {}%
14552     \@gls@defsortcount{#2}%
14553     \@ifnextchar[{\@gls@setcounter{#2}}%
14554       {\@gls@setcounter{#2}[\glscounter]}%

```

```

14555 }%
14556 {%
14557   \gls@gobbleopt
14558 }%
14559 }
14560 \newcommand*{\altnewglossary}[3]{%
14561   \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
14562 }
14563 \@onlypreamble{\newglossary}
14564 \@onlypremakeg\newglossary
14565 \newcommand*{\@newglossary}[4]{%
14566   \def\@gls@setcounter#1[#2]{%
14567     \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
14568     \ifglxindy
14569       \GlsAddXdyCounters{#2}%
14570     \fi
14571   }
14572 \newcommand*{\@gls@getcounter}[1]{%
14573   \csname @glotype@#1@counter\endcsname
14574 }
14575 \glsdefmain
14576 \@gls@do@acronymsdef
14577 \@gls@do@symbolsdef
14578 \@gls@do@numbersdef
14579 \@gls@do@indexdef
14580 \newcommand*{\newignoredglossary}[1]{%
14581   \ifdefempty\@ignored@glossaries
14582     {%
14583       \edef\@ignored@glossaries{#1}%
14584     }%
14585     {%
14586       \eappto\@ignored@glossaries{, #1}%
14587     }%
14588     \csgdef{glolist@#1}{,}%
14589     \ifcsundef{gls@#1@entryfmt}%
14590     {%
14591       \defglsentryfmt[#1]{\glsentryfmt}%
14592     }%
14593     {}%
14594     \ifdefempty\@gls@nohyperlist
14595     {%
14596       \renewcommand*{\@gls@nohyperlist}{#1}%
14597     }%
14598     {%
14599       \eappto\@gls@nohyperlist{, #1}%
14600     }%
14601   }
14602 \newcommand*{\@ignored@glossaries}{}
14603 \newcommand*{\ifignoredglossary}[3]{%

```

```

14604 \edef\@gls@igtype{#1}%
14605 \expandafter\DTLifinlist\expandafter
14606   {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
14607 }
14608 \define@key{glossentry}{name}{%
14609 \def\@glo@name{#1}%
14610 }
14611 \define@key{glossentry}{description}{%
14612 \def\@glo@desc{#1}%
14613 }
14614 \define@key{glossentry}{descriptionplural}{%
14615 \def\@glo@descplural{#1}%
14616 }
14617 \define@key{glossentry}{sort}{%
14618 \def\@glo@sort{#1}}
14619 \define@key{glossentry}{text}{%
14620 \def\@glo@text{#1}%
14621 }
14622 \define@key{glossentry}{plural}{%
14623 \def\@glo@plural{#1}%
14624 }
14625 \define@key{glossentry}{first}{%
14626 \def\@glo@first{#1}%
14627 }
14628 \define@key{glossentry}{firstplural}{%
14629 \def\@glo@firstplural{#1}%
14630 }
14631 \newcommand*{\@gls@default@value}{\relax}
14632 \define@key{glossentry}{symbol}{%
14633 \def\@glo@symbol{#1}%
14634 }
14635 \define@key{glossentry}{symbolplural}{%
14636 \def\@glo@symbolplural{#1}%
14637 }
14638 \define@key{glossentry}{type}{%
14639 \def\@glo@type{#1}}
14640 \define@key{glossentry}{counter}{%
14641 \ifcsundef{c@#1}%
14642   {%
14643     \PackageError{glossaries}%
14644     {There is no counter called ‘#1’}%
14645     {%
14646       The counter key should have the name of a valid counter
14647       as its value%
14648     }%
14649   }%
14650   {%
14651     \def\@glo@counter{#1}%
14652   }%

```

```

14653 }
14654 \define@key{glossentry}{see}{%
14655   \gls@set@xr@key{see}{\@glo@see}{#1}%
14656 }
14657 \newcommand*{\gls@set@xr@key}[3]{%
14658   \renewcommand*{\gls@xr@key}{#1}%
14659   \gls@checkseeallowed
14660   \def#2{#3}%
14661   \@glo@seeautonumberlist
14662 }
14663 \newcommand*{\gls@xr@key}{see}
14664 \newcommand*{\gls@checkseeallowed}{%
14665   \@gls@see@noindex
14666 }
14667 \newcommand*{\gls@checkseeallowed@preambleonly}{%
14668   \GlossariesWarning{glossaries}%
14669   {'\gls@xr@key' key doesn't have any effect when used in the document
14670     environment. Move the definition to the preamble
14671     after \string\makeglossaries\space
14672     or \string\makenoidxglossaries}%
14673 }
14674 \define@key{glossentry}{parent}{%
14675   \def\@glo@parent{#1}}
14676 \define@choicekey{glossentry}{nonumberlist}{%
14677   [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
14678 }%
14679   \ifcase\gls@nonumberlist@nr\relax
14680     \def\@glo@prefix{\glsnonextpages}%
14681     \@gls@savenonumberlist{true}%
14682   \else
14683     \def\@glo@prefix{\glsnextpages}%
14684     \@gls@savenonumberlist{false}%
14685   \fi
14686 }
14687 \newcommand*{\@gls@savenonumberlist}[1]{%
14688   \newcommand*{\@gls@initnonumberlist}{}%
14689   \newcommand*{\@gls@storenonumberlist}[1]{%
14690     \newcommand*{\@gls@enablesavenonumberlist}{%
14691       \renewcommand*{\@gls@initnonumberlist}{%
14692         \undef\@glo@nonumberlist
14693       }%
14694       \renewcommand*{\@gls@savenonumberlist}[1]{%
14695         \def\@glo@nonumberlist{##1}%
14696       }%
14697       \renewcommand*{\@gls@storenonumberlist}[1]{%
14698         \ifdef\@glo@nonumberlist
14699           {%
14700             \cslet{glo@\glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
14701           }%

```

```

14702     {}%
14703 }%
14704 \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
14705 }
14706 \define@key{glossentry}{user1}{%
14707   \def\@glo@useri{#1}%
14708 }
14709 \define@key{glossentry}{user2}{%
14710   \def\@glo@userii{#1}%
14711 }
14712 \define@key{glossentry}{user3}{%
14713   \def\@glo@useriii{#1}%
14714 }
14715 \define@key{glossentry}{user4}{%
14716   \def\@glo@useriv{#1}%
14717 }
14718 \define@key{glossentry}{user5}{%
14719   \def\@glo@userv{#1}%
14720 }
14721 \define@key{glossentry}{user6}{%
14722   \def\@glo@uservi{#1}%
14723 }
14724 \define@key{glossentry}{short}{%
14725   \def\@glo@short{#1}%
14726 }
14727 \define@key{glossentry}{shortplural}{%
14728   \def\@glo@shortpl{#1}%
14729 }
14730 \define@key{glossentry}{long}{%
14731   \def\@glo@long{#1}%
14732 }
14733 \define@key{glossentry}{longplural}{%
14734   \def\@glo@longpl{#1}%
14735 }
14736 \newcommand*{\@glsnoname}{%
14737   \PackageError{glossaries}{name key required in
14738     \string\newglossaryentry\space for entry '\@glo@label'}{You
14739     haven't specified the entry name}}
14740 \newcommand*{\@glsnodelsc{%
14741   \PackageError{glossaries}
14742     {%
14743       description key required in \string\newglossaryentry\space
14744       for entry '\@glo@label'%
14745     }%
14746   {%
14747     You haven't specified the entry description%
14748   }%
14749 }%
14750 \newcommand*{\@glsdefaultplural}{%

```

```

14751 \newcommand*{\@gls@missingnumberlist}[1]{%
14752   ??%
14753   \ifglssavenumberlist
14754     \GlossariesWarning{Missing number list for entry ‘#1’.
14755       Maybe makeglossaries + rerun required}%
14756   \else
14757     \PackageError{glossaries}%
14758       {Package option ‘savenumberlist=true’ required}%
14759     {%
14760       You must use the ‘savenumberlist’ package option
14761       to reference location lists.%
14762     }%
14763   \fi
14764 }
14765 \newcommand*{\@gls@defaultsort}{\@gls@name}
14766 \newcount\gls@level
14767 \newcommand{\@gls@noexpand@field}[3]{%
14768   \expandafter\global\expandafter
14769     \let\csname glo@#1@#2\endcsname#3%
14770 }
14771 \newcommand{\@gls@noexpand@fields}[4]{%
14772   \ifcsdef{gls@assign@#3@field}
14773     {%
14774       \ifdefequal{#4}{\@gls@default@value}%
14775       {%
14776         \edef\@gls@value{\expandonce{#1}}%
14777         \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
14778       }%
14779     }%
14780     \csuse{gls@assign@#3@field}{#2}{#4}%
14781   }%
14782 }%
14783 {%
14784   \ifdefequal{#4}{\@gls@default@value}%
14785   {%
14786     \edef\@gls@value{\expandonce{#1}}%
14787     \@gls@noexpand@field{#2}{#3}{\@gls@value}%
14788   }%
14789 }%
14790   \@gls@noexpand@field{#2}{#3}{#4}%
14791 }%
14792 }%
14793 }
14794 \newcommand{\@gls@expand@field}[3]{%
14795   \expandafter
14796     \protected@xdef\csname glo@#1@#2\endcsname{#3}%
14797 }
14798 \newcommand{\@gls@expand@fields}[4]{%
14799   \ifcsdef{gls@assign@#3@field}

```



```

14800 {%
14801   \ifdefequal{#4}{\@gls@default@value}%
14802   {%
14803     \edef\@gls@value{\expandonce{#1}}%
14804     \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
14805   }%
14806   {%
14807     \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
14808     {%
14809       \@gls@expand@field{#2}{#3}{#4}%
14810     }%
14811     {%
14812       \csuse{gls@assign@#3@field}{#2}{#4}%
14813     }%
14814   }%
14815 }%
14816 {%
14817   \ifdefequal{#4}{\@gls@default@value}%
14818   {%
14819     \@gls@expand@field{#2}{#3}{#1}%
14820   }%
14821   {%
14822     \@gls@expand@field{#2}{#3}{#4}%
14823   }%
14824 }%
14825 }
14826 \def\@gls@expandonce{\expandonce}
14827 \def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
14828   \def\@gls@tmp{#1}%
14829   \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
14830 }
14831 \let\gls@assign@field\@gls@expand@fields
14832 \newcommand*\@gls@expandfields{%
14833   \let\gls@assign@field\@gls@expand@fields
14834 }
14835 \newcommand*\@gls@noexpandfields{%
14836   \let\gls@assign@field\@gls@noexpand@fields
14837 }
14838 \newrobustcmd{\newglossaryentry}[2]{%
14839   \glsdoifnoexists{#1}%
14840   {%
14841     \gls@defglossaryentry{#1}{#2}%
14842   }%
14843 }
14844 \newcommand*\@gls@defdocnewglossaryentry{%
14845   \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
14846   \let\newglossaryentry\new@glossaryentry
14847 }
14848 \newrobustcmd{\provideglossaryentry}[2]{%

```

```

14849 \ifglentryexists{#1}%
14850 {}%
14851 {%
14852   \gls@defglossaryentry{#1}{#2}%
14853 }%
14854 }
14855 \onlypreamble{\provideglossaryentry}
14856 \newrobustcmd{\newglossaryentry}[2]{%
14857   \ifundef\@gls@deffile
14858   {%
14859     \global\newwrite\@gls@deffile
14860     \immediate\openout\@gls@deffile=\jobname.glsdefs
14861   }%
14862   {}%
14863   \ifglentryexists{#1}{}%
14864   {%
14865     \gls@defglossaryentry{#1}{#2}%
14866   }%
14867   \@gls@writedef{#1}%
14868 }
14869 \AtBeginDocument{\gls@begindocdefs}
14870 \AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
14871 \newcommand*\@gls@begindocdefs{%
14872   \@gls@enablesavenonumberlist
14873   \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
14874   \makeatletter
14875   \InputIfFileExists{\jobname.glsdefs}{\@gls@restreat}{\@gls@restreat}
14876   \undef\@gls@restreat
14877   \gls@defdocnewglossaryentry
14878 }
14879 }
14880 \newcommand*\@gls@writedef[1]{%
14881   \immediate\write\@gls@deffile
14882   {%
14883     \string\ifglentryexists{#1}{}\glspercentchar^^J%
14884     \expandafter\@gobble\string{\glspercentchar^^J%
14885       \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
14886       \expandafter\@gobble\string{\glspercentchar%
14887     }%
14888     \@for\@gls@map:=\@gls@keymap\do
14889     {%
14890       \letcs\glo@value{glo\@glsdetoklabel{#1}\@expandafter\@secondoftwo\@gls@map}%
14891       \ifdef\glo@value
14892       {%
14893         \@onelevel@sanitize\glo@value
14894         \immediate\write\@gls@deffile
14895         {%
14896           \expandafter\@firstoftwo\@gls@map
14897           =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%

```

```

14898      \glspercentchar
14899    }%
14900  }%
14901  {}%
14902 }%
14903 \glswritedefhook
14904 \immediate\write\@gls@deffile
14905 {%
14906      \glspercentchar^^J%
14907      \expandafter\@gobble\string\}\glspercentchar^^J%
14908      \expandafter\@gobble\string\}\glspercentchar%
14909  }%
14910 }
14911 \newcommand*{\@gls@keymap}{%
14912   {name}{name},%
14913   {sort}{sortvalue},% unescaped sort value
14914   {type}{type},%
14915   {first}{first},%
14916   {firstplural}{firstpl},%
14917   {text}{text},%
14918   {plural}{plural},%
14919   {description}{desc},%
14920   {descriptionplural}{descplural},%
14921   {symbol}{symbol},%
14922   {symbolplural}{symbolplural},%
14923   {user1}{useri},%
14924   {user2}{userii},%
14925   {user3}{useriii},%
14926   {user4}{useriv},%
14927   {user5}{userv},%
14928   {user6}{uservi},%
14929   {long}{long},%
14930   {longplural}{longpl},%
14931   {short}{short},%
14932   {shortplural}{shortpl},%
14933   {counter}{counter},%
14934   {parent}{parent}%
14935 }
14936 \newcommand*{\@gls@fetchfield}[2]{%
14937   \edef\@gls@thisval{#2}%
14938   \@for\@gls@map:=\@gls@keymap\do{%
14939     \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
14940     \ifdefequal{\@this@key}{\@gls@thisval}%
14941     {%
14942       \edef#1{\expandafter\@secondoftwo\@gls@map}%
14943       \@endfortrue
14944     }%
14945   }%
14946 }%

```

```

14947 }
14948 \newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
14949 \newcommand*{\@sglsaddstoragekey}[1]{%
14950   \key@ifundefined{glossentry}{#1}%
14951   {%
14952     \expandafter\newcommand\expandafter*\expandafter
14953       {\csname gls@assign@#1@field\endcsname}[2]{%
14954         \@gls@expand@field{##1}{#1}{##2}%
14955       }%
14956   }%
14957   {}%
14958   \@glsaddstoragekey{#1}%
14959 }
14960 \newcommand*{\@glsaddstoragekey}[3]{%
14961   \key@ifundefined{glossentry}{#1}%
14962   {%
14963     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
14964     \appto\@gls@keymap{,{#1}{#1}}%
14965     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
14966     \appto\@newglossaryentryposthook{%
14967       \letcs{\@glo@tmp}{@glo@#1}%
14968       \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
14969     }%
14970     \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
14971   }%
14972   {%
14973     \PackageError{glossaries}{Key ‘#1’ already exists}{}%
14974   }%
14975 }
14976 \newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
14977 \newcommand*{\@sglsaddkey}[1]{%
14978   \key@ifundefined{glossentry}{#1}%
14979   {%
14980     \expandafter\newcommand\expandafter*\expandafter
14981       {\csname gls@assign@#1@field\endcsname}[2]{%
14982         \@gls@expand@field{##1}{#1}{##2}%
14983       }%
14984     }%
14985     {}%
14986     \@glsaddkey{#1}%
14987 }
14988 \newcommand*{\@glsaddkey}[7]{%
14989   \key@ifundefined{glossentry}{#1}%
14990   {%
14991     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
14992     \appto\@gls@keymap{,{#1}{#1}}%
14993     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
14994     \appto\@newglossaryentryposthook{%
14995       \letcs{\@glo@tmp}{@glo@#1}%

```

```

14996     \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
14997 }%
14998 \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
14999 \newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
15000 \ifcsdef{@gls@user@#1@}%
15001 {%
15002     \PackageError{glossaries}%
15003     {Can't define '\string#5' as helper command
15004     '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
15005     }%
15006 }%
15007 {%
15008     \expandafter\newcommand\expandafter*\expandafter
15009     {\csname @gls@user@#1\endcsname}[2][\%
15010         \new@ifnextchar[%
15011             {\csuse{@gls@user@#1@}{##1}{##2}}%
15012             {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
15013     \csdef{@gls@user@#1@}##1##2[##3]{%
15014         \@gls@field@link{##1}{##2}{#3{##2}##3}%
15015     }%
15016     \newrobustcmd*{#5}{%
15017         \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
15018     }%
15019     \ifcsdef{@Gls@user@#1@}%
15020     {%
15021         \PackageError{glossaries}%
15022         {Can't define '\string#6' as helper command
15023         '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
15024         }%
15025     }%
15026     {%
15027         \expandafter\newcommand\expandafter*\expandafter
15028         {\csname @Gls@user@#1\endcsname}[2][\%
15029             \new@ifnextchar[%
15030                 {\csuse{@Gls@user@#1@}{##1}{##2}}%
15031                 {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
15032         \csdef{@Gls@user@#1@}##1##2[##3]{%
15033             \@gls@field@link{##1}{##2}{#4{##2}##3}%
15034         }%
15035         \newrobustcmd*{#6}{%
15036             \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
15037         }%
15038         \ifcsdef{@GLS@user@#1@}%
15039         {%
15040             \PackageError{glossaries}%
15041             {Can't define '\string#7' as helper command
15042             '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
15043             }%
15044         }%

```

```

15045   {%
15046     \expandafter\newcommand\expandafter*\expandafter
15047       {\csname @GLS@user@#1\endcsname}[2] [] {%
15048         \new@ifnextchar[%
15049           {\csuse{@GLS@user@#1@}{##1}{##2}}}%
15050         {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
15051     \csdef{@GLS@user@#1@}##1##2[##3]{%
15052       \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}}%
15053   }%
15054   \newrobustcmd*{#7}{%
15055     \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
15056   }%
15057 }%
15058 {%
15059   \PackageError{glossaries}{Key ‘#1’ already exists}{}%
15060 }%
15061 }
15062 \newcommand{\glsfieldxdef}[3]{%
15063   \glsdoifexists{#1}%
15064   {%
15065     \edef\@glo@label{\glsdetoklabel{#1}}%
15066     \ifcsdef{glo@\@glo@label @#2}%
15067     {%
15068       \protected@csxdef{glo@\@glo@label @#2}{#3}%
15069     }%
15070     {%
15071       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15072     }%
15073   }%
15074 }
15075 \newcommand{\glsfielddedef}[3]{%
15076   \glsdoifexists{#1}%
15077   {%
15078     \edef\@glo@label{\glsdetoklabel{#1}}%
15079     \ifcsdef{glo@\@glo@label @#2}%
15080     {%
15081       \protected@csedef{glo@\@glo@label @#2}{#3}%
15082     }%
15083     {%
15084       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15085     }%
15086   }%
15087 }
15088 \newcommand{\glsfieldgdef}[3]{%
15089   \glsdoifexists{#1}%
15090   {%
15091     \edef\@glo@label{\glsdetoklabel{#1}}%
15092     \ifcsdef{glo@\@glo@label @#2}%
15093     {%

```

```

15094     \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
15095 }%
15096 {%
15097     \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15098 }%
15099 }%
15100 }
15101 \newcommand{\glsfielddef}[3]{%
15102   \glsdoifexists{#1}%
15103   {%
15104     \edef\@glo@label{\glsdetoklabel{#1}}%
15105     \ifcsdef{glo@\@glo@label @#2}%
15106     {%
15107       \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
15108     }%
15109     {%
15110       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15111     }%
15112   }%
15113 }
15114 \newcommand{\glsfieldfetch}[3]{%
15115   \glsdoifexists{#1}%
15116   {%
15117     \edef\@glo@label{\glsdetoklabel{#1}}%
15118     \ifcsdef{glo@\@glo@label @#2}%
15119     {%
15120       \letcs#3{glo@\@glo@label @#2}%
15121     }%
15122     {%
15123       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15124     }%
15125   }%
15126 }
15127 \newcommand{\ifglsfieldeq}[5]{%
15128   \glsdoifexists{#1}%
15129   {%
15130     \edef\@glo@label{\glsdetoklabel{#1}}%
15131     \ifcsdef{glo@\@glo@label @#2}%
15132     {%
15133       \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
15134     }%
15135     {%
15136       \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15137     }%
15138   }%
15139 }
15140 \newcommand{\ifglsfielddefeq}[5]{%
15141   \glsdoifexists{#1}%
15142   {%

```

```

15143 \edef\@glo@label{\glsdetoklabel{#1}}%
15144 \ifcsdef{glo@\@glo@label @#2}%
15145 {%
15146     \expandafter\ifdefstrequal
15147     \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
15148 }%
15149 {%
15150     \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15151 }%
15152 }%
15153 }
15154 \newcommand{\ifglsfieldcseq}[5]{%
15155     \glsdoifexists{#1}%
15156     {%
15157         \edef\@glo@label{\glsdetoklabel{#1}}%
15158         \ifcsdef{glo@\@glo@label @#2}%
15159         {%
15160             \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
15161         }%
15162         {%
15163             \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15164         }%
15165     }%
15166 }
15167 \newcommand*\@gls@writedefhook{}
15168 \newcommand*\@gls@assign@desc}[1]{%
15169     \gls@assign@field{#1}{desc}{\@glo@desc}%
15170     \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
15171 }
15172 \newcommand{\longnewglossaryentry}[3]{%
15173     \glsdoifnoexists{#1}%
15174     {%
15175         \bgroup
15176         \let\@org@newglossaryentryprehook\@newglossaryentryprehook
15177         \long\def\@newglossaryentryprehook{%
15178             \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
15179             \@org@newglossaryentryprehook
15180         }%
15181         \renewcommand*\@gls@assign@desc}[1]{%
15182             \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
15183             \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
15184         }
15185         \gls@defglossaryentry{#1}{#2}%
15186     \egroup
15187 }
15188 }
15189 \@onlypreamble{\longnewglossaryentry}
15190 \newcommand{\longprovideglossaryentry}[3]{%
15191     \ifglsentryexists{#1}{}%

```



```

15192 {\longnewglossaryentry{#1}{#2}{#3}}%
15193 }
15194 \@onlypreamble{\longprovideglossaryentry}
15195 \newcommand{\gls@defglossaryentry}[2]{%
15196   \let\GlsSetQuote\gls@nosetquote
15197   \edef\@glo@label{\glsdetoklabel{#1}}%
15198   \let\glslabel\@glo@label
15199   \let\@glo@name\@glsnname
15200   \let\@glo@desc\@glsndesc
15201   \let\@glo@descplural\@gls@default@value
15202   \let\@glo@type\@gls@default@value
15203   \let\@glo@symbol\@gls@default@value
15204   \let\@glo@symbolplural\@gls@default@value
15205   \let\@glo@text\@gls@default@value
15206   \let\@glo@plural\@gls@default@value
15207   \let\@glo@first\@gls@default@value
15208   \let\@glo@firstplural\@gls@default@value
15209   \let\@glo@sort\@gls@default@value
15210   \let\@glo@counter\@gls@default@value
15211   \def\@glo@see{}%
15212   \def\@glo@parent{}%
15213   \def\@glo@prefix{}%
15214   \@gls@initnonnumberlist
15215   \def\@glo@useri{}%
15216   \def\@glo@userii{}%
15217   \def\@glo@useriii{}%
15218   \def\@glo@useriv{}%
15219   \def\@glo@userv{}%
15220   \def\@glo@uservi{}%
15221   \def\@glo@short{}%
15222   \def\@glo@shortpl{}%
15223   \def\@glo@long{}%
15224   \def\@glo@longpl{}%
15225   \@newglossaryentryprehook
15226   \setkeys{glossentry}{#2}%
15227   \ifundef\glsdefaulttype
15228   {%
15229     \PackageError{glossaries}%
15230     {No default glossary type (have you used ‘nomain’ by mistake?)}%
15231     {If you use package option ‘nomain’ you must define
15232      a new glossary before you can define entries}%
15233   }%
15234   {%
15235     \gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
15236     \edef\@glo@type{\glsentrytype{\@glo@label}}%
15237     \ifcsundef{glolist@\@glo@type}%
15238     {%
15239       \PackageError{glossaries}%
15240       {Glossary type ‘\@glo@type’ has not been defined}%

```

```

15241      {You need to define a new glossary type, before making entries
15242      in it}%
15243  }%
15244  {%
15245      \ifignoredglossary\@glo@type
15246      {%
15247          \ifx\@glo@desc\@glsnodesc
15248          \let\@glo@desc\@empty
15249          \fi
15250      }%
15251      {%
15252      }%
15253      \protected@edef\@glo@list@{\csname glo@list@\@glo@type\endcsname}%
15254      \expandafter\xdef\csname glo@list@\@glo@type\endcsname{%
15255          \@glo@list@{\@glo@label},}%
15256  }%
15257  \gls@level=0\relax
15258  \ifx\@glo@parent\@empty
15259      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15260  \else
15261      \ifdefequal\@glo@label\@glo@parent%
15262      {%
15263          \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
15264          \def\@glo@parent{}%
15265          \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15266      }%
15267      {%
15268          \ifglsentryexists{\@glo@parent}%
15269          {%
15270              \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
15271                  \@glo@parent}%
15272              \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
15273              \advance\gls@level by 1\relax
15274              \ifx\@glo@name\@glsnoname
15275                  \expandafter\let\expandafter\@glo@name
15276                  \csname glo@\@glo@parent @name\endcsname
15277                  \ifx\@glo@plural\@gls@default@value
15278                      \expandafter\let\expandafter\@glo@plural
15279                      \csname glo@\@glo@parent @plural\endcsname
15280                  \fi
15281                  \fi
15282              }%
15283              {%
15284                  \PackageError{glossaries}%
15285                  {%
15286                      Invalid parent ‘\@glo@parent’
15287                      for entry ‘\@glo@label’ - parent doesn’t exist%
15288                  }%
15289                  {%

```

```

15290         Parent entries must be defined before their children%
15291     }%
15292     \def\@glo@parent{}%
15293     \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15294 }%
15295 }%
15296 \fi
15297 \expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
15298 \gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
15299 \letcs\@glo@sort{glo@\@glo@label @sortvalue}%
15300 \gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
15301 \expandafter\gls@assign@field\expandafter
15302     {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
15303     {\@glo@label}{plural}{\@glo@plural}%
15304 \expandafter\gls@assign@field\expandafter
15305     {\csname glo@\@glo@label @text\endcsname}%
15306     {\@glo@label}{first}{\@glo@first}%
15307 \ifx\@glo@first\@gls@default@value
15308     \expandafter\gls@assign@field\expandafter
15309         {\csname glo@\@glo@label @plural\endcsname}%
15310         {\@glo@label}{firstpl}{\@glo@firstplural}%
15311 \else
15312     \expandafter\gls@assign@field\expandafter
15313         {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
15314         {\@glo@label}{firstpl}{\@glo@firstplural}%
15315 \fi
15316 \ifcsundef{@glotype@\@glo@type @counter}%
15317 {%
15318     \def\@glo@defaultcounter{\glscounter}%
15319 }%
15320 {%
15321     \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
15322 }%
15323 \gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
15324 \gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
15325 \gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
15326 \gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
15327 \gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
15328 \gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
15329 \gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
15330 \gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
15331 \gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
15332 \gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
15333 \gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
15334 \ifx\@glo@name\@gls@name
15335     \@gls@name
15336     \let\@glo@name\@gls@default@value
15337 \fi
15338 \gls@assign@field{}{\@glo@label}{name}{\@glo@name}%

```

```

15339 \ifcsundef{glo@\@glo@label @numberlist}%
15340 {%
15341   \csxdef{glo@\@glo@label @numberlist}{%
15342     \noexpand\@gls@missingnumberlist{\@glo@label}}%
15343  }%
15344   {%
15345     \@gls@storenonumberlist{\@glo@label}%
15346     \def\@glo@@desc{\@glo@first}%
15347     \ifx\@glo@desc\@glo@@desc
15348       \let\@glo@desc\@glo@first
15349     \fi
15350     \ifx\@glo@desc\@glsnodesc
15351       \@glsnodesc
15352       \let\@glo@desc\@gls@default@value
15353     \fi
15354     \@gls@assign@desc{\@glo@label}%
15355     \@gls@defsort{\@glo@type}{\@glo@label}%
15356     \def\@glo@@symbol{\@glo@text}%
15357     \ifx\@glo@symbol\@glo@@symbol
15358       \let\@glo@symbol\@glo@text
15359     \fi
15360     \@gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
15361     \expandafter
15362       \@gls@assign@field\expandafter
15363         {\csname glo@\@glo@label @symbol\endcsname}
15364         {\@glo@label}{symbolplural}{\@glo@symbolplural}%
15365     \expandafter\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
15366       \noexpand\global
15367         \noexpand\let\expandafter\noexpand
15368         \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
15369     }%
15370     \expandafter\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
15371       \noexpand\global
15372         \noexpand\let\expandafter\noexpand
15373         \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
15374     }%
15375     \csname glo@\@glo@label @flagfalse\endcsname
15376     \@glo@autosee
15377     \ifignoredglossary\@glo@type
15378     {%
15379       \csdef{glo@\@glo@label @index}{}%
15380     }
15381     {%
15382       \do@glo@storeentry{\@glo@label}%
15383     }%
15384     \@newglossaryentry@defcounters
15385     \@newglossaryentryposthook
15386 }
15387 \newcommand*{\@glo@autosee}{%

```

```

15388 \ifdefvoid\@glo@see{}%
15389 {%
15390   \protected@edef\@do@glsee{%
15391     \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
15392     \noexpand\expandafter\noexpand\@glsee\noexpand\@glo@list{\@glo@label}}%
15393   \@do@glsee
15394 }%
15395 \@glo@autoseehook
15396 }%
15397 \newcommand*{\@glo@autoseehook}{}
15398 \newcommand*{\@newglossaryentryprehook}{}
15399 \newcommand*{\@newglossaryentryposthook}{}
15400 \newcommand*{\@newglossaryentry@defcounters}{}
15401 \newcommand*{\glsmoveentry}[2]{%
15402   \edef\@glo@thislabel{\glsdetoklabel{#1}}%
15403   \edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
15404   \def\glo@list{,%}%
15405   \forglseentries[\glo@type]{\glo@label}%
15406   {%
15407     \ifdefequal\@glo@thislabel\glo@label
15408       {\eappto\glo@list{\glo@label,}}%
15409     }%
15410   \cslet{glolist@\glo@type}{\glo@list}%
15411   \csdef{glo@\@glo@thislabel @type}{#2}%
15412 }
15413 \ifglxindy
15414   \newcommand*{\@glossaryentryfield}{\string\glossentry}
15415 \else
15416   \newcommand*{\@glossaryentryfield}{\string\glossentry}
15417 \fi
15418 \ifglxindy
15419   \newcommand*{\@glossarysubentryfield}{%
15420     \string\subglossentry}
15421 \else
15422   \newcommand*{\@glossarysubentryfield}{%
15423     \string\subglossentry}
15424 \fi
15425 \newcommand{\@glo@storeentry}[1]{%
15426   \edef\@glo@esclabel{#1}%
15427   \@gls@checkmkidxchars\@glo@esclabel
15428   \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
15429   \@gls@checkmkidxchars\@glo@sort
15430   \@gls@checkmkidxchars\@glo@prefix
15431   \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
15432   \ifglxindy
15433     \ifx\@glo@parent\@empty
15434       \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15435         (\string"\@glo@sort\string" %
15436         \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %

```

```

15437     }%
15438   \else
15439     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15440       \csname glo@\@glo@parent @index\endcsname
15441       (\string"\@glo@sort\string" %
15442       \string"\@glo@prefix\@glossarysubentryfield
15443       {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
15444     }%
15445   \fi
15446 \else
15447   \ifx\@glo@parent\@empty
15448     \@onelevel@sanitize\@glo@prefix
15449     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15450       \@glo@sort\@gls@actualchar\@glo@prefix
15451       \@glossaryentryfield{\@glo@esclabel}%
15452     }%
15453   \else
15454     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15455       \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
15456       \@glo@sort\@gls@actualchar\@glo@prefix
15457       \@glossarysubentryfield
15458       {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
15459     }%
15460   \fi
15461 \fi
15462 }
15463 \AtBeginDocument{%
15464   \@ifpackageloaded{amsmath}%
15465   {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
15466   {}%
15467 }
15468 \newcommand*{\@gls@ifnotmeasuring}[1]{%
15469   \ifmeasuring@
15470   \else
15471     #1%
15472   \fi
15473 }
15474 \newcommand*\gls@ifnotmeasuring[1]{#1}
15475 \def\@gls@patchtabularx#1\hbox#2#3!!{%
15476   \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
15477 }
15478 \newcommand*\glspatchtabularx{%
15479   \ifdef\TX@trial
15480   {%
15481     \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
15482     \let\glspatchtabularx\relax
15483   }%
15484   {}%
15485 }

```

```

15486 \newcommand*{\glsreset}[1]{%
15487   \gls@ifnotmeasuring
15488   {%
15489     \glsdoifexists{#1}%
15490     {%
15491       \@glsreset{#1}%
15492     }%
15493   }%
15494 }
15495 \newcommand*{\glslocalreset}[1]{%
15496   \gls@ifnotmeasuring
15497   {%
15498     \glsdoifexists{#1}%
15499     {%
15500       \@glslocalreset{#1}%
15501     }%
15502   }%
15503 }
15504 \newcommand*{\glsunset}[1]{%
15505   \gls@ifnotmeasuring
15506   {%
15507     \glsdoifexists{#1}%
15508     {%
15509       \@glsunset{#1}%
15510     }%
15511   }%
15512 }
15513 \newcommand*{\glslocalunset}[1]{%
15514   \gls@ifnotmeasuring
15515   {%
15516     \glsdoifexists{#1}%
15517     {%
15518       \@glslocalunset{#1}%
15519     }%
15520   }%
15521 }
15522 \newcommand*{\@glslocalunset}{\@@glslocalunset}
15523 \newcommand*{\@@glslocalunset}[1]{%
15524   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
15525 }
15526 \newcommand*{\@glsunset}{\@@glsunset}
15527 \newcommand*{\@@glsunset}[1]{%
15528   \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
15529 }
15530 \newcommand*{\@glslocalreset}{\@@glslocalreset}
15531 \newcommand*{\@@glslocalreset}[1]{%
15532   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
15533 }
15534 \newcommand*{\@glsreset}{\@@glsreset}

```

```

15535 \newcommand*{\@glsreset}[1]{%
15536   \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
15537 }
15538 \newcommand*{\glsresetall}[1][\@glo@types]{%
15539   \forallglsentries[#1]{\@glsentry}%
15540   {%
15541     \glsreset{\@glsentry}%
15542   }%
15543 }
15544 \newcommand*{\glslocalresetall}[1][\@glo@types]{%
15545   \forallglsentries[#1]{\@glsentry}%
15546   {%
15547     \glslocalreset{\@glsentry}%
15548   }%
15549 }
15550 \newcommand*{\glsunsetall}[1][\@glo@types]{%
15551   \forallglsentries[#1]{\@glsentry}%
15552   {%
15553     \glsunset{\@glsentry}%
15554   }%
15555 }
15556 \newcommand*{\glslocalunsetall}[1][\@glo@types]{%
15557   \forallglsentries[#1]{\@glsentry}%
15558   {%
15559     \glslocalunset{\@glsentry}%
15560   }%
15561 }
15562 \newcommand*{\@@newglossaryentry@defcounters}{%
15563   \csdef{glo@\@glo@label @currcount}{0}%
15564   \csdef{glo@\@glo@label @prevcount}{0}%
15565 }
15566 \newcommand*{\glsenableentrycount}{%
15567   \let\@newglossaryentry@defcounters\@@newglossaryentry@defcounters
15568   \renewcommand*{\gls@defdocnewglossaryentry}{%
15569     \renewcommand*\newglossaryentry[2]{%
15570       \PackageError{glossaries}{\string\newglossaryentry\space
15571         may only be used in the preamble when entry counting has
15572         been activated}{If you use \string\glsenableentrycount\space
15573         you must place all entry definitions in the preamble not in
15574         the document environment}%
15575     }%
15576   }%
15577   \newcommand*{\glsentrycurrcount}[1]{%
15578     \ifcsundef{glo@glsdetoklabel{##1}@currcount}%
15579     {0}{\@gls@entry@field{##1}{currcount}}%
15580   }%
15581   \newcommand*{\glsentryprevcount}[1]{%
15582     \ifcsundef{glo@glsdetoklabel{##1}@prevcount}%
15583     {0}{\@gls@entry@field{##1}{prevcount}}%

```



```

15584 }%
15585 \renewcommand*{\@glsunset}[1]{%
15586   \@@glsunset{##1}%
15587   \@gls@increment@currcount{##1}%
15588 }%
15589 \renewcommand*{\@glslocalunset}[1]{%
15590   \@@glslocalunset{##1}%
15591   \@gls@local@increment@currcount{##1}%
15592 }%
15593 \renewcommand*{\@glsreset}[1]{%
15594   \@@glsreset{##1}%
15595   \csgdef{glo@glsdetoklabel{##1}@currcount}{0}%
15596 }%
15597 \renewcommand*{\@glslocalreset}[1]{%
15598   \@@glslocalreset{##1}%
15599   \csdef{glo@glsdetoklabel{##1}@currcount}{0}%
15600 }%
15601 \def\@cgl@##1##2[##3]{%
15602   \ifnum\glsentryprevcount{##2}=1\relax
15603     \cgl@format{##2}{##3}%
15604     \glsunset{##2}%
15605   \else
15606     \@gls@{##1}{##2}[##3]%
15607   \fi
15608 }%
15609 \def\@cgl@pl@##1##2[##3]{%
15610   \ifnum\glsentryprevcount{##2}=1\relax
15611     \cgl@pl@format{##2}{##3}%
15612     \glsunset{##2}%
15613   \else
15614     \@gls@pl@{##1}{##2}[##3]%
15615   \fi
15616 }%
15617 \def\@cGls@##1##2[##3]{%
15618   \ifnum\glsentryprevcount{##2}=1\relax
15619     \cGls@format{##2}{##3}%
15620     \glsunset{##2}%
15621   \else
15622     \@Gls@{##1}{##2}[##3]%
15623   \fi
15624 }%
15625 \def\@cGls@pl@##1##2[##3]{%
15626   \ifnum\glsentryprevcount{##2}=1\relax
15627     \cGls@pl@format{##2}{##3}%
15628     \glsunset{##2}%
15629   \else
15630     \@Gls@pl@{##1}{##2}[##3]%
15631   \fi
15632 }%

```

```

15633 \AtEndDocument{\@gls@write@entrycounts}%
15634 \renewcommand*{\@gls@entry@count}[2]{%
15635   \csgdef{glo@glstdetoklabel{##1}@prevcount}{##2}%
15636 }%
15637 \let\glsenableentrycount\relax
15638 }
15639 \@onlypreamble\glsenableentrycount
15640 \newcommand*{\@gls@increment@currcount}[1]{%
15641   \csxdef{glo@glstdetoklabel{##1}@currcount}{%
15642     \number\numexpr\glsentrycurrcount{##1}+1}%
15643 }
15644 \newcommand*{\@gls@local@increment@currcount}[1]{%
15645   \csedef{glo@glstdetoklabel{##1}@currcount}{%
15646     \number\numexpr\glsentrycurrcount{##1}+1}%
15647 }
15648 \newcommand*{\@gls@write@entrycounts}{%
15649   \immediate\write\@auxout
15650     {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
15651   \forallglsentries{\@glsentry}{%
15652     \ifglsused{\@glsentry}%
15653     {\immediate\write\@auxout
15654       {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
15655     }%
15656   }%
15657 }
15658 \newcommand*{\@gls@entry@count}[2]{}
15659 \newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
15660 \newcommand*{\@cgl}[2][]{%
15661   \new@ifnextchar[{\@cgl@{##1}{##2}}{\@cgl@{##1}{##2}[]}%
15662 }
15663 \def\@cgl@#1#2[#3]{%
15664   \GlossariesWarning{\string\cgl\space is defaulting to
15665     \string\gls\space since you haven't enabled entry counting}%
15666   \@gls@{##1}{##2}[##3]}
15667 }
15668 \newcommand*{\cglformat}[2]{%
15669   \ifglshaslong{##1}{\glsentrylong{##1}}{\glsentryfirst{##1}}#2%
15670 }
15671 \newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
15672 \newcommand*{\@cGls}[2][]{%
15673   \new@ifnextchar[{\@cGls@{##1}{##2}}{\@cGls@{##1}{##2}[]}%
15674 }
15675 \def\@cGls@#1#2[#3]{%
15676   \GlossariesWarning{\string\cGls\space is defaulting to
15677     \string\Gls\space since you haven't enabled entry counting}%
15678   \@Gls@{##1}{##2}[##3]}
15679 }
15680 \newcommand*{\cGlsformat}[2]{%
15681   \ifglshaslong{##1}{\Glsentrylong{##1}}{\Glsentryfirst{##1}}#2%

```

```

15682 }
15683 \newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
15684 \newcommand*{\@cglspl}[2][{}]{%
15685   \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}[{}]}%
15686 }
15687 \def\@cglspl@#1#2[#3]{%
15688   \GlossariesWarning{\string\cglspl\space is defaulting to
15689     \string\glspl\space since you haven't enabled entry counting}%
15690   \@glspl@{#1}{#2}[#3]%
15691 }
15692 \newcommand*{\cglsplformat}[2]{%
15693   \ifglshaslong{#1}{\glentrylongpl{#1}}{\glentryfirstplural{#1}}#2%
15694 }
15695 \newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
15696 \newcommand*{\@cGlspl}[2][{}]{%
15697   \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[{}]}%
15698 }
15699 \def\@cGlspl@#1#2[#3]{%
15700   \GlossariesWarning{\string\cGlspl\space is defaulting to
15701     \string\Glspl\space since you haven't enabled entry counting}%
15702   \@Glspl@{#1}{#2}[#3]%
15703 }
15704 \newcommand*{\cGlsplformat}[2]{%
15705   \ifglshaslong{#1}{\Glentrylongpl{#1}}{\Glentryfirstplural{#1}}#2%
15706 }
15707 \newcommand*{\loadglsentries}[2][\@gls@default]{%
15708   \let\@gls@default\glsdefaulttype
15709   \def\glsdefaulttype{#1}\input{#2}%
15710   \let\glsdefaulttype\@gls@default
15711 }
15712 \@onlypreamble{\loadglsentries}
15713 \newcommand*{\glstextformat}[1]{#1}
15714 \newcommand*{\glentryfmt}{%
15715   \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
15716 }
15717 \newcommand*{\@@gls@default@entryfmt}[2]{%
15718   \ifdefempty\glscustomtext
15719     {%
15720       \glsifplural
15721       {%
15722         \glscapscase
15723         {%
15724           \ifglused\glslabel
15725             {%
15726               #2{\glentryplural{\glslabel}}%
15727               {\glentrydescplural{\glslabel}}%
15728               {\glentrysymbolplural{\glslabel}}{\glsinsert}%
15729             }%
15730             {%

```

```

15731      #1{\glsentryfirstplural{\glslabel}}%
15732      {\glsentrydescplural{\glslabel}}%
15733      {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15734  }%
15735 }%
15736 {%
15737   \ifglused\glslabel
15738   {%
15739     \ifbool{glscompatible-3.07}%
15740     {%
15741       \protected@edef\@glo@etext{%
15742         #2{\glsentryplural{\glslabel}}%
15743         {\glsentrydescplural{\glslabel}}%
15744         {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15745       \xmakefirstuc\@glo@etext
15746     }%
15747     {%
15748       #2{\Glsentryplural{\glslabel}}%
15749       {\glsentrydescplural{\glslabel}}%
15750       {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15751     }%
15752   }%
15753   {%
15754     \ifbool{glscompatible-3.07}%
15755     {%
15756       \protected@edef\@glo@etext{%
15757         #1{\glsentryfirstplural{\glslabel}}%
15758         {\glsentrydescplural{\glslabel}}%
15759         {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15760       \xmakefirstuc\@glo@etext
15761     }%
15762     {%
15763       #1{\Glsentryfirstplural{\glslabel}}%
15764       {\glsentrydescplural{\glslabel}}%
15765       {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15766     }%
15767   }%
15768 }%
15769 {%
15770   \ifglused\glslabel
15771   {%
15772     \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
15773     {\glsentrydescplural{\glslabel}}%
15774     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15775   }%
15776   {%
15777     \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
15778     {\glsentrydescplural{\glslabel}}%
15779     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%

```

```

15780     }%
15781 }%
15782 }%
15783 {%
15784   \glscapscase
15785   {%
15786     \ifglused\glslabel
15787     {%
15788       #2{\glsentrytext{\glslabel}}%
15789       {\glsentrydesc{\glslabel}}%
15790       {\glsentrysymbol{\glslabel}}{\glsinsert}%
15791     }%
15792     {%
15793       #1{\glsentryfirst{\glslabel}}%
15794       {\glsentrydesc{\glslabel}}%
15795       {\glsentrysymbol{\glslabel}}{\glsinsert}%
15796     }%
15797   }%
15798   {%
15799     \ifglused\glslabel
15800     {%
15801       \ifbool{glscompatible-3.07}%
15802       {%
15803         \protected@edef\@glo@etext{%
15804           #2{\glsentrytext{\glslabel}}%
15805           {\glsentrydesc{\glslabel}}%
15806           {\glsentrysymbol{\glslabel}}{\glsinsert}}%
15807         \xmakefirstuc\@glo@etext
15808       }%
15809       {%
15810         #2{\Glsentrytext{\glslabel}}%
15811         {\glsentrydesc{\glslabel}}%
15812         {\glsentrysymbol{\glslabel}}{\glsinsert}%
15813       }%
15814     }%
15815     {%
15816       \ifbool{glscompatible-3.07}%
15817       {%
15818         \protected@edef\@glo@etext{%
15819           #1{\glsentryfirst{\glslabel}}%
15820           {\glsentrydesc{\glslabel}}%
15821           {\glsentrysymbol{\glslabel}}{\glsinsert}}%
15822         \xmakefirstuc\@glo@etext
15823       }%
15824       {%
15825         #1{\Glsentryfirst{\glslabel}}%
15826         {\glsentrydesc{\glslabel}}%
15827         {\glsentrysymbol{\glslabel}}{\glsinsert}%
15828       }%

```

```

15829     }%
15830 }%
15831 {%
15832     \ifglused\glslabel
15833     {%
15834         \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}}%
15835         {\glsentrydesc{\glslabel}}}%
15836         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
15837     }%
15838     {%
15839         \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}}%
15840         {\glsentrydesc{\glslabel}}}%
15841         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
15842     }%
15843 }%
15844 }%
15845 }%
15846 {%
15847     \ifglused{\glslabel}%
15848     {%
15849         #2{\glscustomtext}%
15850         {\glsentrydesc{\glslabel}}}%
15851         {\glsentrysymbol{\glslabel}}{}%
15852     }%
15853     {%
15854         #1{\glscustomtext}%
15855         {\glsentrydesc{\glslabel}}}%
15856         {\glsentrysymbol{\glslabel}}{}%
15857     }%
15858 }%
15859 }
15860 \newcommand*{\glsgenentryfmt}{%
15861     \ifdefempty\glscustomtext
15862     {%
15863         \glsifplural
15864         {%
15865             \glscapscase
15866             {%
15867                 \ifglused\glslabel
15868                 {%
15869                     \glsentryplural{\glslabel}\glsinsert
15870                 }%
15871                 {%
15872                     \glsentryfirstplural{\glslabel}\glsinsert
15873                 }%
15874             }%
15875         }%
15876         \ifglused\glslabel
15877         {%

```

```

15878         \Glsentryplural{\glslabel}\glsinsert
15879     }%
15880     {%
15881         \Glsentryfirstplural{\glslabel}\glsinsert
15882     }%
15883 }%
15884 {%
15885     \ifglused\glslabel
15886     {%
15887         \mfirstucMakeUppercase
15888         {\Glsentryplural{\glslabel}\glsinsert}%
15889     }%
15890     {%
15891         \mfirstucMakeUppercase
15892         {\Glsentryfirstplural{\glslabel}\glsinsert}%
15893     }%
15894 }%
15895 }%
15896 {%
15897     \glscapscase
15898     {%
15899         \ifglused\glslabel
15900         {%
15901             \Glsentrytext{\glslabel}\glsinsert
15902         }%
15903         {%
15904             \Glsentryfirst{\glslabel}\glsinsert
15905         }%
15906     }%
15907     {%
15908         \ifglused\glslabel
15909         {%
15910             \Glsentrytext{\glslabel}\glsinsert
15911         }%
15912         {%
15913             \Glsentryfirst{\glslabel}\glsinsert
15914         }%
15915     }%
15916     {%
15917         \ifglused\glslabel
15918         {%
15919             \mfirstucMakeUppercase{\Glsentrytext{\glslabel}\glsinsert}%
15920         }%
15921         {%
15922             \mfirstucMakeUppercase{\Glsentryfirst{\glslabel}\glsinsert}%
15923         }%
15924     }%
15925 }%
15926 }%

```

```

15927 {%
15928   \glscustomtext\glsinsert
15929 }%
15930 }
15931 \newcommand*{\glsgenacfmt}{%
15932   \ifdefempty\glscustomtext
15933   {%
15934     \ifglssused\glslabel
15935     {%
15936       \glsifplural
15937       {%
15938         \glscapscase
15939         {%
15940           \acronymfont{\glstryshortpl{\glslabel}}\glsinsert
15941         }%
15942         {%
15943           \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
15944         }%
15945         {%
15946           \mfirstucMakeUppercase
15947             {\acronymfont{\glstryshortpl{\glslabel}}\glsinsert}%
15948         }%
15949       }%
15950     }%
15951     \glscapscase
15952     {%
15953       \acronymfont{\glstryshort{\glslabel}}\glsinsert
15954     }%
15955     {%
15956       \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
15957     }%
15958     {%
15959       \mfirstucMakeUppercase
15960         {\acronymfont{\glstryshort{\glslabel}}\glsinsert}%
15961     }%
15962   }%
15963 }%
15964 {%
15965   \glsifplural
15966   {%
15967     \glscapscase
15968     {%
15969       \genplacrfullformat{\glslabel}{\glsinsert}%
15970     }%
15971     {%
15972       \Genplacrfullformat{\glslabel}{\glsinsert}%
15973     }%
15974     {%
15975       \mfirstucMakeUppercase

```



```

15976         {\genplacrfullformat{\glslabel}{\glsinsert}}}%
15977     }%
15978 }%
15979 {%
15980     \glscapscase
15981     {%
15982         \genacrfullformat{\glslabel}{\glsinsert}%
15983     }%
15984     {%
15985         \Genacrfullformat{\glslabel}{\glsinsert}%
15986     }%
15987     {%
15988         \mfirstucMakeUppercase
15989         {\genacrfullformat{\glslabel}{\glsinsert}}}%
15990     }%
15991 }%
15992 }%
15993 }%
15994 {%
15995     \glscustomtext
15996 }%
15997 }
15998 \newcommand*{\genacrfullformat}[2]{%
15999     \glsentrylong{#1}#2\space
16000     (\protect\firstacronymfont{\glsentryshort{#1}})%
16001 }
16002 \newcommand*{\Genacrfullformat}[2]{%
16003     \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
16004     \xmakefirstuc\gls@text
16005 }
16006 \newcommand*{\genplacrfullformat}[2]{%
16007     \glsentrylongpl{#1}#2\space
16008     (\protect\firstacronymfont{\glsentryshortpl{#1}})%
16009 }
16010 \newcommand*{\Genplacrfullformat}[2]{%
16011     \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
16012     \xmakefirstuc\gls@text
16013 }
16014 \newcommand*{\glsdisplayfirst}[4]{#1#4}
16015 \newcommand*{\glsdisplay}[4]{#1#4}
16016 \newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
16017     \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
16018     Use \string\defglsentryfmt\space instead}%
16019     \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
16020     \edef\@gls@doentrydef{%
16021         \noexpand\defglsentryfmt[#1]{%
16022             \noexpand\ifcsdef{gls@#1@displayfirst}%
16023             {%
16024                 \noexpand\@@gls@default@entryfmt

```

```

16025         {\noexpand\csuse{gls@#1@displayfirst}}%
16026         {\noexpand\csuse{gls@#1@display}}%
16027     }%
16028     {%
16029         \noexpand\@gls@default@entryfmt
16030         {\noexpand\glsdisplayfirst}%
16031         {\noexpand\csuse{gls@#1@display}}%
16032     }%
16033 }%
16034 }%
16035 \@gls@doentrydef
16036 }
16037 \newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
16038   \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
16039   Use \string\defglsentryfmt\space instead}%
16040   \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
16041   \edef\@gls@doentrydef{%
16042     \noexpand\defglsentryfmt[#1]{%
16043       \noexpand\ifcsdef{gls@#1@display}%
16044       {%
16045         \noexpand\@gls@default@entryfmt
16046         {\noexpand\csuse{gls@#1@displayfirst}}%
16047         {\noexpand\csuse{gls@#1@display}}%
16048       }%
16049       {%
16050         \noexpand\@gls@default@entryfmt
16051         {\noexpand\csuse{gls@#1@displayfirst}}%
16052         {\noexpand\glsdisplay}%
16053       }%
16054     }%
16055   }%
16056   \@gls@doentrydef
16057 }
16058 \define@key{glslink}{counter}{%
16059   \ifcsundef{c@#1}%
16060   {%
16061     \PackageError{glossaries}%
16062     {There is no counter called ‘#1’}%
16063     {%
16064       The counter key should have the name of a valid counter
16065       as its value%
16066     }%
16067   }%
16068   {%
16069     \def\@gls@counter{#1}%
16070   }%
16071 }
16072 \define@key{glslink}{format}{%
16073   \def\@glsnumberformat{#1}}

```

```

16074 \define@boolkey{glslink}{hyper}[true]{}
16075 \ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
16076 \define@boolkey{glslink}{local}[true]{}
16077 \newcommand*{\glslinkvar}[3]{#1}
16078 \newcommand*{\glsifhyper}[2]{%
16079   \glslinkvar{#1}{#2}{#1}%
16080   \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
16081     you mean \string\glsifhyperon\space or \string\glslinkvar?}%
16082 }
16083 \newcommand*{\@gls@hyp@opt}[1]{%
16084   \let\glslinkvar\@firstofthree
16085   \let\@gls@hyp@opt@cs#1\relax
16086   \@ifstar{\s@gls@hyp@opt}%
16087   {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
16088 }
16089 \newcommand*{\s@gls@hyp@opt}[1][]{%
16090   \let\glslinkvar\@secondofthree
16091   \@gls@hyp@opt@cs[hyper=false,#1]}
16092 \newcommand*{\p@gls@hyp@opt}[1][]{%
16093   \let\glslinkvar\@thirdofthree
16094   \@gls@hyp@opt@cs[hyper=true,#1]}
16095 \newrobustcmd*{\glslink}{%
16096   \@gls@hyp@opt\@gls@@link
16097 }
16098 \newcommand*{\@gls@@link}[3][]{%
16099   \glsdoifexistsordo{#2}%
16100   {%
16101     \let\do@gls@link@checkfirsthyper\relax
16102     \@gls@link[#1]{#2}{#3}%
16103   }{%
16104     \glstextformat{#3}%
16105   }%
16106   \glspostlinkhook
16107 }
16108 \newcommand*{\glspostlinkhook}{}
16109 \newcommand*{\@gls@link@checkfirsthyper}{%
16110   \ifglssused{\glslabel}%
16111   {%
16112   }%
16113   {%
16114     \gls@checkisacronymlist\glstype
16115     \ifglshyperfirst
16116       \if@glsisacronymlist
16117         \ifglssacrfootnote
16118           \KV@glslink@hyperfalse
16119           \fi
16120         \fi
16121       \else
16122         \KV@glslink@hyperfalse

```

```

16123 \fi
16124 }%
16125 \glslinkcheckfirsthyperhook
16126 }
16127 \newcommand*{\glslinkcheckfirsthyperhook}{}
16128 \newcommand*{\glslinkpostsetkeys}{}
16129 \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
16130 \newcommand*{\do@gl:disablehyperinlist}{}%
16131 \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
16132 {\KV@glslink@hyperfalse}{}%
16133 }
16134 \newcommand*{\@gls@setdefault@glslink@opts}{}
16135 \def\@gls@link[#1]#2#3{%
16136 \leavevmode
16137 \edef\glslabel{\glsdetoklabel{#2}}%
16138 \def\@gls@link@opts{#1}%
16139 \let\@gls@link@label\glslabel
16140 \def\@glsnumberformat{glsnumberformat}%
16141 \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
16142 \edef\glstype{\csname glo@\glslabel @type\endcsname}%
16143 \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
16144 \@gls@setdefault@glslink@opts
16145 \do@gl:disablehyperinlist
16146 \do@gls@link@checkfirsthyper
16147 \setkeys{glslink}{#1}%
16148 \glslinkpostsetkeys
16149 \@gls@saveentrycounter
16150 \@gls@setsort{\glslabel}%
16151 \@do@wrglossary{#2}%
16152 \ifKV@glslink@hyper
16153 \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
16154 \else
16155 \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
16156 \fi
16157 \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
16158 }
16159 \newcommand*{\glolinkprefix}{glo:}
16160 \def\glsentrycounter{\glscounter}%
16161 \newcommand*{\@gls@saveentrycounter}{}%
16162 \def\@gls@Hcounter{}%
16163 \ifthenelse{\equal{\@gls@counter}{equation}}%
16164 {
16165 \ifcsundef{xatlevel@}%
16166 {%
16167 \edef\theglsentrycounter{\expandafter\noexpand
16168 \csname the\@gls@counter\endcsname}%
16169 }%
16170 {%
16171 \ifx\xatlevel@\@empty

```

```

16172     \edef\theglsentrycounter{\expandafter\noexpand
16173     \csname the\@gls@counter\endcsname}%
16174   \else
16175     \savecounters@
16176     \advance\c@equation by 1\relax
16177     \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
16178     \ifcsundef{theH\@gls@counter}%
16179     {%
16180       \def\@gls@Hcounter{\theglsentrycounter}%
16181     }%
16182     {%
16183       \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
16184     }%
16185     \protected@edef\theHglentrycounter{\@gls@Hcounter}%
16186     \restorecounters@
16187   \fi
16188 }%
16189 }%
16190 {%
16191   \edef\theglsentrycounter{\expandafter\noexpand
16192   \csname the\@gls@counter\endcsname}%
16193 }%
16194 \ifx\@gls@Hcounter\@empty
16195   \ifcsundef{theH\@gls@counter}%
16196   {%
16197     \def\theHglentrycounter{\theglsentrycounter}%
16198   }%
16199   {%
16200     \protected@edef\theHglentrycounter{\expandafter\noexpand
16201     \csname theH\@gls@counter\endcsname}%
16202   }%
16203 \fi
16204 }
16205 \def\@set@glo@numformat#1#2#3#4{%
16206   \expandafter\@glo@check@mkidxrangechar#3\@nil
16207   \protected@edef#1{%
16208     \@glo@prefix setentrycounter[#4]{#2}%
16209     \expandafter\string\csname\@glo@suffix\endcsname
16210   }%
16211   \@gls@checkmkidxchars#1%
16212 }
16213 \def\@glo@check@mkidxrangechar#1#2\@nil{%
16214 \if#1(\relax
16215   \def\@glo@prefix{(%
16216   \if\relax#2\relax
16217     \def\@glo@suffix{glsnumberformat}%
16218   \else
16219     \def\@glo@suffix{#2}%
16220   \fi

```

```

16221 \else
16222   \if#1)\relax
16223     \def\@glo@prefix{}}}%
16224     \if\relax#2\relax
16225       \def\@glo@suffix{glsnumberformat}%
16226     \else
16227       \def\@glo@suffix{#2}%
16228   \fi
16229 \else
16230   \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
16231 \fi
16232 \fi}
16233 \newcommand*{\@gls@escbsdq}[1]{%
16234   \def\@gls@checkedmkidx{%
16235     \let\gls@xdystring=#1\relax
16236     \@onelevel@sanitize\gls@xdystring
16237     \edef\do@gls@xdycheckbackslash{%
16238       \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
16239       \@backslashchar\@backslashchar\noexpand\null}%
16240     \do@gls@xdycheckbackslash
16241     \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
16242     \def\@gls@checkedmkidx{%
16243       \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
16244       \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
16245       \@for\@gls@tmp:=\gls@protected@pagefmts\do
16246       {%
16247         \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
16248         \@onelevel@sanitize\@gls@sanitized@tmp
16249         \edef\gls@dostubst{%
16250           \noexpand\DTLsubstituteall\noexpand\gls@xdystring
16251           {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
16252         }%
16253         \gls@dostubst
16254       }%
16255       \let#1=\gls@xdystring
16256 }
16257 \newcommand{\@gls@checkmkidxchars}[1]{%
16258   \ifglsxindy
16259     \@gls@escbsdq{#1}%
16260   \else
16261     \def\@gls@checkedmkidx{%
16262       \expandafter\@gls@checkquote#1\@nil""\null
16263       \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16264       \def\@gls@checkedmkidx{%
16265         \expandafter\@gls@checkescquote#1\@nil\""\null
16266         \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16267         \def\@gls@checkedmkidx{%
16268           \expandafter\@gls@checkescactual#1\@nil\?\?\null
16269           \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%

```

```

16270 \def\@gls@checkedmkidx{%
16271 \expandafter\@gls@checkactual#1\@nil??\null
16272 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16273 \def\@gls@checkedmkidx{%
16274 \expandafter\@gls@checkbar#1\@nil||\null
16275 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16276 \def\@gls@checkedmkidx{%
16277 \expandafter\@gls@checkescbar#1\@nil\\|\null
16278 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16279 \def\@gls@checkedmkidx{%
16280 \expandafter\@gls@checklevel#1\@nil!!\null
16281 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16282 \fi
16283 }
16284 \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
16285 \newtoks\@gls@tmpb
16286 \def\@gls@checkquote#1"#2"#3\null{%
16287 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16288 \toks@={#1}%
16289 \ifx\null#2\null
16290 \ifx\null#3\null
16291 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16292 \def\@gls@checkquote{\relax}%
16293 \else
16294 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16295 \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
16296 \def\@gls@checkquote{\@gls@checkquote#3\null}%
16297 \fi
16298 \else
16299 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16300 \@gls@quotechar\@gls@quotechar}%
16301 \ifx\null#3\null
16302 \def\@gls@checkquote{\@gls@checkquote#2""\null}%
16303 \else
16304 \def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
16305 \fi
16306 \fi
16307 \@gls@checkquote
16308 }
16309 \def\@gls@checkescquote#1\ "#2\ "#3\null{%
16310 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16311 \toks@={#1}%
16312 \ifx\null#2\null
16313 \ifx\null#3\null
16314 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16315 \def\@gls@checkescquote{\relax}%
16316 \else
16317 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16318 \@gls@quotechar\string"\@gls@quotechar

```

```

16319     \@gls@quotechar\string\" \@gls@quotechar}%
16320     \def\@gls@checkescquote{\@gls@checkescquote#3\null}%
16321     \fi
16322   \else
16323     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16324       \@gls@quotechar\string\" \@gls@quotechar}%
16325     \ifx\null#3\null
16326       \def\@gls@checkescquote{\@gls@checkescquote#2\" \"\null}%
16327     \else
16328       \def\@gls@checkescquote{\@gls@checkescquote#2\"#3\null}%
16329     \fi
16330   \fi
16331 \@@gls@checkescquote
16332 }
16333 \def\@gls@checkescactual#1\?#2\?#3\null{%
16334   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16335   \toks@={#1}%
16336   \ifx\null#2\null
16337     \ifx\null#3\null
16338       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16339       \def\@gls@checkescactual{\relax}%
16340     \else
16341       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16342         \@gls@quotechar\string\" \@gls@actualchar
16343         \@gls@quotechar\string\" \@gls@actualchar}%
16344       \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
16345     \fi
16346   \else
16347     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16348       \@gls@quotechar\string\" \@gls@actualchar}%
16349     \ifx\null#3\null
16350       \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
16351     \else
16352       \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
16353     \fi
16354   \fi
16355 \@@gls@checkescactual
16356 }
16357 \def\@gls@checkescbar#1\|#2\|#3\null{%
16358   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16359   \toks@={#1}%
16360   \ifx\null#2\null
16361     \ifx\null#3\null
16362       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16363       \def\@gls@checkescbar{\relax}%
16364     \else
16365       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16366         \@gls@quotechar\string\" \@gls@encapchar
16367         \@gls@quotechar\string\" \@gls@encapchar}%

```



```

16368 \def\@gls@checkesbar{\@gls@checkesbar#3\null}%
16369 \fi
16370 \else
16371 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16372 \gls@quotechar\string"\@gls@encapchar}%
16373 \ifx\null#3\null
16374 \def\@gls@checkesbar{\@gls@checkesbar#2\|\|\null}%
16375 \else
16376 \def\@gls@checkesbar{\@gls@checkesbar#2\|#3\null}%
16377 \fi
16378 \fi
16379 \@gls@checkesbar
16380 }
16381 \def\@gls@checkesclevel#1\!#2\!#3\null{%
16382 \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16383 \toks@={#1}%
16384 \ifx\null#2\null
16385 \ifx\null#3\null
16386 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16387 \def\@gls@checkesclevel{\relax}%
16388 \else
16389 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16390 \gls@quotechar\string"\@gls@levelchar
16391 \gls@quotechar\string"\@gls@levelchar}%
16392 \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
16393 \fi
16394 \else
16395 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16396 \gls@quotechar\string"\@gls@levelchar}%
16397 \ifx\null#3\null
16398 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
16399 \else
16400 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
16401 \fi
16402 \fi
16403 \@gls@checkesclevel
16404 }
16405 \def\@gls@checkbar#1|#2|#3\null{%
16406 \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16407 \toks@={#1}%
16408 \ifx\null#2\null
16409 \ifx\null#3\null
16410 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16411 \def\@gls@checkbar{\relax}%
16412 \else
16413 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16414 \gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
16415 \def\@gls@checkbar{\@gls@checkbar#3\null}%
16416 \fi

```

```

16417 \else
16418 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16419 \@gls@quotechar\@gls@encapchar}%
16420 \ifx\null#3\null
16421 \def\@gls@checkbar{\@gls@checkbar#2||\null}%
16422 \else
16423 \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
16424 \fi
16425 \fi
16426 \@gls@checkbar
16427 }
16428 \def\@gls@checklevel#1!#2!#3\null{%
16429 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16430 \toks@={#1}%
16431 \ifx\null#2\null
16432 \ifx\null#3\null
16433 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16434 \def\@gls@checklevel{\relax}%
16435 \else
16436 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16437 \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
16438 \def\@gls@checklevel{\@gls@checklevel#3\null}%
16439 \fi
16440 \else
16441 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16442 \@gls@quotechar\@gls@levelchar}%
16443 \ifx\null#3\null
16444 \def\@gls@checklevel{\@gls@checklevel#2!!\null}%
16445 \else
16446 \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
16447 \fi
16448 \fi
16449 \@gls@checklevel
16450 }
16451 \def\@gls@checkactual#1?#2?#3\null{%
16452 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16453 \toks@={#1}%
16454 \ifx\null#2\null
16455 \ifx\null#3\null
16456 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16457 \def\@gls@checkactual{\relax}%
16458 \else
16459 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16460 \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
16461 \def\@gls@checkactual{\@gls@checkactual#3\null}%
16462 \fi
16463 \else
16464 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16465 \@gls@quotechar\@gls@actualchar}%

```

```

16466 \ifx\null#3\null
16467 \def\@gls@checkactual{\@gls@checkactual#2??\null}%
16468 \else
16469 \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
16470 \fi
16471 \fi
16472 \@gls@checkactual
16473 }
16474 \def\@gls@xdycheckquote#1"#2"#3\null{%
16475 \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16476 \toks@={#1}%
16477 \ifx\null#2\null
16478 \ifx\null#3\null
16479 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16480 \def\@gls@xdycheckquote{\relax}%
16481 \else
16482 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16483 \string"\string"}%
16484 \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
16485 \fi
16486 \else
16487 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16488 \string"}%
16489 \ifx\null#3\null
16490 \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
16491 \else
16492 \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
16493 \fi
16494 \fi
16495 \@gls@xdycheckquote
16496 }
16497 \edef\def\@gls@xdycheckbackslash{%
16498 \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
16499 ##2\@backslashchar##3\noexpand\null{%
16500 \noexpand\@gls@tmpb=\noexpand\expandafter
16501 {\noexpand\@gls@checkedmkidx}%
16502 \noexpand\toks@={##1}%
16503 \noexpand\ifx\noexpand\null##2\noexpand\null
16504 \noexpand\ifx\noexpand\null##3\noexpand\null
16505 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16506 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
16507 \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
16508 \noexpand\else
16509 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16510 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
16511 \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
16512 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16513 \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
16514 \noexpand\fi

```

```

16515 \noexpand\else
16516 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16517 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
16518 \@backslashchar\@backslashchar}%
16519 \noexpand\ifx\noexpand\null##3\noexpand\null
16520 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16521 \noexpand\@gls@xdycheckbackslash##2\@backslashchar
16522 \@backslashchar\noexpand\null}%
16523 \noexpand\else
16524 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16525 \noexpand\@gls@xdycheckbackslash##2\@backslashchar
16526 ##3\noexpand\null}%
16527 \noexpand\fi
16528 \noexpand\fi
16529 \noexpand\@gls@xdycheckbackslash
16530 }%
16531 }
16532 \def@gls@xdycheckbackslash
16533
16534 \newlength\gls@tmplen
16535 \newcommand*\@glsdohypertarget}[2]{%
16536 \@glsshowtarget{#1}%
16537 \settoheight{\gls@tmplen}{#2}%
16538 \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
16539 }
16540 \newcommand*\@glsdohyperlink}[2]{%
16541 \@glsshowtarget{#1}%
16542 \hyperlink{#1}{#2}%
16543 }
16544 \newcommand*\@glsdonohyperlink}[2]{#2}
16545 \ifcsundef{hyperlink}%
16546 {%
16547 \let\@glslink\glsdonohyperlink
16548 }%
16549 {%
16550 \let\@glslink\glsdohyperlink
16551 }
16552 \ifcsundef{hypertarget}%
16553 {%
16554 \let\@gls@target\@secondoftwo
16555 }%
16556 {%
16557 \let\@gls@target\glsdohypertarget
16558 }
16559 \newcommand{\glsdisablehyper}{%
16560 \KV@glslink@hyperfalse
16561 \let\@glslink\glsdonohyperlink
16562 \let\@gls@target\@secondoftwo
16563 }

```

```

16564 \newcommand{\glsenablehyper}{%
16565   \KV@glslink@hypertrue
16566   \let\@glslink\glsdohyperlink
16567   \let\@glstarget\glsdohypertarget
16568 }
16569 \providecommand{\@firstofthree}[3]{#1}
16570 \providecommand{\@secondofthree}[3]{#2}
16571 \newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
16572 \newcommand*{\@gls}[2][{}]{%
16573   \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
16574 }
16575 \def\@gls@#1#2[#3]{%
16576   \glstoifexists{#2}%
16577   {%
16578     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16579     \let\glsifplural\@secondoftwo
16580     \let\glscapscase\@firstofthree
16581     \let\glscustomtext\@empty
16582     \def\glsinsert{#3}%
16583     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16584     \@gls@link[#1]{#2}{\@glo@text}%
16585     \ifKV@glslink@local
16586       \glslocalunset{#2}%
16587     \else
16588       \glsunset{#2}%
16589     \fi
16590   }%
16591   \glspostlinkhook
16592 }
16593 \newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
16594 \newcommand*{\@Gls}[2][{}]{%
16595   \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
16596 }
16597 \def\@Gls@#1#2[#3]{%
16598   \glstoifexists{#2}%
16599   {%
16600     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16601     \let\glsifplural\@secondoftwo
16602     \let\glscapscase\@secondofthree
16603     \let\glscustomtext\@empty
16604     \def\glsinsert{#3}%
16605     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16606     \@gls@link[#1]{#2}{\@glo@text}%
16607     \ifKV@glslink@local
16608       \glslocalunset{#2}%
16609     \else
16610       \glsunset{#2}%
16611     \fi
16612   }%

```

```

16613 \glspostlinkhook
16614 }
16615 \newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
16616 \newcommand*{\@GLS}[2] [] {%
16617   \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}]%
16618 }
16619 \def\@GLS@#1#2[#3] {%
16620   \glsdoifexists{#2}%
16621   {%
16622     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16623     \let\glsifplural\@secondoftwo
16624     \let\glscapscase\@thirdofthree
16625     \let\glscustomtext\@empty
16626     \def\glsinsert{#3}%
16627     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16628     \@gls@link[#1]{#2}{\@glo@text}%
16629     \ifKV@glslink@local
16630       \glslocalunset{#2}%
16631     \else
16632       \glsunset{#2}%
16633     \fi
16634   }%
16635   \glspostlinkhook
16636 }
16637 \newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
16638 \newcommand*{\@glspl}[2] [] {%
16639   \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}]%
16640 }
16641 \def\@glspl@#1#2[#3] {%
16642   \glsdoifexists{#2}%
16643   {%
16644     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16645     \let\glsifplural\@firstoftwo
16646     \let\glscapscase\@firstofthree
16647     \let\glscustomtext\@empty
16648     \def\glsinsert{#3}%
16649     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16650     \@gls@link[#1]{#2}{\@glo@text}%
16651     \ifKV@glslink@local
16652       \glslocalunset{#2}%
16653     \else
16654       \glsunset{#2}%
16655     \fi
16656   }%
16657   \glspostlinkhook
16658 }
16659 \newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
16660 \newcommand*{\@Glspl}[2] [] {%
16661   \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}]%

```

```

16662 }
16663 \def\@GLSpl@#1#2[#3]{%
16664   \glsdoifexists{#2}%
16665   {%
16666     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16667     \let\glsifplural\@firstoftwo
16668     \let\glscapscase\@secondofthree
16669     \let\glscustomtext\@empty
16670     \def\glsinsert{#3}%
16671     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16672     \@gl@link[#1]{#2}{\@glo@text}%
16673     \ifKV@gl@link@local
16674       \glslocalunset{#2}%
16675     \else
16676       \glsunset{#2}%
16677     \fi
16678   }%
16679   \glspostlinkhook
16680 }
16681 \newrobustcmd*{\GLSpl}{\@gl@hyp@opt\@GLSpl}
16682 \newcommand*{\@GLSpl}[2][{}]{%
16683   \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
16684 }
16685 \def\@GLSpl@#1#2[#3]{%
16686   \glsdoifexists{#2}%
16687   {%
16688     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16689     \let\glsifplural\@firstoftwo
16690     \let\glscapscase\@thirdofthree
16691     \let\glscustomtext\@empty
16692     \def\glsinsert{#3}%
16693     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16694     \@gl@link[#1]{#2}{\@glo@text}%
16695     \ifKV@gl@link@local
16696       \glslocalunset{#2}%
16697     \else
16698       \glsunset{#2}%
16699     \fi
16700   }%
16701   \glspostlinkhook
16702 }
16703 \newrobustcmd*{\glsdisp}{\@gl@hyp@opt\@glsdisp}
16704 \newcommand*{\@glsdisp}[3][{}]{%
16705   \glsdoifexists{#2}{%
16706     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16707     \let\glsifplural\@secondoftwo
16708     \let\glscapscase\@firstofthree
16709     \def\glscustomtext{#3}%
16710     \def\glsinsert{}%

```

```

16711 \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16712 \@gls@link[#1]{#2}{\@glo@text}%
16713 \ifKV@gls@link@local
16714 \glslocalunset{#2}%
16715 \else
16716 \glsunset{#2}%
16717 \fi
16718 }%
16719 \gls@postlinkhook
16720 }
16721 \newcommand*{\@gls@link@nocheckfirsthyper}{%
16722 \newcommand{\@gls@field@link}[3]{%
16723 \glsdoifexists{#2}%
16724 {%
16725 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16726 \@gls@link[#1]{#2}{#3}%
16727 }%
16728 \gls@postlinkhook
16729 }
16730 \newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
16731 \newcommand*{\@glstext}[2][]{%
16732 \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
16733 \def\@glstext@#1#2[#3]{%
16734 \@gls@field@link{#1}{#2}{\glstentrytext{#2}#3}%
16735 }
16736 \newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
16737 \newcommand*{\@GLStext}[2][]{%
16738 \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
16739 \def\@GLStext@#1#2[#3]{%
16740 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrytext{#2}#3}%
16741 }
16742 \newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
16743 \newcommand*{\@Glstext}[2][]{%
16744 \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
16745 \def\@Glstext@#1#2[#3]{%
16746 \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
16747 }
16748 \newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
16749 \newcommand*{\@glsfirst}[2][]{%
16750 \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
16751 \def\@glsfirst@#1#2[#3]{%
16752 \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
16753 }
16754 \newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
16755 \newcommand*{\@Glsfirst}[2][]{%
16756 \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}}
16757 \def\@Glsfirst@#1#2[#3]{%
16758 \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
16759 }

```



```

16760 \newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
16761 \newcommand*{\@GLSfirst}[2] [] {%
16762   \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
16763 \def\@GLSfirst@#1#2[#3] {%
16764   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}{#3}}}%
16765 }
16766 \newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
16767 \newcommand*{\@glsplural}[2] [] {%
16768   \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
16769 \def\@glsplural@#1#2[#3] {%
16770   \@gls@field@link{#1}{#2}{\glsentryplural{#2}{#3}}%
16771 }
16772 \newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
16773 \newcommand*{\@Glsplural}[2] [] {%
16774   \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
16775 \def\@Glsplural@#1#2[#3] {%
16776   \@gls@field@link{#1}{#2}{\Glsentryplural{#2}{#3}}%
16777 }
16778 \newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
16779 \newcommand*{\@GLSplural}[2] [] {%
16780   \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
16781 \def\@GLSplural@#1#2[#3] {%
16782   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}{#3}}}%
16783 }
16784 \newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
16785 \newcommand*{\@glsfirstplural}[2] [] {%
16786   \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
16787 \def\@glsfirstplural@#1#2[#3] {%
16788   \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}{#3}}%
16789 }
16790 \newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
16791 \newcommand*{\@Glsfirstplural}[2] [] {%
16792   \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
16793 \def\@Glsfirstplural@#1#2[#3] {%
16794   \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}{#3}}%
16795 }
16796 \newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
16797 \newcommand*{\@GLSfirstplural}[2] [] {%
16798   \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
16799 \def\@GLSfirstplural@#1#2[#3] {%
16800   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}{#3}}}%
16801 }
16802 \newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
16803 \newcommand*{\@glsname}[2] [] {%
16804   \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
16805 \def\@glsname@#1#2[#3] {%
16806   \@gls@field@link{#1}{#2}{\glsentryname{#2}{#3}}%
16807 }
16808 \newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}

```

```

16809 \newcommand*{\@GLsname}[2] [] {%
16810   \new@ifnextchar[{\@GLsname@{#1}{#2}}{\@GLsname@{#1}{#2} []}]
16811 \def\@GLsname@#1#2[#3] {%
16812   \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
16813 }
16814 \newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
16815 \newcommand*{\@GLSname}[2] [] {%
16816   \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
16817 \def\@GLSname@#1#2[#3] {%
16818   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryname{#2}#3}}%
16819 }
16820 \newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
16821 \newcommand*{\@glsdesc}[2] [] {%
16822   \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
16823 \def\@glsdesc@#1#2[#3] {%
16824   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
16825 }
16826 \newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
16827 \newcommand*{\@GLSdesc}[2] [] {%
16828   \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
16829 \def\@GLSdesc@#1#2[#3] {%
16830   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
16831 }
16832 \newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
16833 \newcommand*{\@GLSdesc}[2] [] {%
16834   \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
16835 \def\@GLSdesc@#1#2[#3] {%
16836   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydesc{#2}#3}}%
16837 }
16838 \newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
16839 \newcommand*{\@glsdescplural}[2] [] {%
16840   \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
16841 \def\@glsdescplural@#1#2[#3] {%
16842   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
16843 }
16844 \newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
16845 \newcommand*{\@GLSdescplural}[2] [] {%
16846   \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
16847 \def\@GLSdescplural@#1#2[#3] {%
16848   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
16849 }
16850 \newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
16851 \newcommand*{\@GLSdescplural}[2] [] {%
16852   \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
16853 \def\@GLSdescplural@#1#2[#3] {%
16854   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydescplural{#2}#3}}%
16855 }
16856 \newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
16857 \newcommand*{\@glssymbol}[2] [] {%

```

```

16858 \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
16859 \def\@glssymbol@#1#2[#3]{%
16860 \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
16861 }
16862 \newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
16863 \newcommand*{\@Glsymbol}[2] [] {%
16864 \new@ifnextchar[{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
16865 \def\@Glsymbol@#1#2[#3]{%
16866 \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
16867 }
16868 \newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
16869 \newcommand*{\@GLSsymbol}[2] [] {%
16870 \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
16871 \def\@GLSsymbol@#1#2[#3]{%
16872 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrysymbol{#2}#3}}%
16873 }
16874 \newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
16875 \newcommand*{\@glssymbolplural}[2] [] {%
16876 \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}]
16877 \def\@glssymbolplural@#1#2[#3]{%
16878 \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
16879 }
16880 \newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
16881 \newcommand*{\@Glsymbolplural}[2] [] {%
16882 \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
16883 \def\@Glsymbolplural@#1#2[#3]{%
16884 \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
16885 }
16886 \newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
16887 \newcommand*{\@GLSsymbolplural}[2] [] {%
16888 \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
16889 \def\@GLSsymbolplural@#1#2[#3]{%
16890 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrysymbolplural{#2}#3}}%
16891 }
16892 \newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
16893 \newcommand*{\@glsuseri}[2] [] {%
16894 \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
16895 \def\@glsuseri@#1#2[#3]{%
16896 \@gls@field@link{#1}{#2}{\glstentryuseri{#2}#3}%
16897 }
16898 \newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
16899 \newcommand*{\@Glsuseri}[2] [] {%
16900 \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
16901 \def\@Glsuseri@#1#2[#3]{%
16902 \@gls@field@link{#1}{#2}{\glstentryuseri{#2}#3}%
16903 }
16904 \newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
16905 \newcommand*{\@GLSuseri}[2] [] {%
16906 \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]

```

```

16907 \def\@GLSuseri@#1#2[#3]{%
16908   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}%
16909 }
16910 \newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
16911 \newcommand*{\@glsuserii}[2][{}]{%
16912   \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
16913 \def\@glsuserii@#1#2[#3]{%
16914   \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
16915 }
16916 \newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
16917 \newcommand*{\@Glsuserii}[2][{}]{%
16918   \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
16919 \def\@Glsuserii@#1#2[#3]{%
16920   \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
16921 }
16922 \newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
16923 \newcommand*{\@GLSuserii}[2][{}]{%
16924   \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}}
16925 \def\@GLSuserii@#1#2[#3]{%
16926   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}%
16927 }
16928 \newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
16929 \newcommand*{\@glsuseriii}[2][{}]{%
16930   \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}}
16931 \def\@glsuseriii@#1#2[#3]{%
16932   \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
16933 }
16934 \newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
16935 \newcommand*{\@Glsuseriii}[2][{}]{%
16936   \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
16937 \def\@Glsuseriii@#1#2[#3]{%
16938   \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
16939 }
16940 \newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
16941 \newcommand*{\@GLSuseriii}[2][{}]{%
16942   \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}}
16943 \def\@GLSuseriii@#1#2[#3]{%
16944   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriii{#2}#3}}%
16945 }
16946 \newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
16947 \newcommand*{\@glsuseriv}[2][{}]{%
16948   \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2}[]}}
16949 \def\@glsuseriv@#1#2[#3]{%
16950   \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
16951 }
16952 \newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
16953 \newcommand*{\@Glsuseriv}[2][{}]{%
16954   \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}}
16955 \def\@Glsuseriv@#1#2[#3]{%

```

```

16956 \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
16957 }
16958 \newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
16959 \newcommand*{\@GLSuseriv}[2] [] {%
16960 \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}}
16961 \def\@GLSuseriv@#1#2[#3]{%
16962 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuseriv{#2}#3}}%
16963 }
16964 \newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
16965 \newcommand*{\@glsuserv}[2] [] {%
16966 \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}}
16967 \def\@glsuserv@#1#2[#3]{%
16968 \@gls@field@link{#1}{#2}{\glentryuserv{#2}#3}%
16969 }
16970 \newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
16971 \newcommand*{\@GLSuserv}[2] [] {%
16972 \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}}
16973 \def\@GLSuserv@#1#2[#3]{%
16974 \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
16975 }
16976 \newrobustcmd*{\GLSserv}{\@gls@hyp@opt\@GLSserv}
16977 \newcommand*{\@GLSserv}[2] [] {%
16978 \new@ifnextchar[{\@GLSserv@{#1}{#2}}{\@GLSserv@{#1}{#2} []}}
16979 \def\@GLSserv@#1#2[#3]{%
16980 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuserv{#2}#3}}%
16981 }
16982 \newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
16983 \newcommand*{\@glsuservi}[2] [] {%
16984 \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}}
16985 \def\@glsuservi@#1#2[#3]{%
16986 \@gls@field@link{#1}{#2}{\glentryuservi{#2}#3}%
16987 }
16988 \newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
16989 \newcommand*{\@GLSuservi}[2] [] {%
16990 \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}}
16991 \def\@GLSuservi@#1#2[#3]{%
16992 \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
16993 }
16994 \newrobustcmd*{\GLSservi}{\@gls@hyp@opt\@GLSservi}
16995 \newcommand*{\@GLSservi}[2] [] {%
16996 \new@ifnextchar[{\@GLSservi@{#1}{#2}}{\@GLSservi@{#1}{#2} []}}
16997 \def\@GLSservi@#1#2[#3]{%
16998 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuservi{#2}#3}}%
16999 }
17000 \newrobustcmd*{\acrshort}{\@gls@hyp@opt\@nsacrshort}
17001 \newcommand*{\@nsacrshort}[2] [] {%
17002 \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}}%
17003 }
17004 \def\@acrshort#1#2[#3]{%

```

```

17005 \glsdoifexists{#2}%
17006 {%
17007   \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17008   \let\glsifplural\@secondoftwo
17009   \let\glscapscase\@firstofthree
17010   \let\glsinsert\@empty
17011   \def\glscustomtext{%
17012     \acronymfont{\glsentryshort{#2}}#3%
17013   }%
17014   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17015 }%
17016 \glspostlinkhook
17017 }
17018 \newrobustcmd*{\Acrshort}{\@gl@hyp@opt\ns@Acrshort}
17019 \newcommand*{\ns@Acrshort}[2][]{%
17020   \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2}[]}%
17021 }
17022 \def\@Acrshort#1#2[#3]{%
17023   \glsdoifexists{#2}%
17024   {%
17025     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17026     \def\glslabel{#2}%
17027     \let\glsifplural\@secondoftwo
17028     \let\glscapscase\@secondofthree
17029     \let\glsinsert\@empty
17030     \def\glscustomtext{%
17031       \acronymfont{\Glsentryshort{#2}}#3%
17032     }%
17033     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17034   }%
17035   \glspostlinkhook
17036 }
17037 \newrobustcmd*{\ACRshort}{\@gl@hyp@opt\ns@ACRshort}
17038 \newcommand*{\ns@ACRshort}[2][]{%
17039   \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2}[]}%
17040 }
17041 \def\@ACRshort#1#2[#3]{%
17042   \glsdoifexists{#2}%
17043   {%
17044     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17045     \def\glslabel{#2}%
17046     \let\glsifplural\@secondoftwo
17047     \let\glscapscase\@thirdofthree
17048     \let\glsinsert\@empty
17049     \def\glscustomtext{%
17050       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
17051     }%
17052     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17053   }%

```

```

17054 \glspostlinkhook
17055 }
17056 \newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
17057 \newcommand*{\ns@acrshortpl}[2] [] {%
17058   \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
17059 }
17060 \def\@acrshortpl#1#2[#3]{%
17061   \glsdoifexists{#2}%
17062   {%
17063     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17064     \def\glslabel{#2}%
17065     \let\glsifplural\@firstoftwo
17066     \let\glsapscase\@firstofthree
17067     \let\glsinsert\@empty
17068     \def\glscustomtext{%
17069       \acronymfont{\glsentryshortpl{#2}}#3%
17070     }%
17071     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17072   }%
17073   \glspostlinkhook
17074 }
17075 \newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
17076 \newcommand*{\ns@Acrshortpl}[2] [] {%
17077   \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
17078 }
17079 \def\@Acrshortpl#1#2[#3]{%
17080   \glsdoifexists{#2}%
17081   {%
17082     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17083     \def\glslabel{#2}%
17084     \let\glsifplural\@firstoftwo
17085     \let\glsapscase\@secondofthree
17086     \let\glsinsert\@empty
17087     \def\glscustomtext{%
17088       \acronymfont{\Glsentryshortpl{#2}}#3%
17089     }%
17090     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17091   }%
17092   \glspostlinkhook
17093 }
17094 \newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
17095 \newcommand*{\ns@ACRshortpl}[2] [] {%
17096   \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
17097 }
17098 \def\@ACRshortpl#1#2[#3]{%
17099   \glsdoifexists{#2}%
17100   {%
17101     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17102     \def\glslabel{#2}%

```

```

17103 \let\glsifplural\@firstoftwo
17104 \let\glschapscase\@thirdofthree
17105 \let\glsinsert\@empty
17106 \def\glscustomtext{%
17107     \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
17108 }%
17109 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17110 }%
17111 \glspostlinkhook
17112 }
17113 \newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}
17114 \newcommand*{\ns@acrlong}[2][{}]{%
17115     \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
17116 }
17117 \def\@acrlong#1#2[#3]{%
17118     \glsdoifexists{#2}%
17119     {%
17120         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17121         \def\glslabel{#2}%
17122         \let\glsifplural\@secondoftwo
17123         \let\glschapscase\@firstofthree
17124         \let\glsinsert\@empty
17125         \def\glscustomtext{%
17126             \glsentrylong{#2}#3%
17127         }%
17128         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17129     }%
17130     \glspostlinkhook
17131 }
17132 \newrobustcmd*{\Acrlong}{\@gls@hyp@opt\@ns@Acrlong}
17133 \newcommand*{\ns@Acrlong}[2][{}]{%
17134     \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
17135 }
17136 \def\@Acrlong#1#2[#3]{%
17137     \glsdoifexists{#2}%
17138     {%
17139         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17140         \def\glslabel{#2}%
17141         \let\glsifplural\@secondoftwo
17142         \let\glschapscase\@secondofthree
17143         \let\glsinsert\@empty
17144         \def\glscustomtext{%
17145             \Glsentrylong{#2}#3%
17146         }%
17147         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17148     }%
17149     \glspostlinkhook
17150 }
17151 \newrobustcmd*{\ACRlong}{\@gls@hyp@opt\@ns@ACRlong}

```



```

17152 \newcommand*{\ns@ACRlong}[2][{}]{%
17153   \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
17154 }
17155 \def\@ACRlong#1#2[#3]{%
17156   \glsdoifexists{#2}%
17157   {%
17158     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17159     \def\glslabel{#2}%
17160     \let\glsifplural\@secondoftwo
17161     \let\glsapscase\@thirdofthree
17162     \let\glsinsert\@empty
17163     \def\glscustomtext{%
17164       \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
17165     }%
17166     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17167   }%
17168   \glspostlinkhook
17169 }
17170 \newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
17171 \newcommand*{\ns@acrlongpl}[2][{}]{%
17172   \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
17173 }
17174 \def\@acrlongpl#1#2[#3]{%
17175   \glsdoifexists{#2}%
17176   {%
17177     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17178     \def\glslabel{#2}%
17179     \let\glsifplural\@firstoftwo
17180     \let\glsapscase\@firstofthree
17181     \let\glsinsert\@empty
17182     \def\glscustomtext{%
17183       \glsentrylongpl{#2}#3%
17184     }%
17185     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17186   }%
17187   \glspostlinkhook
17188 }
17189 \newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
17190 \newcommand*{\ns@Acrlongpl}[2][{}]{%
17191   \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
17192 }
17193 \def\@Acrlongpl#1#2[#3]{%
17194   \glsdoifexists{#2}%
17195   {%
17196     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17197     \def\glslabel{#2}%
17198     \let\glsifplural\@firstoftwo
17199     \let\glsapscase\@secondofthree
17200     \let\glsinsert\@empty

```

```

17201 \def\glscustomtext{%
17202 \Glsentrylongpl{#2}#3%
17203 }%
17204 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17205 }%
17206 \glspostlinkhook
17207 }
17208 \newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
17209 \newcommand*{\ns@ACRlongpl}[2][{}]{%
17210 \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
17211 }
17212 \def\@ACRlongpl#1#2[#3]{%
17213 \glsdoifexists{#2}%
17214 {%
17215 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17216 \def\glslabel{#2}%
17217 \let\glsifplural\@firstoftwo
17218 \let\gls caps case\@thirdofthree
17219 \let\glsinsert\@empty
17220 \def\glscustomtext{%
17221 \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
17222 }%
17223 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17224 }%
17225 \glspostlinkhook
17226 }
17227 \newcommand*{\@gls@entry@field}[2]{%
17228 \csname glo@\glsdetoklabel{#1}@#2\endcsname
17229 }
17230 \newcommand*{\glsletentryfield}[3]{%
17231 \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
17232 }
17233 \newcommand*{\@Gls@entry@field}[2]{%
17234 \glsdoifexistsordo{#1}%
17235 {%
17236 \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
17237 \ifdef\@glo@text
17238 {%
17239 \xmakefirstuc{\@glo@text}%
17240 }%
17241 {%
17242 ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
17243 entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
17244 label and the field name}%
17245 }%
17246 }%
17247 {%
17248 ???%
17249 }%

```

```

17250 }
17251 \newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
17252 \newrobustcmd*{\Glsentryname}[1]{%
17253   \@Gls@entryname{#1}%
17254 }
17255 \newcommand*{\@Gls@entryname}[1]{%
17256   \@Gls@entry@field{#1}{name}%
17257 }
17258 \newcommand*{\@Gls@acentryname}[1]{%
17259   \ifglshaslong{#1}%
17260   {%
17261     \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
17262     \expandafter\@gls@getbody\@glo@text{}\@nil
17263     \expandafter\ifx\@gls@body\glsentrylong\relax
17264       \expandafter\Glsentrylong\@gls@rest
17265     \else
17266       \expandafter\ifx\@gls@body\glsentryshort\relax
17267         \expandafter\Glsentryshort\@gls@rest
17268       \else
17269         \expandafter\ifx\@gls@body\acronymfont\relax
17270         {%
17271           \let\glsentryshort\Glsentryshort
17272           \@glo@text
17273         }%
17274       \else
17275         \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
17276         {%
17277           \let\glsentryshort\Glsentryshort
17278           \@glo@text
17279         }%
17280       \else
17281         \xmakefirstuc{\@glo@text}%
17282       \fi
17283     \fi
17284   \fi
17285 }%
17286 }%
17287 {%
17288   \@Gls@entry@field{#1}{name}%
17289 }%
17290 }
17291 \newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
17292 \newrobustcmd*{\Glsentrydesc}[1]{%
17293   \@Gls@entry@field{#1}{desc}%
17294 }
17295 \newcommand*{\glsentrydescplural}[1]{%
17296   \@gls@entry@field{#1}{descplural}%
17297 }
17298 \newrobustcmd*{\Glsentrydescplural}[1]{%

```

```

17299 \@Gls@entry@field{#1}{descplural}%
17300 }
17301 \newcommand*{\glentrytext}[1]{\@Gls@entry@field{#1}{text}}
17302 \newrobustcmd*{\Glentrytext}[1]{%
17303 \@Gls@entry@field{#1}{text}%
17304 }
17305 \newcommand*{\glentryplural}[1]{%
17306 \@Gls@entry@field{#1}{plural}%
17307 }
17308 \newrobustcmd*{\Glentryplural}[1]{%
17309 \@Gls@entry@field{#1}{plural}%
17310 }
17311 \newcommand*{\glentrysymbol}[1]{%
17312 \@Gls@entry@field{#1}{symbol}%
17313 }
17314 \newrobustcmd*{\Glentrysymbol}[1]{%
17315 \@Gls@entry@field{#1}{symbol}%
17316 }
17317 \newcommand*{\glentrysymbolplural}[1]{%
17318 \@Gls@entry@field{#1}{symbolplural}%
17319 }
17320 \newrobustcmd*{\Glentrysymbolplural}[1]{%
17321 \@Gls@entry@field{#1}{symbolplural}%
17322 }
17323 \newcommand*{\glentryfirst}[1]{%
17324 \@Gls@entry@field{#1}{first}%
17325 }
17326 \newrobustcmd*{\Glentryfirst}[1]{%
17327 \@Gls@entry@field{#1}{first}%
17328 }
17329 \newcommand*{\glentryfirstplural}[1]{%
17330 \@Gls@entry@field{#1}{firstpl}%
17331 }
17332 \newrobustcmd*{\Glentryfirstplural}[1]{%
17333 \@Gls@entry@field{#1}{firstpl}%
17334 }
17335 \newrobustcmd*{\@glentrytitlecase}[2]{%
17336 \glstoifexists{#1}%
17337 {%
17338 \glstfieldfetch{#1}{#2}{\@Gls@value}%
17339 \xcapitalisewords{\@Gls@value}%
17340 }%
17341 }
17342 \ifdef\texorpdfstring
17343 {
17344 \newcommand*{\glentrytitlecase}[2]{%
17345 \texorpdfstring
17346 {\@glentrytitlecase{#1}{#2}}%
17347 {\@Gls@entry@field{#1}{#2}}%

```

```

17348 }
17349 }
17350 {
17351   \newcommand*{\glentrytitlecase}[2]{\@glentrytitlecase{#1}{#2}}
17352 }
17353 \newcommand*{\glentrytype}[1]{\@gls@entry@field{#1}{type}}
17354 \newcommand*{\glentrysort}[1]{%
17355   \@gls@entry@field{#1}{sort}}%
17356 }
17357 \newcommand*{\glentryparent}[1]{%
17358   \@gls@entry@field{#1}{parent}}%
17359 }
17360 \newcommand*{\glentryuseri}[1]{%
17361   \@gls@entry@field{#1}{useri}}%
17362 }
17363 \newrobustcmd*{\Glsentryuseri}[1]{%
17364   \@Gls@entry@field{#1}{useri}}%
17365 }
17366 \newcommand*{\glentryuserii}[1]{%
17367   \@gls@entry@field{#1}{userii}}%
17368 }
17369 \newrobustcmd*{\Glsentryuserii}[1]{%
17370   \@Gls@entry@field{#1}{userii}}%
17371 }
17372 \newcommand*{\glentryuseriii}[1]{%
17373   \@gls@entry@field{#1}{useriii}}%
17374 }
17375 \newrobustcmd*{\Glsentryuseriii}[1]{%
17376   \@Gls@entry@field{#1}{useriii}}%
17377 }
17378 \newcommand*{\glentryuseriv}[1]{%
17379   \@gls@entry@field{#1}{useriv}}%
17380 }
17381 \newrobustcmd*{\Glsentryuseriv}[1]{%
17382   \@Gls@entry@field{#1}{useriv}}%
17383 }
17384 \newcommand*{\glentryuserv}[1]{%
17385   \@gls@entry@field{#1}{userv}}%
17386 }
17387 \newrobustcmd*{\Glsentryuserv}[1]{%
17388   \@Gls@entry@field{#1}{userv}}%
17389 }
17390 \newcommand*{\glentryuservi}[1]{%
17391   \@gls@entry@field{#1}{uservi}}%
17392 }
17393 \newrobustcmd*{\Glsentryuservi}[1]{%
17394   \@Gls@entry@field{#1}{uservi}}%
17395 }
17396 \newcommand*{\glentryshort}[1]{\@gls@entry@field{#1}{short}}

```

```

17397 \newrobustcmd*{\Glsentryshort}[1]{%
17398   \@Gls@entry@field{#1}{short}%
17399 }
17400 \newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
17401 \newrobustcmd*{\Glsentryshortpl}[1]{%
17402   \@Gls@entry@field{#1}{shortpl}%
17403 }
17404 \newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
17405 \newrobustcmd*{\Glsentrylong}[1]{%
17406   \@Gls@entry@field{#1}{long}%
17407 }
17408 \newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
17409 \newrobustcmd*{\Glsentrylongpl}[1]{%
17410   \@Gls@entry@field{#1}{longpl}%
17411 }
17412 \newcommand*{\glsentryfull}[1]{%
17413   \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
17414 }
17415 \newrobustcmd*{\Glsentryfull}[1]{%
17416   \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
17417 }
17418 \newcommand*{\glsentryfullpl}[1]{%
17419   \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
17420 }
17421 \newrobustcmd*{\Glsentryfullpl}[1]{%
17422   \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
17423 }
17424 \newcommand*{\glsentrynumberlist}[1]{%
17425   \glsdoifexists{#1}%
17426   {%
17427     \@gls@entry@field{#1}{numberlist}%
17428   }%
17429 }
17430 \@ifpackageloaded{hyperref} {%
17431   \newcommand*{\glsdisplaynumberlist}[1]{%
17432     \GlossariesWarning
17433     {%
17434       \string\glsdisplaynumberlist\space
17435       doesn't work with hyperref.^^JUsing
17436       \string\glsentrynumberlist\space instead%
17437     }%
17438     \glsentrynumberlist{#1}%
17439   }%
17440 }%
17441 {%
17442   \newcommand*{\glsdisplaynumberlist}[1]{%
17443     \glsdoifexists{#1}%
17444     {%
17445       \bgroup

```



```

17495 }
17496 \newcommand*{\@gls@adjustmode}{%
17497 \AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
17498 \newrobustcmd*{\glsaddall}[1][1]{%
17499   \edef\@glo@type{\@glo@types}%
17500   \setkeys{glossadd}{#1}%
17501   \forallglsentries[\@glo@type]{\@glo@entry}{%
17502     \glsadd[#1]{\@glo@entry}%
17503   }%
17504 }
17505 \newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
17506   \forallglsentries[#1]{\@glo@entry}%
17507   {%
17508     \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
17509   }%
17510 }
17511 \newcommand*{\glsignore}[1]{%
17512   \edef\glsopenbrace{\expandafter\@gobble\string\{ }
17513   \edef\glsclosebrace{\expandafter\@gobble\string\} }
17514   \edef\glsbackslash{\expandafter\@gobble\string\ }
17515   \edef\glsquote#1{\string"#1\string"}
17516   \edef\glspercentchar{\expandafter\@gobble\string\% }
17517   \edef\glstildechar{\string~ }
17518   \ifglxsindy
17519     \newcommand*{\@glsfirstletter}{A}
17520 \fi
17521 \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
17522   \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
17523 \ifglxsindy
17524   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
17525     \renewcommand*{\@glsfirstletter}{#1}}
17526   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
17527     \renewcommand*{\@glsfirstletter}{#1}%
17528     \@onelevel@sanitize\@glsfirstletter
17529   }
17530 \else
17531   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
17532     \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
17533   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
17534     \@GlsSetXdyFirstLetterAfterDigits
17535   }
17536 \fi
17537 \ifglxsindy
17538   \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
17539 \fi
17540 \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
17541   \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
17542 }
17543 \ifglxsindy

```



```

17544 \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
17545   \renewcommand*{\@xdynumbergrouporder}{#1}%
17546 }
17547 \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
17548   \renewcommand*{\@xdynumbergrouporder}{#1}%
17549   \@onelevel@sanitize\@xdynumbergrouporder
17550 }
17551 \else
17552   \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
17553     \glsnxindywarning\GlsSetXdyNumberGroupOrder}
17554   \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
17555     \@GlsSetXdyNumberGroupOrder}
17556 \fi
17557 \newcommand*{\@glxminrange}{2}
17558 \ifglsxindy
17559   \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
17560     \renewcommand*{\@glxminrange}{#1}}
17561 \else
17562   \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
17563     \glsnxindywarning\GlsSetXdyMinRangeLength}
17564 \fi
17565
17566 \ifglsxindy
17567   \def\writeist{%
17568     \ifundef{\glswrite}{\newwrite\glswrite}{}%
17569     \@gl@addpredefinedattributes
17570     \openout\glswrite=\istfilename
17571     \write\glswrite{;; xindy style file created by the glossaries
17572       package}%
17573     \write\glswrite{;; for document '\jobname' on
17574       \the\year-\the\month-\the\day}%
17575     \write\glswrite{^^J; required styles^^J}
17576     \@for\@xdystyle:=\@xdyrequiredstyles\do{%
17577       \ifx\@xdystyle\@empty
17578       \else
17579         \protected@write\glswrite{{(require
17580           \string"\@xdystyle.xdy\string")}}%
17581       \fi
17582     }%
17583     \write\glswrite{^^J%
17584       ; list of allowed attributes (number formats)^^J}%
17585     \write\glswrite{(define-attributes ((\@xdyattributes)))}%
17586     \write\glswrite{^^J; user defined alphabets^^J}%
17587     \write\glswrite{\@xdyuseralphabets}%
17588     \write\glswrite{^^J; location class definitions^^J}%
17589     \@for\@gl@classI:=\@gl@xdy@locationlist\do{%
17590       \protected@write\glswrite{{(define-location-class
17591         \string"\@gl@classI\string"^^J\space\space\space
17592         (

```

```

17593         :sep "{}{"
17594         \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
17595         :sep "}"
17596     )
17597     ^^J\space\space\space
17598     :min-range-length \@glsminrange^^J%
17599 )
17600 }%
17601 {%
17602     \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
17603         \protected@write\glswrite{}\{(define-location-class
17604             \string"\@gls@classII-\@gls@classI\string"
17605             ^^J\space\space\space
17606             (
17607                 :sep "{"
17608                 \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
17609                 :sep "{}{"
17610                 \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
17611                 :sep "}"
17612             )
17613             ^^J\space\space\space
17614             :min-range-length \@glsminrange^^J%
17615         )
17616     }%
17617 }%
17618 }%
17619 }%
17620 \write\glswrite{^^J; user defined location classes}%
17621 \write\glswrite{\@xdyuserlocationdefs}%
17622 \write\glswrite{^^J; define cross-reference class^^J}%
17623 \write\glswrite{(define-crossref-class \string"see\string"
17624     :unverified )}%
17625 \write\glswrite{(markup-crossref-list
17626     :class \string"see\string"^^J\space\space\space
17627     :open \string"\string\glsseeformat\string"
17628     :close \string"{}\string")}%
17629 \@xdycrossrefhook
17630 \write\glswrite{^^J; define the order of the location classes}%
17631 \write\glswrite{(define-location-class-order
17632     (\@xdylocationclassorder))}%
17633 \write\glswrite{^^J; define the glossary markup^^J}%
17634 \write\glswrite{(markup-index^^J\space\space\space
17635     :open \string"\string
17636     \glossarysection[\string\glossarytoctitle]{\string
17637     \glossarytitle}\string\glossarypreamble}%
17638 \@for\@this@ctr:=\@xdycounters\do{%
17639     {%
17640         \@for\@this@attr:=\@xdyattributelist\do{%
17641             \protected@write\glswrite{}\{\string\providecommand*%

```

```

17642         \expandafter\string
17643         \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
17644         {%
17645             \string\setentrycounter
17646             [\expandafter\@gobble\string\#1]{\@this@ctr}%
17647             \expandafter\string
17648             \csname\@this@attr\endcsname
17649             {\expandafter\@gobble\string\#2}%
17650         }%
17651     }%
17652 }%
17653 }%
17654 }%
17655 \write\glswrite{%
17656     \string\begin
17657     {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
17658     \space\space:close \string"\glspercentchar\glstildechar n\string
17659     \end{theglossary}\string\glossarypostamble
17660     \glstildechar n\string" ^^J\space\space\space
17661     :tree)}}%
17662 \write\glswrite{(markup-letter-group-list
17663     :sep \string"\string\glsgroupskip\glstildechar n\string"))%
17664 \write\glswrite{(markup-indexentry
17665     :open \string"\string\relax \string\glsresetentrylist
17666     \glstildechar n\string"))%
17667 \write\glswrite{(markup-locclass-list :open
17668     \string"\glsopenbrace\string\glossaryentrynumbers
17669     \glsopenbrace\string\relax\space \string"^^J\space\space\space
17670     :sep \string", \string"
17671     :close \string"\glsclosebrace\glsclosebrace\string"))%
17672 \write\glswrite{(markup-locref-list
17673     :sep \string"\string\delimN\space\string"))%
17674 \write\glswrite{(markup-range
17675     :sep \string"\string\delimR\space\string"))%
17676 \@onelevel@sanitize\gls@suffixF
17677 \@onelevel@sanitize\gls@suffixFF
17678 \ifx\gls@suffixF\@empty
17679 \else
17680     \write\glswrite{(markup-range
17681         :close "\gls@suffixF" :length 1 :ignore-end)}}%
17682 \fi
17683 \ifx\gls@suffixFF\@empty
17684 \else
17685     \write\glswrite{(markup-range
17686         :close "\gls@suffixFF" :length 2 :ignore-end)}}%
17687 \fi
17688 \write\glswrite{^^J; define format to use for locations^^J}%
17689 \write\glswrite{\@xdylocref}%
17690 \write\glswrite{^^J; define letter group list format^^J}%

```

```

17691 \write\glswrite{(markup-letter-group-list
17692 :sep \string"\string\glsgroupskip\glstildechar n\string")}%
17693 \write\glswrite{^^J; letter group headings^^J}%
17694 \write\glswrite{(markup-letter-group
17695 :open-head \string"\string\glsgroupheading
17696 \glsoopenbrace\string"^^J\space\space\space
17697 :close-head \string"\glsclosebrace\string")}%
17698 \write\glswrite{^^J; additional letter groups^^J}%
17699 \write\glswrite{@xdylettergroups}%
17700 \write\glswrite{^^J; additional sort rules^^J}
17701 \write\glswrite{@xdysortrules}%
17702 \@gls@writeisthook
17703 \closeout\glswrite
17704 \let\writeist\relax
17705 }
17706 \else
17707 \edef\@gls@actualchar{\string?}
17708 \edef\@gls@encapchar{\string|}
17709 \edef\@gls@levelchar{\string!}
17710 \edef\@gls@quotechar{\string"}%
17711 \let\GlsSetQuote\gls@nosetquote
17712 \def\writeist{\relax
17713 \ifundef{\glswrite}{\newwrite\glswrite}{\relax
17714 \openout\glswrite=\istfilename
17715 \write\glswrite{\glspersentchar\space makeindex style file
17716 created by the glossaries package}
17717 \write\glswrite{\glspersentchar\space for document
17718 '\jobname' on \the\year-\the\month-\the\day}
17719 \write\glswrite{actual '@gls@actualchar'}
17720 \write\glswrite{encap '@gls@encapchar'}
17721 \write\glswrite{level '@gls@levelchar'}
17722 \write\glswrite{quote '@gls@quotechar'}
17723 \write\glswrite{keyword \string"\string\glossaryentry\string"}
17724 \write\glswrite{preamble \string"\string\glossarysection[\string
17725 \glossarytoctitle]{\string\glossarytitle}\string
17726 \glossarypreamble\string\n\string\begin{theglossary}\string
17727 \glossaryheader\string\n\string"}
17728 \write\glswrite{postamble \string"\string%\string\n\string
17729 \end{theglossary}\string\glossarypostamble\string\n
17730 \string"}
17731 \write\glswrite{group_skip \string"\string\glsgroupskip\string\n
17732 \string"}
17733 \write\glswrite{item_0 \string"\string%\string\n\string"}
17734 \write\glswrite{item_1 \string"\string%\string\n\string"}
17735 \write\glswrite{item_2 \string"\string%\string\n\string"}
17736 \write\glswrite{item_01 \string"\string%\string\n\string"}
17737 \write\glswrite{item_x1
17738 \string"\string\relax \string\glsresetentrylist\string\n
17739 \string"}

```

```

17740 \write\glswrite{item_12 \string"\string%\string\n\string"}
17741 \write\glswrite{item_x2
17742 \string"\string\relax \string\glresetentrylist\string\n
17743 \string"}
17744 \write\glswrite{delim_0 \string"\string\{\string
17745 \glossaryentrynumbers\string\{\string\relax \string"}
17746 \write\glswrite{delim_1 \string"\string\{\string
17747 \glossaryentrynumbers\string\{\string\relax \string"}
17748 \write\glswrite{delim_2 \string"\string\{\string
17749 \glossaryentrynumbers\string\{\string\relax \string"}
17750 \write\glswrite{delim_t \string"\string\}\string\}\string"}
17751 \write\glswrite{delim_n \string"\string\delimN \string"}
17752 \write\glswrite{delim_r \string"\string\delimR \string"}
17753 \write\glswrite{headings_flag 1}
17754 \write\glswrite{heading_prefix
17755 \string"\string\glsgroupheading\string\{\string"}
17756 \write\glswrite{heading_suffix
17757 \string"\string\}\string\relax
17758 \string\glresetentrylist \string"}
17759 \write\glswrite{symhead_positive \string"glssymbols\string"}
17760 \write\glswrite{numhead_positive \string"glnumbers\string"}
17761 \write\glswrite{page_compositor \string"glscpositor\string"}
17762 \@gls@escbsdq\gls@suffixF
17763 \@gls@escbsdq\gls@suffixFF
17764 \ifx\gls@suffixF\@empty
17765 \else
17766 \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
17767 \fi
17768 \ifx\gls@suffixFF\@empty
17769 \else
17770 \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
17771 \fi
17772 \@gls@writeisthook
17773 \closeout\glswrite
17774 \let\writeist\relax
17775 }
17776 \fi
17777 \newcommand*{\GlsSetWriteIstHook}[1]{\renewcommand*{\@gls@writeisthook}{#1}}
17778 \@onlypremakeg\GlsSetWriteIstHook
17779 \newcommand*{\@gls@writeisthook}{\}
17780 \ifglxindy
17781 \newcommand*{\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
17782 \newcommand*{\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
17783 \else
17784 \newcommand*{\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
17785 \@ifpackageloaded{tracklang}%
17786 {%
17787 \IfTrackedLanguage{german}%
17788 {%

```

```

17789     \def\@gls@extramakeindexopts{-g}%
17790 }%
17791 {}%
17792 }%
17793 {}%
17794 \edef\@gls@docheckquotedef{%
17795     \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
17796         \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
17797         \noexpand\toks@={####1}%
17798         \noexpand\ifx\noexpand\null####2\noexpand\null
17799         \noexpand\ifx\noexpand\null####3\noexpand\null
17800         \noexpand\edef\noexpand\@gls@checkedmkidx{%
17801             \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
17802         \noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
17803     \noexpand\else
17804         \noexpand\edef\noexpand\@gls@checkedmkidx{%
17805             \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17806             \noexpand\@gls@quotechar\noexpand\@gls@quotechar
17807             \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
17808         \noexpand\def\noexpand\@gls@checkquote{%
17809             \noexpand\@gls@checkquote####3\noexpand\null}%
17810     \noexpand\fi
17811 \noexpand\else
17812     \noexpand\edef\noexpand\@gls@checkedmkidx{%
17813         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17814         \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
17815     \noexpand\ifx\noexpand\null####3\noexpand\null
17816     \noexpand\def\noexpand\@gls@checkquote{%
17817         \noexpand\@gls@checkquote####2#1#1\noexpand\null}%
17818     \noexpand\else
17819     \noexpand\def\noexpand\@gls@checkquote{%
17820         \noexpand\@gls@checkquote####2#1####3\noexpand\null}%
17821     \noexpand\fi
17822 \noexpand\fi
17823 \noexpand\@gls@checkquote
17824 }%
17825 }%
17826 \@gls@docheckquotedef
17827 \edef\@gls@docheckquotedef{%
17828     \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
17829         \noexpand\def\noexpand\@gls@checkedmkidx{%
17830             \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
17831             #1#1\noexpand\null
17832             \noexpand\expandafter\noexpand\@gls@updatechecked
17833             \noexpand\@gls@checkedmkidx{####1}%
17834             \noexpand\def\noexpand\@gls@checkedmkidx{%
17835                 \noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
17836                 \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
17837                 \noexpand\null

```

```

17838 \noexpand\expandafter\noexpand\@gls@updatechecked
17839 \noexpand\@gls@checkedmkidx{####1}%
17840 \noexpand\def\noexpand\@gls@checkedmkidx{%
17841 \noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
17842 \noexpand\?\noexpand\?\noexpand\null
17843 \noexpand\expandafter\noexpand\@gls@updatechecked
17844 \noexpand\@gls@checkedmkidx{####1}%
17845 \noexpand\def\noexpand\@gls@checkedmkidx{%
17846 \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
17847 \noexpand?\noexpand?\noexpand\null
17848 \noexpand\expandafter\noexpand\@gls@updatechecked
17849 \noexpand\@gls@checkedmkidx{####1}%
17850 \noexpand\def\noexpand\@gls@checkedmkidx{%
17851 \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
17852 \noexpand|\noexpand|\noexpand\null
17853 \noexpand\expandafter\noexpand\@gls@updatechecked
17854 \noexpand\@gls@checkedmkidx{####1}%
17855 \noexpand\def\noexpand\@gls@checkedmkidx{%
17856 \noexpand\expandafter\noexpand\@gls@checkescbar####1\noexpand\@nil
17857 \noexpand||\noexpand||\noexpand\null
17858 \noexpand\expandafter\noexpand\@gls@updatechecked
17859 \noexpand\@gls@checkedmkidx{####1}%
17860 \noexpand\def\noexpand\@gls@checkedmkidx{%
17861 \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
17862 \noexpand!\noexpand!\noexpand\null
17863 \noexpand\expandafter\noexpand\@gls@updatechecked
17864 \noexpand\@gls@checkedmkidx{####1}%
17865 }%
17866 }%
17867 \@gls@docheckquotedef
17868 \edef\@gls@docheckquotedef{%
17869 \noexpand\def\noexpand\@gls@checkescquote####1%
17870 \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
17871 ####3\noexpand\null{%
17872 \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
17873 \noexpand\toks@={####1}%
17874 \noexpand\ifx\noexpand\null####2\noexpand\null
17875 \noexpand\ifx\noexpand\null####3\noexpand\null
17876 \noexpand\edef\noexpand\@gls@checkedmkidx{%
17877 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
17878 \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
17879 \noexpand\else
17880 \noexpand\edef\noexpand\@gls@checkedmkidx{%
17881 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17882 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
17883 \csname#1\endcsname}\noexpand\@gls@quotechar
17884 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
17885 \csname#1\endcsname}\noexpand\@gls@quotechar}%
17886 \noexpand\def\noexpand\@gls@checkescquote{%

```

```

17887         \noexpand\@gls@checkescquote####3\noexpand\null}%
17888     \noexpand\fi
17889     \noexpand\else
17890     \noexpand\edef\noexpand\@gls@checkedmkidx{%
17891         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17892         \noexpand\@gls@quotearch\noexpand\string
17893         \expandonce{\csname#1\endcsname}\noexpand\@gls@quotearch}%
17894     \noexpand\ifx\noexpand\null####3\noexpand\null
17895     \noexpand\def\noexpand\@gls@checkescquote{%
17896         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
17897         \expandonce{\csname#1\endcsname}\noexpand\null}%
17898     \noexpand\else
17899     \noexpand\def\noexpand\@gls@checkescquote{%
17900         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
17901         ####3\noexpand\null}%
17902     \noexpand\fi
17903     \noexpand\fi
17904     \noexpand\@gls@checkescquote
17905 }%
17906 }%
17907 \@gls@docheckquotedef
17908 }
17909 \newcommand*{\@gls@nosetquote}[1]{\PackageError{glossaries}%
17910     {\string\GlsSetQuote\space not permitted here}%
17911     {\Move \string\GlsSetQuote\space earlier in the preamble, as
17912     soon as possible after glossaries.sty has been loaded}}
17913 \fi
17914 \newcommand*{\@gls@extramakeindexopts}[1]{%
17915 \newcommand{\noist}{%
17916     \@gls@addpredefinedattributes
17917     \let\writeist\relax
17918 }
17919 \newcommand*{\@makeglossary}[1]{%
17920     \ifglossaryexists{#1}%
17921     {%
17922         \ifglssavewrites
17923         \expandafter\newtoks\csname glo@#1@filetok\endcsname
17924         \else
17925         \expandafter\newwrite\csname glo@#1@file\endcsname
17926         \expandafter\@gls@openfile\csname glo@#1@file\endcsname{#1}%
17927         \fi
17928         \@gls@renewglossary
17929         \writeist
17930     }%
17931     {%
17932         \PackageError{glossaries}%
17933         {Glossary type ‘#1’ not defined}%
17934         {New glossaries must be defined before using \string\makeglossaries}%
17935     }%

```



```

17936 }
17937 \newcommand*{\@glsopenfile}[2]{%
17938   \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
17939   \PackageInfo{glossaries}{Writing glossary file
17940     \jobname.\csname @glotype@#2@out\endcsname}%
17941 }
17942 \newcommand*{\@closegls}[1]{%
17943   \closeout\csname glo@#1@file\endcsname
17944 }
17945 \ifglxindy
17946   \newcommand*{\@gls@automake}[1]{%
17947     \ifglossaryexists{#1}
17948     {%
17949       \@closegls{#1}%
17950       \ifdefstring{\glsorder}{letter}%
17951       {\def\@gls@order{-M ord/letorder }}%
17952       {\let\@gls@order\@empty}%
17953       \ifcsundef{\xdy@#1@language}%
17954       {\let\@gls@langmod\xdy@main@language}%
17955       {\letcs\@gls@langmod{\xdy@#1@language}}%
17956       \edef\@gls@dothiswrite{\noexpand\write18{xindy
17957         -I xindy
17958         \@gls@order
17959         -L \@gls@langmod\space
17960         -M \@gls@istfilebase\space
17961         -C \@gls@codepage\space
17962         -t \jobname.\csuse{@glotype@#1@log}
17963         -o \jobname.\csuse{@glotype@#1@in}
17964         \jobname.\csuse{@glotype@#1@out}}}%
17965     }%
17966     \@gls@dothiswrite
17967   }%
17968   {%
17969     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17970   }%
17971 }
17972 \else
17973   \newcommand*{\@gls@automake}[1]{%
17974     \ifglossaryexists{#1}
17975     {%
17976       \@closegls{#1}%
17977       \ifdefstring{\glsorder}{letter}%
17978       {\def\@gls@order{-l }}%
17979       {\let\@gls@order\@empty}%
17980       \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
17981         -s \istfilename\space
17982         -t \jobname.\csuse{@glotype@#1@log}
17983         -o \jobname.\csuse{@glotype@#1@in}
17984         \jobname.\csuse{@glotype@#1@out}}}%

```

```

17985     }%
17986     \@gls@dothiswrite
17987 }%
17988 {%
17989     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17990 }%
17991 }
17992 \fi
17993 \ifglsxindy
17994 \newcommand*{\@gls@automake@immediate}[1]{%
17995     \ifglossaryexists{#1}
17996     {%
17997         \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
17998             {%
17999                 \ifdefstring{\glsorder}{letter}%
18000                 {\def\@gls@order{-M ord/letorder }}%
18001                 {\let\@gls@order\empty}%
18002                 \ifcsundef{xdy@#1@language}%
18003                 {\let\@gls@langmod\@xdy@main@language}%
18004                 {\letcs\@gls@langmod{\@xdy@#1@language}}%
18005                 \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
18006                     -I xindy
18007                     \@gls@order
18008                     -L \@gls@langmod\space
18009                     -M \gls@istfilebase\space
18010                     -C \gls@codepage\space
18011                     -t \jobname.\csuse{@glotype@#1@log}
18012                     -o \jobname.\csuse{@glotype@#1@in}
18013                     \jobname.\csuse{@glotype@#1@out}}}%
18014             }%
18015             \@gls@dothiswrite
18016         }%
18017         {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
18018             doesn't exist. Rerun may be required}}%
18019     }%
18020 }%
18021     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
18022 }%
18023 }
18024 \else
18025 \newcommand*{\@gls@automake@immediate}[1]{%
18026     \ifglossaryexists{#1}
18027     {%
18028         \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
18029             {%
18030                 \ifdefstring{\glsorder}{letter}%
18031                 {\def\@gls@order{-l }}%
18032                 {\let\@gls@order\empty}%
18033                 \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order

```

```

18034         -s \istfilename\space
18035         -t \jobname.\csuse{@glotype@#1@log}
18036         -o \jobname.\csuse{@glotype@#1@in}
18037         \jobname.\csuse{@glotype@#1@out}}}%
18038     }%
18039     \@gls@dothiswrite
18040 }%
18041 {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
18042     doesn't exist. Rerun may be required}}}%
18043 }%
18044 {%
18045     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
18046 }%
18047 }
18048 \fi
18049 \newcommand*{\@warn@nomakeglossaries}{}
18050 \newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
18051 \newcommand{\@gls@@automake@immediate}{%
18052     \ifnum\gls@automake@nr=2\relax
18053         \@for\@gls@type:=\@glo@types\do{%
18054             \ifdefempty{\@gls@type}{}%
18055             {\@gls@automake@immediate{\@gls@type}}%
18056         }%
18057         \glsautomakefalse
18058         \renewcommand*{\@gls@doautomake}{}%
18059     \fi
18060 }
18061 \newcommand*{\makeglossaries}{%
18062     \@domakeglossaries
18063 {%
18064     \@gls@@automake@immediate
18065     \ifundef{\glswrite}{\newwrite\glswrite}{}%
18066     \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
18067     \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
18068     \ifundef\@gls@extramakeindexopts
18069     {}%
18070     {%
18071         \protected@write\@auxout{}{\string\providecommand
18072             \string\@gls@extramakeindexopts[1]{} }
18073         \protected@write\@auxout{}{\string\@gls@extramakeindexopts
18074             {\@gls@extramakeindexopts}}%
18075     }%
18076     \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
18077     \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
18078     \@for\@glo@type:=\@glo@types\do{%
18079         \ifthenelse{equal{\@glo@type}{} }{}{%
18080             \@makeglossary{\@glo@type}}%
18081     }%
18082     \renewcommand*{\newglossary[4] []}{%

```

```

18083 \PackageError{glossaries}{New glossaries
18084 must be created before \string\makeglossaries}{You need
18085 to move \string\makeglossaries\space after all your
18086 \string\newglossary\space commands}}%
18087 \let\@makeglossary\@gobble
18088 \let\makeglossaries\relax
18089 \@disable@onlypremakeg
18090 \let\gls@checkseeallowed\relax
18091 \let\warn@nomakeglossaries\relax
18092 \def\warn@noprintglossary{%
18093   \ifdefstring{\@glo@types}{,}%
18094   {%
18095     \GlossariesWarningNoLine{No glossaries have been defined}%
18096   }%
18097   {%
18098     \GlossariesWarningNoLine{No \string\printglossary\space
18099       or \string\printglossaries\space
18100       found. ^^J(Remove \string\makeglossaries\space if you
18101       don't want any glossaries.) ^^JThis document will not
18102       have a glossary}%
18103   }%
18104 }%
18105 \ifglssavenumberlist
18106   \edef\@gls@dodeflistparser{\noexpand\DeclareListParser
18107     {\noexpand\glsnumlistparser}{\delimN}}%
18108   \@gls@dodeflistparser
18109 \fi
18110 \let\makenoidxglossaries\@no@makeglossaries
18111 \renewcommand*{\@printgloss@setsort}{%
18112   \let\@glo@assign@sortkey\@glo@no@assign@sortkey
18113 }%
18114 \ifglssautomake
18115   \renewcommand*{\@gls@doautomake}{%
18116     \@for\@gls@type:=\@glo@types\do{%
18117       \ifdefempty{\@gls@type}{}%
18118       {\@gls@automake{\@gls@type}}%
18119     }%
18120   }%
18121 \fi
18122 \@glo@check@sortallowed\makeglossaries
18123 }%
18124 }
18125 \@onlypreamble{\makeglossaries}
18126 \AtEndDocument{%
18127   \warn@nomakeglossaries
18128   \warn@noprintglossary
18129 }
18130 \newcommand*{\makenoidxglossaries}{%
18131   \@domakeglossaries

```

```

18132 {%
18133 \renewcommand{\@gls@noref@warn}[1]{%
18134 \GlossariesWarning{Empty glossary for
18135 \string\printnoidxglossary[type={##1}].
18136 Rerun may be required (or you may have forgotten to use
18137 commands like \string\gls)}}%
18138 }%
18139 \let\@gls@checkmkidxchars\@gobble
18140 \glsclocationsfalse
18141 \let\@do@@wrglossary\gls@noidxglossary
18142 \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
18143 \let\gls@checkseeallowed\relax
18144 \renewcommand{\@do@seeglossary}[2]{%
18145 \edef\@gls@label{\glsdetoklabel{##1}}%
18146 \protected@write\@auxout{}{%
18147 \string\@gls@reference
18148 {\csname glo@\@gls@label @type\endcsname}%
18149 {\@gls@label}%
18150 {%
18151 \string\glsseeformat##2}%
18152 }%
18153 }%
18154 }%
18155 \AtBeginDocument
18156 {%
18157 \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
18158 }%
18159 \def\warn@noprintglossary{%
18160 \GlossariesWarningNoLine{No \string\printnoidxglossary\space
18161 or \string\printnoidxglossaries ^^J
18162 found. (Remove \string\makenoidxglossaries\space if you
18163 don't want any glossaries.)^^JThis document will not have a glossary}%
18164 }%
18165 \let\warn@nomakeglossaries\relax
18166 \let\makeglossaries\@no@makeglossaries
18167 \renewcommand*\@printgloss@setsort{%
18168 \let\@glo@assign@sortkey\@glo@assign@sortkey
18169 \def\@glo@sorttype{\@glo@default@sorttype}%
18170 }%
18171 \renewcommand*\new@glossaryentry[2]{%
18172 \PackageError{glossaries}{Glossary entries must be
18173 defined in the preamble^^Jwhen you use
18174 \string\makenoidxglossaries}%
18175 {Either move your definitions to the preamble or use
18176 \string\makeglossaries}%
18177 }%
18178 \renewcommand*\@glsentrynumberlist}[1]{%
18179 \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18180 \ifdef\@gls@loclist

```

```

18181 {%
18182     \glsnoidxloclist{\@gls@loclist}%
18183 }%
18184 {%
18185     ??\glsdoifexists{##1}%
18186     {%
18187         \GlossariesWarning{Missing location list for ‘##1’. Either
18188             a rerun is required or you haven’t referenced the entry}%
18189     }%
18190 }%
18191 }%
18192 \renewcommand*{\glsdisplaynumberlist}[1]{%
18193     \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18194     \ifdef\@gls@loclist
18195     {%
18196         \def\@gls@noidxloclist@sep{%
18197             \def\@gls@noidxloclist@sep{%
18198                 \def\@gls@noidxloclist@sep{%
18199                     \glsnumlistsep
18200                 }%
18201                 \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
18202             }%
18203         }%
18204         \def\@gls@noidxloclist@finalsep{}%
18205         \def\@gls@noidxloclist@prev{}%
18206         \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
18207         \@gls@noidxloclist@finalsep
18208         \@gls@noidxloclist@prev
18209     }%
18210     {%
18211         ??\glsdoifexists{##1}%
18212         {%
18213             \GlossariesWarning{Missing location list for ‘##1’. Either
18214                 a rerun is required or you haven’t referenced the entry}%
18215         }%
18216     }%
18217 }%
18218 \renewcommand*{\glsnumberlistloop}[3]{%
18219     \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18220     \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
18221     \let\@gls@org@glsseeformat\glsseeformat
18222     \let\glsnoidxdisplayloc##2\relax
18223     \let\glsseeformat##3\relax
18224     \ifdef\@gls@loclist
18225     {%
18226         \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
18227     }%
18228     {%
18229         ??\glsdoifexists{##1}%

```

```

18230      {%
18231          \GlossariesWarning{Missing location list for ‘##1’. Either
18232              a rerun is required or you haven’t referenced the entry}%
18233      }%
18234  }%
18235      \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
18236      \let\glsseeformat\@gls@org@glsseeformat
18237  }%
18238      \let\@gls@sanitizesort\@gls@noidx@sanitizesort
18239      \let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
18240      \@gls@noidx@setsanitizesort
18241      \@glo@check@sortallowed\makenoidxglossaries
18242  }%
18243 }
18244 \onlypreamble{\makenoidxglossaries}
18245 \newcommand*{\glsnumberlistloop}[2]{%
18246     \PackageError{glossaries}{\string\glsnumberlistloop\space
18247         only works with \string\makenoidxglossaries}{}%
18248 }
18249 \newcommand*{\glsnoidxnumberlistloop}[1]{%
18250     #1%
18251 }
18252 \newcommand*{\@no@makeglossaries}{%
18253     \PackageError{glossaries}{You can’t use both
18254         \string\makeglossaries\space and \string\makenoidxglossaries}%
18255     {Either use one or other (or none) of those commands but not both
18256         together.}%
18257 }
18258 \newcommand{\@gls@noref@warn}[1]{%
18259     \GlossariesWarning{\string\makenoidxglossaries\space
18260         is required to make \string\printnoidxglossary[type={#1}] work}%
18261 }
18262 \newcommand*{\gls@noidxglossary}{%
18263     \protected@write\@auxout{}{%
18264         \string\@gls@reference
18265             {\csname glo@\@gls@label @type\endcsname}%
18266             {\@gls@label}%
18267             {\string\glsnoidxdisplayloc
18268                 {\@glo@counterprefix}%
18269                 {\@gls@counter}%
18270                 {\@glsnumberformat}%
18271                 {\@glslocref}%
18272             }%
18273     }%
18274 }
18275 \providecommand\istfile{\glswrite}
18276 \AtEndDocument{%
18277     \glswritefiles
18278 }

```

```

18279 \newcommand*{\@glswritefiles}{%
18280   \forallglossaries{\@glo@type}{%
18281     \ifcsundef{glo@\@glo@type @filetok}%
18282     {%
18283       \def\gls@tmp{}%
18284     }%
18285     {%
18286       \edef\gls@tmp{\expandafter\the
18287         \csname glo@\@glo@type @filetok\endcsname}%
18288     }%
18289     \ifx\gls@tmp\@empty
18290       \ifx\@glo@type\glsdefaulttype
18291         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
18292           entries.^^JRemember to use package option 'nomain' if
18293 you
18294           don't want to^^Juse the main glossary}%
18295       \else
18296         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
18297           entries}%
18298       \fi
18299     \else
18300       \@glsopenfile{\glswrite}{\@glo@type}%
18301       \immediate\write\glswrite{%
18302         \expandafter\the
18303         \csname glo@\@glo@type @filetok\endcsname}%
18304       \immediate\closeout\glswrite
18305     \fi
18306   }%
18307 }
18308 \newcommand*{\gls@glossary}[1]{%
18309   \@gls@glossary{#1}%
18310 }
18311 \newcommand*{\@gls@glossary}[2]{%
18312   \if@gls@debug
18313     \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
18314   \fi
18315 }
18316 \newcommand{\@gls@renewglossary}{%
18317   \gdef\@gls@glossary##1{\@bsphack\beginingroup\gls@wrglossary{##1}}%
18318   \let\@gls@renewglossary\@empty
18319 }
18320 \newcommand*{\gls@wrglossary}[2]{%
18321   \ifglssavewrites
18322     \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
18323     \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
18324     \expandafter{\@gls@tmp^^J}%
18325   \else
18326     \ifcsdef{glo@#1@file}%
18327     {%

```



```

18328     \expandafter\protected@write\csname glo@#1@file\endcsname{%
18329     \gls@disablepagerefexpansion}{#2}%
18330 }%
18331 {%
18332     \ifignoredglossary{#1}{}%
18333     {%
18334         \GlossariesWarning{No file defined for glossary ‘#1’}%
18335     }%
18336 }%
18337 \fi
18338 \endgroup\@esphack
18339 }
18340 \newcommand*{\@do@wrglossary}[1]{%
18341     \glswriteentry{#1}{\@do@wrglossary{#1}}%
18342 }
18343 \newcommand*{\glswriteentry}[2]{%
18344     \ifglsindexonlyfirst
18345     \ifglsused{#1}{#2}%
18346     \else
18347         #2%
18348     \fi
18349 }
18350 \newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
18351 \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
18352 \newcommand*{\gls@disablepagerefexpansion}{%
18353     \@for\@gls@this:=\gls@protected@pagefmts\do
18354     {%
18355         \expandafter\let\@gls@this\relax
18356     }%
18357 }
18358 \newcommand*{\gls@alphpage}{\@alph\c@page}
18359 \newcommand*{\gls@Alphpage}{\@Alph\c@page}
18360 \newcommand*{\gls@numberpage}{\number\c@page}
18361 \newcommand*{\gls@arabicpage}{\@arabic\c@page}
18362 \newcommand*{\gls@romanpage}{\romannumeral\c@page}
18363 \newcommand*{\gls@Romanpage}{\@Roman\c@page}
18364 \newcommand*{\glsaddprotectedpagefmt}[1]{%
18365     \eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
18366     \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
18367     \eappto\@wrglossarynumberhook{%
18368         \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
18369         \expandonce{\csname#1\endcsname}%
18370         \noexpand\def\expandonce{\csname#1\endcsname}{%
18371             \noexpand\@wrglossary@pageformat
18372             \expandonce{\csname gls#1page\endcsname}%
18373             \expandonce{\csname org@gls#1\endcsname}%
18374         }%
18375     }%
18376 }

```

```

18377 \newcommand*\@wrglossarynumberhook{}
18378 \newcommand{\@wrglossary@pageformat}[3]{%
18379   \ifx#3\c@page #1\else #2#3\fi
18380 }
18381 \newcommand*{\@do@wrglossary}[1]{%
18382   \ifglswrallowlocations
18383     \@do@esc@wrglossary{#1}%
18384   \else
18385     \@do@noesc@wrglossary{#1}%
18386   \fi
18387 }
18388 \newcommand*{\@do@noesc@wrglossary}[1]{%
18389   \expandafter\def\expandafter\@glslocref\expandafter{\theglslsentrycounter}%
18390   \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglslsentrycounter}%
18391   \ifx\@glsHlocref\@glslocref
18392     \def\@glo@counterprefix{}%
18393   \else
18394     \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
18395       {\@glslocref}{\@glsHlocref}}%
18396   }%
18397   \@do@gls@getcounterprefix
18398   \fi
18399   \edef\@gls@label{\glsdetoklabel{#1}}%
18400   \@do@@@wrglossary
18401 }
18402 \newif\ifglswrallowprimitivemods
18403 \glswrallowprimitivemodstrue
18404 \newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
18405   \begingroup
18406     \let\gls@orgthe\the
18407     \let\gls@orgnumber\number
18408     \let\gls@orgarabic\@arabic
18409     \let\gls@orgromannumeral\romannumeral
18410     \let\gls@orgalph\@alph
18411     \let\gls@orgAlph\@Alph
18412     \let\gls@orgRoman\@Roman
18413     \ifglswrallowprimitivemods
18414       \def\gls@the##1{%
18415         \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
18416       \def\the{\expandafter\gls@the}%
18417       \def\gls@number##1{%
18418         \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
18419       \def\number{\expandafter\gls@number}%
18420     \fi
18421     \def\@arabic##1{%
18422       \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
18423     \def\romannumeral##1{%
18424       \ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
18425     \def\@Roman##1{%

```

```

18426 \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
18427 \def\@alph##1{%
18428 \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
18429 \def\@Alph##1{%
18430 \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
18431 \@wrglossarynumberhook
18432 \gls@disablepagerefexpansion
18433 \protected@xdef\@glslocref{\theglsentrycounter}%
18434 \endgroup
18435 \@gls@checkmkidxchars\@glslocref
18436 \expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
18437 \def\@glo@counterprefix{}%
18438 \else
18439 \protected@edef\@glsHlocref{\theglsentrycounter}%
18440 \@gls@checkmkidxchars\@glsHlocref
18441 \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
18442 {\@glslocref}{\@glsHlocref}}%
18443 }%
18444 \@do@gls@getcounterprefix
18445 \fi
18446 \edef\@gls@label{\glsdetoklabel{#1}}%
18447 \@do@wrglossary
18448 }
18449 \newcommand*{\@do@wrglossary}{%
18450 \ifglxsindy
18451 \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
18452 \def\@glo@range{}%
18453 \expandafter\if\@glo@prefix(\relax
18454 \def\@glo@range{:open-range}%
18455 \else
18456 \expandafter\if\@glo@prefix)\relax
18457 \def\@glo@range{:close-range}%
18458 \fi
18459 \fi
18460 \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
18461 (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
18462 :locref \string{\@glo@counterprefix}{\@glslocref}\string" %
18463 :attr \string"\@gls@counter\@glo@suffix\string"
18464 \@glo@range
18465 )
18466 }%
18467 \else
18468 \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
18469 {\@glo@counterprefix}%
18470 \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
18471 \string\glossaryentry{\csname glo@\@gls@label @index\endcsname
18472 \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
18473 \fi
18474 }

```

```

18475 \newcommand*\@gls@getcounterprefix[2]{%
18476   \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
18477   \ifx\@gls@thisloc\@gls@thisHloc
18478     \def\@glo@counterprefix{}%
18479   \else
18480     \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
18481       \def\@glo@tmp{##2}%
18482       \ifx\@glo@tmp\@empty
18483         \def\@glo@counterprefix{}%
18484       \else
18485         \def\@glo@counterprefix{##1}%
18486       \fi
18487     }%
18488     \@gls@get@counterprefix#2.#1\end@getprefix
18489     \ifx\@glo@counterprefix\@empty
18490       \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
18491         prefixing^^Jlocation ‘#1’. You need to modify the
18492         definition of \string\theH\@gls@counter^^Jotherwise you
18493         will get the warning: “‘name{\@gls@counter.#1}’ has been^^J
18494         referenced but does not exist”}%
18495     \fi
18496   \fi
18497 }
18498 \newcommand{\@do@seeglossary}[2]{%
18499 \def\@gls@xref{#2}%
18500 \@onelevel@sanitize\@gls@xref
18501 \@gls@checkmkidxchars\@gls@xref
18502 \ifglxsindy
18503   \gls@glossary{\csname glo@#1@type\endcsname}{%
18504     (indexentry
18505       :tkey (\csname glo@#1@index\endcsname)
18506       :xref (\string"\@gls@xref\string")
18507       :attr \string"see\string"
18508     )
18509   }%
18510 \else
18511   \gls@glossary{\csname glo@#1@type\endcsname}{%
18512     \string@glossaryentry{\csname glo@#1@index\endcsname
18513       \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
18514 \fi
18515 }
18516 \def\@gls@fixbraces#1#2#3\@nil{%
18517   \ifx#2[\relax
18518     \@gls@fixbraces#1#2#3\@end@fixbraces
18519   \else
18520     \def#1{{#2#3}}%
18521   \fi
18522 }
18523 \def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%

```

```

18524 \def#1{[#2]{#3}}%
18525 }
18526 \newrobustcmd*{\glssee}[3][\seename]{%
18527   \@do@seeglossary{#2}{[#1]{#3}}}
18528 \newcommand*{\@glssee}[3][\seename]{%
18529   \glssee[#1]{#3}{#2}}
18530 \newrobustcmd*{\glsseeformat}[3][\seename]{%
18531   \emph{#1} \glsseelist{#2}}
18532 \newrobustcmd*{\glsseelist}[1]{%
18533   \let\@gls@dolast\relax
18534   \let\@gls@donext\relax
18535   \@for\@gls@thislabel:=#1\do{%
18536     \ifx\@xfor@nextelement\@nnil
18537       \@gls@dolast
18538     \else
18539       \@gls@donext
18540     \fi
18541     \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
18542     \let\@gls@dolast\glsseelastsep
18543     \let\@gls@donext\glsseesep
18544   }%
18545 }
18546 \newcommand*{\glsseelastsep}{\space\andname\space}
18547 \newcommand*{\glsseesep}{, }
18548 \newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
18549 \newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
18550 \newcommand*{\gls@save@numberlist}[1]{%
18551   \ifglssavenumberlist
18552     \toks@{#1}%
18553     \edef\@do@writeaux@info{%
18554       \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
18555     }%
18556     \@onelevel@sanitize\@do@writeaux@info
18557     \protected@write\@auxout{}\@do@writeaux@info%
18558   \fi
18559 }
18560 \newcommand*{\warn@noprintglossary}{}%
18561 \ifcsundef{printglossary}{}%
18562 {%
18563   \@gls@warnonglossdefined
18564   \undef\printglossary
18565 }
18566 \newcommand*{\@printgloss@checkexists@noignored}[2]{%
18567   \@ifglossaryexists{#1}%
18568   {#2}%
18569   {%
18570     \ifignoredglossary{#1}%
18571     {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
18572     {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%

```

```

18573 }%
18574 }
18575 \newcommand*{\@printgloss@checkexists@allowignored}[2]{%
18576   \s@ifglossaryexists{#1}%
18577   {#2}%
18578   {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
18579 }
18580 \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18581 \newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
18582   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18583   \@printglossary{#1}{\@print@glossary}%
18584 }
18585 \newcommand*{\printglossaries}{%
18586   \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
18587 }
18588 \newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
18589   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18590   \@printglossary{#1}{\@print@noidx@glossary}%
18591 }
18592 \newcommand*{\printnoidxglossaries}{%
18593   \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
18594 }
18595 \newcommand*{\@printgloss@setsort}{%
18596   \newcommand*{\@gls@preglossaryhook}{%
18597     \newcommand{\@printglossary}[2]{%
18598       \def\@glo@type{\glsdefaulttype}%
18599       \def\glossarytitle{\csname @glotype@\@glo@type @title\endcsname}%
18600       \def\glossarytoctitle{\glossarytitle}%
18601       \let\org@glossarytitle\glossarytitle
18602       \def\@glossarystyle{%
18603         \ifx\@glossary@default@style\relax
18604           \GlossariesWarning{No default glossary style provided \MessageBreak
18605             for the glossary ‘\@glo@type’. \MessageBreak
18606             Using deprecated fallback. \MessageBreak
18607             To fix this set the style with \MessageBreak
18608             \string\setglossarystyle\space or use the \MessageBreak
18609             style key=value option}%
18610         \fi
18611       }%
18612       \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
18613       \let\@org@glossaryentrynumbers\glossaryentrynumbers
18614       \bgroup
18615       \@printgloss@setsort
18616       \setkeys{printgloss}{#1}%
18617       \@printgloss@checkexists{\@glo@type}%
18618       {%
18619         \ifx\glossarytitle\org@glossarytitle
18620           \else
18621             \expandafter\let\csname @glotype@\@glo@type @title\endcsname

```

```

18622             \glossarytitle
18623     \fi
18624     \let\currentglossary\@glo@type
18625     \let\org@glossaryentrynumbers@glossaryentrynumbers
18626     \let\glsnonextpages@glsnonextpages
18627     \let\glsnextpages@glsnextpages
18628     \let\nopostdesc@nopostdesc
18629     \gls@dotoc@title
18630     \@glossarystyle
18631     \let\gls@org@glossaryentryfield@glossentry
18632     \let\gls@org@glossarysubentryfield@subglossentry
18633     \renewcommand{\glossentry}[1]{%
18634         \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
18635         \gls@org@glossaryentryfield{##1}%
18636     }%
18637     \renewcommand{\subglossentry}[2]{%
18638         \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
18639         \gls@org@glossarysubentryfield{##1}{##2}%
18640     }%
18641     \@gls@preglossaryhook
18642     #2%
18643 }%
18644 \egroup
18645 \global\let@glossaryentrynumbers\@org@glossaryentrynumbers
18646 \global\let\warn@noprintglossary\relax
18647 }
18648 \newcommand{\@print@glossary}{%
18649     \makeatletter
18650     \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
18651     \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
18652     {}%
18653     {\null}%
18654     \ifglxindy
18655         \ifcsundef{@xdy@\@glo@type @language}%
18656         {%
18657             \edef\@do@auxoutstuff{%
18658                 \noexpand\AtEndDocument{%
18659                     \noexpand\immediate\noexpand\write\@auxout{%
18660                         \string\providecommand\string\@xdylanguage[2]{}}%
18661                     \noexpand\immediate\noexpand\write\@auxout{%
18662                         \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
18663                 }%
18664             }%
18665         }%
18666     {%
18667         \edef\@do@auxoutstuff{%
18668             \noexpand\AtEndDocument{%
18669                 \noexpand\immediate\noexpand\write\@auxout{%
18670                     \string\providecommand\string\@xdylanguage[2]{}}%

```

```

18671         \noexpand\immediate\noexpand\write\@auxout{%
18672             \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
18673                 @language\endcsname}}}%
18674     }%
18675 }%
18676 }%
18677 \do@auxoutstuff
18678 \edef\do@auxoutstuff{%
18679     \noexpand\AtEndDocument{%
18680         \noexpand\immediate\noexpand\write\@auxout{%
18681             \string\providecommand\string\@gls@codepage[2]{}}%
18682         \noexpand\immediate\noexpand\write\@auxout{%
18683             \string\@gls@codepage{\@glo@type}{\@gls@codepage}}}%
18684     }%
18685 }%
18686 \do@auxoutstuff
18687 \fi
18688 \renewcommand*{\@warn@nomakeglossaries}{%
18689     \GlossariesWarningNoLine{\string\makeglossaries\space
18690         hasn't been used,^^Jthe glossaries will not be updated}%
18691 }%
18692 }
18693 \newcommand*{\@glo@sortentries}[2]{%
18694     \glosortentrieswarning
18695     \def\@glo@sortinglist{}%
18696     \def\@glo@sortinghandler{#1}%
18697     \edef\@glo@type{#2}%
18698     \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
18699     \csdef{\@glsref@#2}{}%
18700     \@for\@this@label:=\@glo@sortinglist\do{%
18701         \xifinlistcs{\@this@label}{\@glsref@#2}%
18702         {}%
18703         {%
18704             \listcsxadd{\@glsref@#2}{\@this@label}%
18705         }%
18706         \ifcsdef{\@glo@sortingchildren@\@this@label}%
18707         {%
18708             \@glo@addchildren{#2}{\@this@label}%
18709         }%
18710     }%
18711 }%
18712 }
18713 \newcommand*{\@glo@addchildren}[2]{%
18714     \bgroup
18715     \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
18716     \@for\@this@childlabel:=\@glo@childlist\do
18717     {%
18718         \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
18719         {}%

```



```

18720      {%
18721      \listcsxadd{@glsref{#1}}{\@this@childlabel}%
18722      }%
18723      \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
18724      {%
18725      \@glo@addchildren{#1}{\@this@childlabel}%
18726      }%
18727      {%
18728      }%
18729      }%
18730 \egroup
18731 }
18732 \newcommand*{\@glo@do@sortentries}[1]{%
18733 \ifglshasparent{#1}%
18734 {%
18735 \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
18736 \ifcsundef{@glo@sortingchildren@\@glo@parent}%
18737 {%
18738 \csdef{@glo@sortingchildren@\@glo@parent}{}%
18739 }%
18740 {}%
18741 \expandafter\@glo@sortedinsert
18742 \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
18743 \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
18744 {%
18745 }%
18746 {%
18747 \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
18748 }%
18749 }%
18750 {%
18751 \@glo@sortedinsert{\@glo@sortinglist}{#1}%
18752 }%
18753 }
18754 \newcommand*{\@glo@sortedinsert}[2]{%
18755 \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
18756 }%
18757 \newcommand*{\@glo@sorthandler@word}[2]{%
18758 \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
18759 \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
18760 \edef\@glo@do@compare{%
18761 \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
18762 {\expandonce\@gls@sort@B}%
18763 {\expandonce\@gls@sort@A}%
18764 }%
18765 \@glo@do@compare
18766 }
18767 \newcommand*{\@glo@sorthandler@letter}[2]{%
18768 \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%

```

```

18769 \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18770 \edef\glo@do@compare{%
18771   \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
18772   {\expandonce\@gls@sort@B}%
18773   {\expandonce\@gls@sort@A}%
18774 }%
18775 \glo@do@compare
18776 }
18777 \newcommand*{\@glo@sorthandler@case}[2]{%
18778   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
18779   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18780   \edef\glo@do@compare{%
18781     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
18782     {\expandonce\@gls@sort@B}%
18783     {\expandonce\@gls@sort@A}%
18784   }%
18785   \glo@do@compare
18786 }
18787 \newcommand*{\@glo@sorthandler@nocase}[2]{%
18788   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
18789   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18790   \edef\glo@do@compare{%
18791     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
18792     {\expandonce\@gls@sort@B}%
18793     {\expandonce\@gls@sort@A}%
18794   }%
18795   \glo@do@compare
18796 }
18797 \newcommand*{\@glo@sortmacro@word}[1]{%
18798   \ifdefstring{\@glo@default@sorttype}{standard}%
18799   {%
18800     \@glo@sortentries{\@glo@sorthandler@word}{#1}%
18801   }%
18802   {%
18803     \PackageError{glossaries}{Conflicting sort options:^^J
18804       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18805       \string\printnoidxglossary[sort=word]}{}}%
18806   }%
18807 }
18808 \newcommand*{\@glo@sortmacro@letter}[1]{%
18809   \ifdefstring{\@glo@default@sorttype}{standard}%
18810   {%
18811     \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
18812   }%
18813   {%
18814     \PackageError{glossaries}{Conflicting sort options:^^J
18815       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18816       \string\printnoidxglossary[sort=letter]}{}}%
18817   }%

```

```

18818 }
18819 \newcommand*{\@glo@sortmacro@standard}[1]{%
18820   \ifdefstring{\@glo@default@sorttype}{standard}%
18821   {%
18822     \ifcsdef{\@glo@sorthandler@\glsorder}%
18823     {%
18824       \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
18825     }%
18826     {%
18827       \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}%
18828     }%
18829   }%
18830   {%
18831     \PackageError{glossaries}{Conflicting sort options:^^J
18832       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18833       \string\printnoidxglossary[sort=standard]}{}%
18834   }%
18835 }
18836 \newcommand*{\@glo@sortmacro@case}[1]{%
18837   \ifdefstring{\@glo@default@sorttype}{standard}%
18838   {%
18839     \@glo@sortentries{\@glo@sorthandler@case}{#1}%
18840   }%
18841   {%
18842     \PackageError{glossaries}{Conflicting sort options:^^J
18843       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18844       \string\printnoidxglossary[sort=case]}{}%
18845   }%
18846 }
18847 \newcommand*{\@glo@sortmacro@nocase}[1]{%
18848   \ifdefstring{\@glo@default@sorttype}{standard}%
18849   {%
18850     \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
18851   }%
18852   {%
18853     \PackageError{glossaries}{Conflicting sort options:^^J
18854       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18855       \string\printnoidxglossary[sort=nocase]}{}%
18856   }%
18857 }
18858 \newcommand*{\@glo@sortmacro@def}[1]{%
18859   \def\@glo@sortinglist{}%
18860   \for\glsentries[#1]{\@gls@thislabel}%
18861   {%
18862     \xifinlistcs{\@gls@thislabel}{\@gls@ref@#1}%
18863     {%
18864       \listadd{\@glo@sortinglist}{\@gls@thislabel}%
18865     }%
18866     {%

```

```

18867 }%
18868 }%
18869 \cslet{@glsref@#1}{\@glo@sortinglist}%
18870 }
18871 \newcommand*{\@glo@sortmacro@def@do}[1]{%
18872 \ifinlistcs{#1}{@glsref@\@glo@type}%
18873 {}%
18874 {%
18875 \listcsadd{@glsref@\@glo@type}{#1}%
18876 }%
18877 \ifcsdef{@glo@sortingchildren@#1}%
18878 {%
18879 \@glo@addchildren{\@glo@type}{#1}%
18880 }%
18881 {}%
18882 }
18883 \newcommand*{\@glo@sortmacro@use}[1]{%
18884 \newcommand*{\@print@noidx@glossary}{%
18885 \ifcsdef{@glsref@\@glo@type}%
18886 {%
18887 \ifcsdef{@glo@sortmacro@\@glo@sorttype}%
18888 {%
18889 \csuse{@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
18890 }%
18891 {%
18892 \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
18893 }%
18894 \glossarysection[\glossarytoctitle]{\glossarytitle}%
18895 \glossarypreamble
18896 \def\@gls@currentlettergroup{}%
18897 \begin{theglossary}%
18898 \glossaryheader
18899 \glsresetentrylist
18900 \forlistcsloop{\@gls@noidx@do}{@glsref@\@glo@type}%
18901 \end{theglossary}%
18902 \glossarypostamble
18903 }%
18904 {%
18905 \@gls@noref@warn{\@glo@type}%
18906 }%
18907 }
18908 \def\glo@grabfirst#1#2\@nil{%
18909 \def\@gls@firsttok{#1}%
18910 \ifdefempty\@gls@firsttok
18911 {%
18912 \def\@glo@thislettergrp{0}%
18913 }%
18914 {%
18915 \@onelevel@sanitize\@gls@firsttok

```

```

18916 \expandafter\@glo@grabfirst\@gls@firsttok{}\}\@nil
18917 }%
18918 }
18919 \def\@glo@grabfirst#1#2\@nil{%
18920 \ifdefempty\@glo@thislettergrp
18921 {%
18922 \def\@glo@thislettergrp{glssymbols}%
18923 }%
18924 {%
18925 \count@=\uccode'#1\relax
18926 \ifnum\count@=0\relax
18927 \def\@glo@thislettergrp{glssymbols}%
18928 \else
18929 \ifdefstring\@glo@sorttype{case}%
18930 {%
18931 \count@='#1\relax
18932 }%
18933 {%
18934 }%
18935 \edef\@glo@thislettergrp{\the\count@}%
18936 \fi
18937 }%
18938 }
18939 \newcommand{\@gls@noidx@do}[1]{%
18940 \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
18941 \ifglshasparent{#1}%
18942 {%
18943 \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
18944 \ifdefvoid{\@gls@loclist}
18945 {%
18946 \subglossentry{\gls@level}{#1}{}%
18947 }%
18948 {%
18949 \subglossentry{\gls@level}{#1}%
18950 }%
18951 \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
18952 }%
18953 }%
18954 }%
18955 {%
18956 \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
18957 \expandafter\glo@grabfirst\@gls@sort{}\}\@nil
18958 \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
18959 {}%
18960 {%
18961 \ifdefempty{\@gls@currentlettergroup}{}%
18962 {%
18963 \global\let\@glo@thislettergrp\@glo@thislettergrp
18964 \glsgroupskip

```

```

18965     }%
18966     \glsgroupheading{\@glo@thislettergrp}%
18967 }%
18968 \global\let\@gls@currentlettergroup\@glo@thislettergrp
18969 \ifdefvoid{\@gls@loclist}
18970 {%
18971     \glossentry{#1}{}%
18972 }%
18973 {%
18974     \glossentry{#1}%
18975     {%
18976         \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
18977     }%
18978 }%
18979 }%
18980 }
18981 \newcommand*{\glsnoidxloclist}[1]{%
18982     \def\@gls@noidxloclist@sep{}%
18983     \def\@gls@noidxloclist@prev{}%
18984     \forlistloop{\glsnoidxloclisthandler}{#1}%
18985 }
18986 \newcommand*{\glsnoidxloclisthandler}[1]{%
18987     \ifdefstring{\@gls@noidxloclist@prev}{#1}%
18988     {%
18989     }%
18990     {%
18991         \@gls@noidxloclist@sep
18992         #1%
18993         \def\@gls@noidxloclist@sep{\delimN}%
18994         \def\@gls@noidxloclist@prev{#1}%
18995     }%
18996 }
18997 \newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
18998     \ifdefstring{\@gls@noidxloclist@prev}{#1}%
18999     {%
19000     }%
19001     {%
19002         \@gls@noidxloclist@sep
19003         \@gls@noidxloclist@prev
19004         \def\@gls@noidxloclist@prev{#1}%
19005     }%
19006 }
19007 \newcommand*{\glsnoidxdisplayloc}[4]{%
19008     \setentrycounter{#1}{#2}%
19009     \csuse{#3}{#4}%
19010 }
19011 \newcommand*{\@gls@reference}[3]{%
19012     \glsdoifexistsorwarn{#2}%
19013     {%

```

```

19014 \ifcsundef{@glsref@#1}{\csgdef{@glsref@#1}{}}{}%
19015 \ifinlistcs{#2}{@glsref@#1}%
19016 {}%
19017 {\listcsgadd{@glsref@#1}{#2}}%
19018 \ifcsundef{glo@glstdetoklabel{#2}@loclist}%
19019 {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%
19020 {}%
19021 \listcsgadd{glo@glstdetoklabel{#2}@loclist}{#3}%
19022 }%
19023 }
19024 \define@key{printgloss}{type}{\def@glo@type{#1}}
19025 \define@key{printgloss}{title}{%
19026 \def@glossarytitle{#1}%
19027 \let@gls@dotoc@title\relax
19028 }
19029 \define@key{printgloss}{toctitle}{%
19030 \def@glossarytoctitle{#1}%
19031 \let@gls@dotoc@title\relax
19032 }
19033 \define@key{printgloss}{style}{%
19034 \ifcsundef{@glsstyle@#1}%
19035 {%
19036 \PackageError{glossaries}%
19037 {Glossary style ‘#1’ undefined}{}%
19038 }%
19039 {%
19040 \def@glossarystyle{\setglossentrycompatibility
19041 \csname @glsstyle@#1\endcsname}%
19042 }%
19043 }
19044 \define@choicekey{printgloss}{numberedsection}%
19045 [\gls@numberedsection@val\gls@numberedsection@nr]%
19046 {false,nolabel,autolabel,nameref}[nolabel]%
19047 {%
19048 \ifcase\gls@numberedsection@nr\relax
19049 \renewcommand*{\@@glossarysecstar}{*}%
19050 \renewcommand*{\@@glossaryseclabel}{}%
19051 \or
19052 \renewcommand*{\@@glossarysecstar}{}%
19053 \renewcommand*{\@@glossaryseclabel}{}%
19054 \or
19055 \renewcommand*{\@@glossarysecstar}{*}%
19056 \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix@glo@type}}%
19057 \or
19058 \renewcommand*{\@@glossarysecstar}{*}%
19059 \renewcommand*{\@@glossaryseclabel}{%
19060 \protected@edef\@currentlabelname{\glossarytoctitle}%
19061 \label{\glsautoprefix@glo@type}}%
19062 \fi

```

```

19063 }
19064 \define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
19065   \csuse{glsnogroupskip#1}%
19066 }
19067 \define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
19068   \csuse{glsnopostdot#1}%
19069 }
19070 \newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
19071 \define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
19072   \csuse{glsentrycounter#1}%
19073   \@gls@define@glossaryentrycounter
19074 }
19075 \define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
19076   \csuse{glssubentrycounter#1}%
19077   \@gls@define@glossarysubentrycounter
19078 }
19079 \define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
19080 \ifglsnonumberlist
19081   \def\glossaryentrynumbers##1{%
19082 \else
19083   \def\glossaryentrynumbers##1{##1}%
19084 \fi}
19085 \define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
19086 \newcommand*{\@glo@no@assign@sortkey}[1]{%
19087   \PackageError{glossaries}{‘sort’ key not permitted with
19088     \string\printglossary}%
19089   {The ‘sort’ key may only be used with \string\printnoidxglossary}%
19090 }
19091 \newcommand*{\@glo@assign@sortkey}[1]{%
19092   \def\@glo@sorttype{#1}%
19093 }
19094 \newcommand*{\@glsnonextpages}{%
19095   \gdef\glossaryentrynumbers##1{%
19096     \glsresetentrylist
19097   }%
19098 }
19099 \newcommand*{\@glsnextpages}{%
19100   \gdef\glossaryentrynumbers##1{%
19101     ##1\glsresetentrylist}}
19102 \newcommand*{\glsresetentrylist}{%
19103   \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
19104 \newcommand*{\glsnonextpages}{}
19105 \newcommand*{\glsnextpages}{}
19106 \@gls@define@glossaryentrycounter
19107 \@gls@define@glossarysubentrycounter
19108 \newcommand*{\glsresetsubentrycounter}{%
19109   \ifglssubentrycounter
19110     \setcounter{glossarysubentry}{0}%
19111   \fi

```



```

19112 }
19113 \newcommand*{\glsresetentrycounter}{%
19114   \ifglssentrycounter
19115     \setcounter{glossaryentry}{0}%
19116   \fi
19117 }
19118 \newcommand*{\glsstepentry}[1]{%
19119   \ifglssentrycounter
19120     \refstepcounter{glossaryentry}%
19121     \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19122   \fi
19123 }
19124 \newcommand*{\glsstepsubentry}[1]{%
19125   \ifglsssubentrycounter
19126     \edef\currentglsssubentry{\glsdetoklabel{#1}}%
19127     \refstepcounter{glossarysubentry}%
19128     \label{\GlsEntryCounterLabelPrefix\currentglsssubentry}%
19129   \fi
19130 }
19131 \newcommand*{\glsrefentry}[1]{%
19132   \ifglssentrycounter
19133     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19134   \else
19135     \ifglsssubentrycounter
19136       \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19137     \else
19138       \gls{#1}%
19139     \fi
19140   \fi
19141 }
19142 \newcommand*{\glsentrycounterlabel}{%
19143   \ifglssentrycounter
19144     \theglossaryentry.\space
19145   \fi
19146 }
19147 \newcommand*{\glsssubentrycounterlabel}{%
19148   \ifglsssubentrycounter
19149     \theglossarysubentry)\space
19150   \fi
19151 }
19152 \newcommand*{\glsentryitem}[1]{%
19153   \ifglssentrycounter
19154     \glsstepentry{#1}\glsentrycounterlabel
19155   \else
19156     \glsresetsubentrycounter
19157   \fi
19158 }
19159 \newcommand*{\glssubentryitem}[1]{%
19160   \ifglsssubentrycounter

```

```

19161 \glsstepsubentry{#1}\glssubentrycounterlabel
19162 \fi
19163 }
19164 \ifcsundef{theglossary}%
19165 {%
19166 \newenvironment{theglossary}{}{}%
19167 }%
19168 {%
19169 \@gls@warnontheglossdefined
19170 \renewenvironment{theglossary}{}{}%
19171 }
19172 \newcommand*{\glossaryheader}{}
19173 \newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
19174 \providecommand*{\compatibleglossentry}[2]{%
19175 \toks@{#2}%
19176 \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
19177 {\noexpand\glsnamefont
19178 {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
19179 {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
19180 {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
19181 {\the\toks@}%
19182 }%
19183 \@do@glossentry
19184 }
19185 \newcommand*{\glossentryname}[1]{%
19186 \glsdoifexistsorwarn{#1}%
19187 {%
19188 \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
19189 \expandafter\glsnamefont\expandafter{\glo@name}%
19190 }%
19191 }
19192 \newcommand*{\Glossentryname}[1]{%
19193 \glsdoifexistsorwarn{#1}%
19194 {%
19195 \glsnamefont{\Glsentryname{#1}}%
19196 }%
19197 }
19198 \newcommand*{\glossentrydesc}[1]{%
19199 \glsdoifexistsorwarn{#1}%
19200 {%
19201 \glsentrydesc{#1}%
19202 }%
19203 }
19204 \newcommand*{\Glossentrydesc}[1]{%
19205 \glsdoifexistsorwarn{#1}%
19206 {%
19207 \Glsentrydesc{#1}%
19208 }%
19209 }

```

```

19210 \newcommand*{\glossentrysymbol}[1]{%
19211   \glsdoifexistsorwarn{#1}%
19212   {%
19213     \glsentrysymbol{#1}%
19214   }%
19215 }
19216 \newcommand*{\Glossentrysymbol}[1]{%
19217   \glsdoifexistsorwarn{#1}%
19218   {%
19219     \Glsentrysymbol{#1}%
19220   }%
19221 }
19222 \providecommand*{\compatiblesubglossentry}[3]{%
19223   \toks@{#3}%
19224   \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
19225     {#2}%
19226     {\noexpand\glsnamefont
19227       {\expandafter\expandonce\csname glo@#2@name\endcsname}}}%
19228     {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
19229     {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
19230     {\the\toks@}%
19231   }%
19232   \@do@subglossentry
19233 }
19234 \newcommand*{\setglossentrycompatibility}{%
19235   \let\glossentry\compatibleglossentry
19236   \let\subglossentry\compatiblesubglossentry
19237 }
19238 \setglossentrycompatibility
19239 \newcommand{\glossaryentryfield}[5]{%
19240   \GlossariesWarning
19241   {Deprecated use of \string\glossaryentryfield.^^J
19242     I recommend you change to \string\glossentry.^^J
19243     If you've just upgraded, try removing your gls auxiliary
19244     files^^J and recompile}%
19245   \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
19246 \newcommand*{\glossarysubentryfield}[6]{%
19247   \GlossariesWarning
19248   {Deprecated use of \string\glossarysubentryfield.^^J
19249     I recommend you change to \string\subglossentry.^^J
19250     If you've just upgraded, try removing your gls auxiliary
19251     files^^J and recompile}%
19252   \glstarget{#2}{\strut}#4. #6\par}
19253 \newcommand*{\glsgroupskip}{}
19254 \newcommand*{\glsgroupheading}[1]{}
19255 \newcommand*{\glsgetgrouptitle}[1]{%
19256   \@gls@getgrouptitle{#1}{\@gls@grptitle}%
19257   \@gls@grptitle
19258 }

```

```

19259 \newcommand*{\@gls@getgrouptitle}[2]{%
19260   \dtl@ifsingle{#1}%
19261   {%
19262     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19263   }%
19264   {%
19265     \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
19266                 or test{\ifstrequal{#1}{glsnumbers}}}%
19267     {%
19268       \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19269     }%
19270     {%
19271       \def#2{#1}%
19272     }%
19273   }%
19274 }
19275 \newcommand*{\@gls@noidx@getgrouptitle}[2]{%
19276   \DTLifint{#1}%
19277   {\edef#2{\char#1\relax}}%
19278   {%
19279     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19280   }%
19281 }
19282 \newcommand*{\glsgetgrouplabel}[1]{%
19283   \ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
19284   \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
19285 \newcommand*{\setentrycounter}[2][1]{%
19286   \def\@glo@counterprefix{#1}%
19287   \ifx\@glo@counterprefix\empty
19288     \def\@glo@counterprefix{.}%
19289   \else
19290     \def\@glo@counterprefix{.#1.}%
19291   \fi
19292   \def\glsentrycounter{#2}%
19293 }
19294 \newcommand*{\setglossarystyle}[1]{%
19295   \ifcsundef{@glsstyle@#1}%
19296   {%
19297     \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
19298   }%
19299   {%
19300     \csname @glsstyle@#1\endcsname
19301   }%
19302   \ifx\@glossary@default@style\relax
19303     \protected@edef\@glossary@default@style{#1}%
19304   \fi
19305 }
19306 \newcommand*{\glossarystyle}[1]{%
19307   \ifcsundef{@glsstyle@#1}%

```

```

19308 {%
19309   \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
19310 }%
19311 {%
19312   \GlossariesWarning
19313   {Deprecated command \string\glossarystyle.^~J
19314     I recommend you switch to \string\setglossarystyle\space unless
19315     you want to maintain backward compatibility}%
19316   \setglossentrycompatibility
19317   \csname @glsstyle@#1\endcsname
19318   \ifcsdef{@glscompstyle@#1}%
19319     {\setglossentrycompatibility\csuse{@glscompstyle@#1}}%
19320   }%
19321 }%
19322 \ifx\@glossary@default@style\relax
19323   \protected@edef\@glossary@default@style{#1}%
19324 \fi
19325 }
19326 \newcommand{\newglossarystyle}[2]{%
19327   \ifcsundef{@glsstyle@#1}%
19328   {%
19329     \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
19330   }%
19331   {%
19332     \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
19333   }%
19334 }
19335 \newcommand{\renewglossarystyle}[2]{%
19336   \ifcsundef{@glsstyle@#1}%
19337   {%
19338     \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
19339   }%
19340   {%
19341     \csdef{@glsstyle@#1}{#2}%
19342   }%
19343 }
19344 \newcommand*{\glsnamefont}[1]{#1}
19345 \ifcsundef{hyperlink}%
19346 {%
19347   \def\glshypernumber#1{#1}%
19348 }%
19349 {%
19350   \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
19351 }
19352 \def\@glshypernumber#1\nohyperpage#2#3\@nil{%
19353   \ifx\#1\%
19354     \else
19355       \@delimR#1\delimR\delimR\%
19356   \fi

```

```

19357 \ifx\\#2\\%
19358 \else
19359 #2%
19360 \fi
19361 \ifx\\#3\\%
19362 \else
19363 \@glshypernumber#3\@nil
19364 \fi
19365 }
19366 \def\@delimR#1\delimR #2\delimR #3\\{%
19367 \ifx\\#2\\%
19368 \@delimN{#1}%
19369 \else
19370 \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
19371 \fi}
19372 \def\@delimN#1{\@delimN#1\delimN \delimN\\}
19373 \def\@delimN#1\delimN #2\delimN#3\\{%
19374 \ifx\\#3\\%
19375 \@gls@numberlink{#1}%
19376 \else
19377 \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
19378 \fi
19379 }
19380 \def\@gls@numberlink#1{%
19381 \begingroup
19382 \toks@={}%
19383 \@gls@removespaces#1 \@nil
19384 \endgroup}
19385 \def\@gls@removespaces#1 #2\@nil{%
19386 \toks@=\expandafter{\the\toks@#1}%
19387 \ifx\\#2\\%
19388 \edef\x{\the\toks@}%
19389 \ifx\x\empty
19390 \else
19391 \hyperlink{\glsentrycounter\@glo@counterprefix\the\toks@}%
19392 {\the\toks@}%
19393 \fi
19394 \else
19395 \@gls@ReturnAfterFi{%
19396 \@gls@removespaces#2\@nil
19397 }%
19398 \fi
19399 }
19400 \long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
19401 \newcommand*{\hyperrm}[1]{\textrm{\glshypernumber{#1}}}
19402 \newcommand*{\hypersf}[1]{\textsf{\glshypernumber{#1}}}
19403 \newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}
19404 \newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
19405 \newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}

```

```

19406 \newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}
19407 \newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
19408 \newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
19409 \newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
19410 \newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}
19411 \newcommand{\oldacronym}[4][\gls@label]{%
19412   \def\gls@label{#2}%
19413   \newacronym[#4]{#1}{#2}{#3}%
19414   \ifcsundef{xspace}%
19415   {%
19416     \expandafter\edef\csname#1\endcsname{%
19417       \noexpand@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
19418   }%
19419 }%
19420 {%
19421   \expandafter\edef\csname#1\endcsname{%
19422     \noexpand@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
19423       \noexpand\gls{#1}\noexpand\xspace}%
19424   }%
19425 }%
19426 }
19427 \newcommand{\newacronym}[4][{}]{
19428 \newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
19429 \newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
19430 \newcommand*{\glsshortkey}{short}
19431 \newcommand*{\glsshortpluralkey}{shortplural}
19432 \newcommand*{\glslongkey}{long}
19433 \newcommand*{\glslongpluralkey}{longplural}
19434 \newrobustcmd*{\acrfull}{\@gls@hyp@opt\@ns@acrfull}
19435 \newcommand*\ns@acrfull[2][{}]{%
19436   \new@ifnextchar[{\@acrfull{#1}{#2}}{%
19437     {\@acrfull{#1}{#2}}[{}]}%
19438 }
19439 \def\@acrfull#1#2[#3]{%
19440   \acrfullfmt{#1}{#2}{#3}%
19441 }
19442 \newcommand*{\acrfullfmt}[3]{%
19443   \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
19444 }
19445 \newcommand{\acrlinkfullformat}[5]{%
19446   \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[{}]}%
19447 }
19448 \newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
19449 \newrobustcmd{\glsspace}{\space}
19450 \newrobustcmd*{\Acrfull}{\@gls@hyp@opt\@ns@Acrfull}
19451 \newcommand*\ns@Acrfull[2][{}]{%
19452   \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
19453     {\@Acrfull{#1}{#2}}[{}]}%
19454 }

```

```

19455 \def\@Acrfull#1#2[#3]{%
19456   \Acrfullfmt{#1}{#2}{#3}%
19457 }
19458 \newcommand*\@Acrfullfmt}[3]{%
19459   \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
19460 }
19461 \newrobustcmd*\@ACRfull{\@gls@hyp@opt\@ns@ACRfull}
19462 \newcommand*\@ns@ACRfull[2][{}]{%
19463   \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
19464     {\@ACRfull{#1}{#2}[]}%
19465 }
19466 \def\@ACRfull#1#2[#3]{%
19467   \ACRfullfmt{#1}{#2}{#3}%
19468 }
19469 \newcommand*\@ACRfullfmt}[3]{%
19470   \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
19471 }
19472 \newrobustcmd*\@acrfullpl{\@gls@hyp@opt\@ns@acrfullpl}
19473 \newcommand*\@ns@acrfullpl[2][{}]{%
19474   \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
19475     {\@acrfullpl{#1}{#2}[]}%
19476 }
19477 \def\@acrfullpl#1#2[#3]{%
19478   \acrfullplfmt{#1}{#2}{#3}%
19479 }
19480 \newcommand*\@acrfullplfmt}[3]{%
19481   \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
19482 }
19483 \newrobustcmd*\@Acrfullpl{\@gls@hyp@opt\@ns@Acrfullpl}
19484 \newcommand*\@ns@Acrfullpl[2][{}]{%
19485   \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
19486     {\@Acrfullpl{#1}{#2}[]}%
19487 }
19488 \def\@Acrfullpl#1#2[#3]{%
19489   \Acrfullplfmt{#1}{#2}{#3}%
19490 }
19491 \newcommand*\@Acrfullplfmt}[3]{%
19492   \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
19493 }
19494 \newrobustcmd*\@ACRfullpl{\@gls@hyp@opt\@ns@ACRfullpl}
19495 \newcommand*\@ns@ACRfullpl[2][{}]{%
19496   \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
19497     {\@ACRfullpl{#1}{#2}[]}%
19498 }
19499 \def\@ACRfullpl#1#2[#3]{%
19500   \ACRfullplfmt{#1}{#2}{#3}%
19501 }
19502 \newcommand*\@ACRfullplfmt}[3]{%
19503   \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%

```



```

19504 }
19505 \newcommand{\acronymfont}[1]{#1}
19506 \newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
19507 \newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
19508 \newtoks\glskeylisttok
19509 \newtoks\glslabeltok
19510 \newtoks\glsshorttok
19511 \newtoks\glslongtok
19512 \newcommand*{\newacronymhook}{}
19513 \newcommand*{\SetGenericNewAcronym}{%
19514   \let\@Gls@entryname\@Gls@acrenryname
19515   \renewcommand{\newacronym}[4][]{%
19516     \ifdefempty{\@glsacronymlists}%
19517     {%
19518       \def\@glo@type{\acronymtype}%
19519       \setkeys{glossentry}{##1}%
19520       \DeclareAcronymList{\@glo@type}%
19521     }%
19522   }%
19523   \glskeylisttok{##1}%
19524   \glslabeltok{##2}%
19525   \glsshorttok{##3}%
19526   \glslongtok{##4}%
19527   \newacronymhook
19528   \protected@edef\@do@newglossaryentry{%
19529     \noexpand\newglossaryentry{\the\glslabeltok}%
19530     {%
19531       type=\acronymtype,%
19532       name={\expandonce{\acronymentry{##2}}},%
19533       sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
19534       text={\the\glsshorttok},%
19535       short={\the\glsshorttok},%
19536       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
19537       long={\the\glslongtok},%
19538       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
19539       \GenericAcronymFields,%
19540       \the\glskeylisttok
19541     }%
19542   }%
19543   \@do@newglossaryentry
19544 }%
19545 \renewcommand*{\acrfullfmt}[3]{%
19546   \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
19547 \renewcommand*{\Acrfullfmt}[3]{%
19548   \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
19549 \renewcommand*{\ACRfullfmt}[3]{%
19550   \glslink[##1]{##2}{%
19551     \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
19552   \renewcommand*{\acrfullplfmt}[3]{%

```

```

19553 \glslink{##1}{##2}{\genplacrfullformat{##2}{##3}}}%
19554 \renewcommand*{\Acrfullplfmt}[3]{%
19555 \glslink{##1}{##2}{\Genplacrfullformat{##2}{##3}}}%
19556 \renewcommand*{\ACRfullplfmt}[3]{%
19557 \glslink{##1}{##2}{%
19558 \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}}}%
19559 \renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
19560 \renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
19561 \renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
19562 \renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
19563 }
19564 \newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
19565 \newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
19566 \newcommand*{\acronymsort}[2]{#1}
19567 \newcommand*{\setacronymstyle}[1]{%
19568 \ifcsundef{@glsacr@dispstyle@#1}
19569 {%
19570 \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
19571 }%
19572 {%
19573 \ifdefempty{\@glsacronymlists}%
19574 {%
19575 \DeclareAcronymList{\acronymtype}%
19576 }%
19577 }%
19578 \SetGenericNewAcronym
19579 \GlsUseAcrStyleDefs{#1}%
19580 \@for\@gls@type:=\@glsacronymlists\do{%
19581 \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDisplayStyle{#1}}%
19582 }%
19583 }%
19584 }
19585 \newcommand*{\newacronymstyle}[3]{%
19586 \ifcsdef{@glsacr@dispstyle@#1}%
19587 {%
19588 \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
19589 }%
19590 {%
19591 \csdef{@glsacr@dispstyle@#1}{#2}%
19592 \csdef{@glsacr@styledefs@#1}{#3}%
19593 }%
19594 }
19595 \newcommand*{\renewacronymstyle}[3]{%
19596 \ifcsdef{@glsacr@dispstyle@#1}%
19597 {%
19598 \csdef{@glsacr@dispstyle@#1}{#2}%
19599 \csdef{@glsacr@styledefs@#1}{#3}%
19600 }%
19601 {%

```

```

19602 \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
19603 }%
19604 }
19605 \newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
19606 \newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
19607 \newacronymstyle{long-short}%
19608 {%
19609 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19610 }%
19611 {%
19612 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19613 \renewcommand*{\genacrfullformat}[2]{%
19614 \glsentrylong{##1}##2\space
19615 (\protect\firstacronymfont{\glsentryshort{##1}})%
19616 }%
19617 \renewcommand*{\Genacrfullformat}[2]{%
19618 \Glsentrylong{##1}##2\space
19619 (\protect\firstacronymfont{\glsentryshort{##1}})%
19620 }%
19621 \renewcommand*{\genplacrfullformat}[2]{%
19622 \glsentrylongpl{##1}##2\space
19623 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19624 }%
19625 \renewcommand*{\Genplacrfullformat}[2]{%
19626 \Glsentrylongpl{##1}##2\space
19627 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19628 }%
19629 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19630 \renewcommand*{\acronymsort}[2]{##1}%
19631 \renewcommand*{\acronymfont}[1]{##1}%
19632 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19633 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19634 }
19635 \newacronymstyle{long-sp-short}%
19636 {%
19637 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19638 }%
19639 {%
19640 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19641 \renewcommand*{\genacrfullformat}[2]{%
19642 \glsentrylong{##1}##2\glsacspace{##1}%
19643 (\protect\firstacronymfont{\glsentryshort{##1}})%
19644 }%
19645 \renewcommand*{\Genacrfullformat}[2]{%
19646 \Glsentrylong{##1}##2\glsacspace{##1}%
19647 (\protect\firstacronymfont{\glsentryshort{##1}})%
19648 }%
19649 \renewcommand*{\genplacrfullformat}[2]{%
19650 \glsentrylongpl{##1}##2\glsacspace{##1}%

```

```

19651 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19652 }%
19653 \renewcommand*{\Genplacrfullformat}[2]{%
19654 \Glsentrylongpl{##1}##2\glsacspace{##1}%
19655 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19656 }%
19657 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19658 \renewcommand*{\acronymsort}[2]{##1}%
19659 \renewcommand*{\acronymfont}[1]{##1}%
19660 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19661 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19662 }
19663 \newcommand*{\glsacspace}[1]{%
19664 \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{#1}})}%
19665 \ifdim\dimen@<3em~\else\spacefi
19666 }
19667 \newacronymstyle{short-long}%
19668 {%
19669 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19670 }%
19671 {%
19672 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19673 \renewcommand*{\genacrfullformat}[2]{%
19674 \protect\firstacronymfont{\glsentryshort{##1}}##2\space
19675 (\glsentrylong{##1})%
19676 }%
19677 \renewcommand*{\Genacrfullformat}[2]{%
19678 \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
19679 (\glsentrylong{##1})%
19680 }%
19681 \renewcommand*{\genplacrfullformat}[2]{%
19682 \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
19683 (\glsentrylongpl{##1})%
19684 }%
19685 \renewcommand*{\Genplacrfullformat}[2]{%
19686 \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
19687 (\glsentrylongpl{##1})%
19688 }%
19689 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19690 \renewcommand*{\acronymsort}[2]{##1}%
19691 \renewcommand*{\acronymfont}[1]{##1}%
19692 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19693 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19694 }
19695 \newacronymstyle{long-sc-short}%
19696 {%
19697 \GlsUseAcrEntryDispStyle{long-short}%
19698 }%
19699 {%

```

```

19700 \GlsUseAcrStyleDefs{long-short}%
19701 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19702 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19703 }
19704 \newacronymstyle{long-sm-short}%
19705 {%
19706 \GlsUseAcrEntryDispStyle{long-short}%
19707 }%
19708 {%
19709 \GlsUseAcrStyleDefs{long-short}%
19710 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19711 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19712 }
19713 \newacronymstyle{sc-short-long}%
19714 {%
19715 \GlsUseAcrEntryDispStyle{short-long}%
19716 }%
19717 {%
19718 \GlsUseAcrStyleDefs{short-long}%
19719 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19720 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19721 }
19722 \newacronymstyle{sm-short-long}%
19723 {%
19724 \GlsUseAcrEntryDispStyle{short-long}%
19725 }%
19726 {%
19727 \GlsUseAcrStyleDefs{short-long}%
19728 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19729 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19730 }
19731 \newacronymstyle{long-short-desc}%
19732 {%
19733 \GlsUseAcrEntryDispStyle{long-short}%
19734 }%
19735 {%
19736 \GlsUseAcrStyleDefs{long-short}%
19737 \renewcommand*{\GenericAcronymFields}{}%
19738 \renewcommand*{\acronymsort}[2]{##2}%
19739 \renewcommand*{\acronymentry}[1]{%
19740 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
19741 }
19742 \newacronymstyle{long-sp-short-desc}%
19743 {%
19744 \GlsUseAcrEntryDispStyle{long-sp-short}%
19745 }%
19746 {%
19747 \GlsUseAcrStyleDefs{long-sp-short}%
19748 \renewcommand*{\GenericAcronymFields}{}%

```

```

19749 \renewcommand*{\acronymsort}[2]{##2}%
19750 \renewcommand*{\acronymentry}[1]{%
19751     \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##1}})}%
19752 }
19753 \newacronymstyle{long-sc-short-desc}%
19754 {%
19755     \GlsUseAcrEntryDisplayStyle{long-sc-short}%
19756 }%
19757 {%
19758     \GlsUseAcrStyleDefs{long-sc-short}%
19759 \renewcommand*{\GenericAcronymFields}{}%
19760 \renewcommand*{\acronymsort}[2]{##2}%
19761 \renewcommand*{\acronymentry}[1]{%
19762     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19763 }
19764 \newacronymstyle{long-sm-short-desc}%
19765 {%
19766     \GlsUseAcrEntryDisplayStyle{long-sm-short}%
19767 }%
19768 {%
19769     \GlsUseAcrStyleDefs{long-sm-short}%
19770 \renewcommand*{\GenericAcronymFields}{}%
19771 \renewcommand*{\acronymsort}[2]{##2}%
19772 \renewcommand*{\acronymentry}[1]{%
19773     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19774 }
19775 \newacronymstyle{short-long-desc}%
19776 {%
19777     \GlsUseAcrEntryDisplayStyle{short-long}%
19778 }%
19779 {%
19780     \GlsUseAcrStyleDefs{short-long}%
19781 \renewcommand*{\GenericAcronymFields}{}%
19782 \renewcommand*{\acronymsort}[2]{##2}%
19783 \renewcommand*{\acronymentry}[1]{%
19784     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19785 }
19786 \newacronymstyle{sc-short-long-desc}%
19787 {%
19788     \GlsUseAcrEntryDisplayStyle{sc-short-long}%
19789 }%
19790 {%
19791     \GlsUseAcrStyleDefs{sc-short-long}%
19792 \renewcommand*{\GenericAcronymFields}{}%
19793 \renewcommand*{\acronymsort}[2]{##2}%
19794 \renewcommand*{\acronymentry}[1]{%
19795     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19796 }
19797 \newacronymstyle{sm-short-long-desc}%

```

```

19798 {%
19799   \GlsUseAcrEntryDispStyle{sm-short-long}%
19800 }%
19801 {%
19802   \GlsUseAcrStyleDefs{sm-short-long}%
19803   \renewcommand*{\GenericAcronymFields}{}%
19804   \renewcommand*{\acronymsort}[2]{##2}%
19805   \renewcommand*{\acronymentry}[1]{%
19806     \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19807 }
19808 \newacronymstyle{dua}%
19809 {%
19810   \ifdefempty\glscustomtext
19811   {%
19812     \ifglshaslong{\glslabel}%
19813     {%
19814       \glcifplural
19815       {%
19816         \glscapscase
19817         {%
19818           \glentrylongpl{\glslabel}\glinsert
19819         }%
19820         {%
19821           \Glsentrylongpl{\glslabel}\glinsert
19822         }%
19823         {%
19824           \mfirstucMakeUppercase
19825             {\glentrylongpl{\glslabel}\glinsert}%
19826         }%
19827       }%
19828     }%
19829     \glscapscase
19830     {%
19831       \glentrylong{\glslabel}\glinsert
19832     }%
19833     {%
19834       \Glsentrylong{\glslabel}\glinsert
19835     }%
19836     {%
19837       \mfirstucMakeUppercase
19838         {\glentrylong{\glslabel}\glinsert}%
19839     }%
19840   }%
19841 }%
19842 {%
19843   \glsgenentryfmt
19844 }%
19845 }%
19846 {\glscustomtext\glinsert}%

```

```

19847 }%
19848 {%
19849 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19850 \renewcommand*{\acrfullfmt}[3]{%
19851 \glslink{##1}{##2}{\glsentrylong{##2}##3\space
19852 (\acronymfont{\glsentryshort{##2}})}}%
19853 \renewcommand*{\Acrfullfmt}[3]{%
19854 \glslink{##1}{##2}{\Glsentrylong{##2}##3\space
19855 (\acronymfont{\glsentryshort{##2}})}}%
19856 \renewcommand*{\ACRfullfmt}[3]{%
19857 \glslink{##1}{##2}{%
19858 \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
19859 (\acronymfont{\glsentryshort{##2}})}}}%
19860 \renewcommand*{\acrfullplfmt}[3]{%
19861 \glslink{##1}{##2}{\glsentrylongpl{##2}##3\space
19862 (\acronymfont{\glsentryshortpl{##2}})}}%
19863 \renewcommand*{\Acrfullplfmt}[3]{%
19864 \glslink{##1}{##2}{\Glsentrylongpl{##2}##3\space
19865 (\acronymfont{\glsentryshortpl{##2}})}}%
19866 \renewcommand*{\ACRfullplfmt}[3]{%
19867 \glslink{##1}{##2}{%
19868 \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
19869 (\acronymfont{\glsentryshortpl{##2}})}}}%
19870 \renewcommand*{\glsentryfull}[1]{%
19871 \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
19872 }%
19873 \renewcommand*{\Glsentryfull}[1]{%
19874 \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
19875 }%
19876 \renewcommand*{\glsentryfullpl}[1]{%
19877 \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
19878 }%
19879 \renewcommand*{\Glsentryfullpl}[1]{%
19880 \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
19881 }%
19882 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19883 \renewcommand*{\acronymsort}[2]{##1}%
19884 \renewcommand*{\acronymfont}[1]{##1}%
19885 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19886 }
19887 \newacronymstyle{dua-desc}%
19888 {%
19889 \GlsUseAcrEntryDispStyle{dua}%
19890 }%
19891 {%
19892 \GlsUseAcrStyleDefs{dua}%
19893 \renewcommand*{\GenericAcronymFields}{}%
19894 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
19895 \renewcommand*{\acronymsort}[2]{##2}%

```



```

19896 }%
19897 \newacronymstyle{footnote}%
19898 {%
19899   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
19900 }%
19901 {%
19902   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19903   \glshyperfirstfalse
19904   \renewcommand*{\genacrfullformat}[2]{%
19905     \protect\firstacronymfont{\glsentryshort{##1}}##2%
19906     \protect\footnote{\glsentrylong{##1}}%
19907   }%
19908   \renewcommand*{\Genacrfullformat}[2]{%
19909     \firstacronymfont{\Glsentryshort{##1}}##2%
19910     \protect\footnote{\glsentrylong{##1}}%
19911   }%
19912   \renewcommand*{\genplacrfullformat}[2]{%
19913     \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
19914     \protect\footnote{\glsentrylongpl{##1}}%
19915   }%
19916   \renewcommand*{\Genplacrfullformat}[2]{%
19917     \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
19918     \protect\footnote{\glsentrylongpl{##1}}%
19919   }%
19920   \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19921   \renewcommand*{\acronymsort}[2]{##1}%
19922   \renewcommand*{\acronymfont}[1]{##1}%
19923   \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
19924   \renewcommand*{\acrfullfmt}[3]{%
19925     \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
19926       (\glsentrylong{##2})}%
19927   \renewcommand*{\Acrfullfmt}[3]{%
19928     \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
19929       (\glsentrylong{##2})}%
19930   \renewcommand*{\ACRfullfmt}[3]{%
19931     \glslink[##1]{##2}{%
19932       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
19933         (\glsentrylong{##2})}}}%
19934   \renewcommand*{\acrfullplfmt}[3]{%
19935     \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
19936       (\glsentrylongpl{##2})}%
19937   \renewcommand*{\Acrfullplfmt}[3]{%
19938     \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
19939       (\glsentrylongpl{##2})}%
19940   \renewcommand*{\ACRfullplfmt}[3]{%
19941     \glslink[##1]{##2}{%
19942       \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
19943         (\glsentrylongpl{##2})}}}%
19944   \renewcommand*{\glsentryfull}[1]{%

```

```

19945 \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1}))%
19946 \renewcommand*{\Glsentryfull}[1]{%
19947 \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1}))%
19948 \renewcommand*{\glsentryfullpl}[1]{%
19949 \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1}))%
19950 \renewcommand*{\Glsentryfullpl}[1]{%
19951 \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1}))%
19952 }
19953 \newacronymstyle{footnote-sc}%
19954 {%
19955 \GlsUseAcrEntryDisplayStyle{footnote}%
19956 }%
19957 {%
19958 \GlsUseAcrStyleDefs{footnote}%
19959 \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
19960 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19961 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19962 }%
19963 \newacronymstyle{footnote-sm}%
19964 {%
19965 \GlsUseAcrEntryDisplayStyle{footnote}%
19966 }%
19967 {%
19968 \GlsUseAcrStyleDefs{footnote}%
19969 \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
19970 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19971 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19972 }%
19973 \newacronymstyle{footnote-desc}%
19974 {%
19975 \GlsUseAcrEntryDisplayStyle{footnote}%
19976 }%
19977 {%
19978 \GlsUseAcrStyleDefs{footnote}%
19979 \renewcommand*{\GenericAcronymFields}{}%
19980 \renewcommand*{\acronymsort}[2]{##2}%
19981 \renewcommand*{\acronymentry}[1]{%
19982 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}))%
19983 }
19984 \newacronymstyle{footnote-sc-desc}%
19985 {%
19986 \GlsUseAcrEntryDisplayStyle{footnote-sc}%
19987 }%
19988 {%
19989 \GlsUseAcrStyleDefs{footnote-sc}%
19990 \renewcommand*{\GenericAcronymFields}{}%
19991 \renewcommand*{\acronymsort}[2]{##2}%
19992 \renewcommand*{\acronymentry}[1]{%
19993 \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}))%

```

```

19994 }
19995 \newacronymstyle{footnote-sm-desc}%
19996 {%
19997   \GlsUseAcrEntryDispStyle{footnote-sm}%
19998 }%
19999 {%
20000   \GlsUseAcrStyleDefs{footnote-sm}%
20001   \renewcommand*{\GenericAcronymFields}{}%
20002   \renewcommand*{\acronymsort}[2]{##2}%
20003   \renewcommand*{\acronymentry}[1]{%
20004     \glstrylong{##1}\space (\acronymfont{\glstryshort{##1}})}%
20005 }
20006 \newcommand*{\DefineAcronymSynonyms}{%
20007   \let\acs\acrshort
20008   \let\Acs\Acrshort
20009   \let\acsp\acrshortpl
20010   \let\Acsp\Acrshortpl
20011   \let\acl\acrlong
20012   \let\aclp\acrlongpl
20013   \let\Acl\Acrlong
20014   \let\Aclp\Acrlongpl
20015   \let\acf\acrfull
20016   \let\acfp\acrfullpl
20017   \let\Acf\Acrfull
20018   \let\Acfp\Acrfullpl
20019   \let\ac\gls
20020   \let\Ac\Gls
20021   \let\acp\glspl
20022   \let\Acp\Glspl
20023 }
20024 \ifglssacrshortcuts
20025   \DefineAcronymSynonyms
20026 \fi
20027 \newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
20028   \defglstryfmt[#1]{\glsgenentryfmt}%
20029 }
20030 \newcommand*{\DefaultNewAcronymDef}{%
20031   \edef\@do@newglossaryentry{%
20032     \noexpand\newglossaryentry{\the\glslabeltok}%
20033     {%
20034       type=\acronymtype,%
20035       name={\the\glsshorttok},%
20036       sort={\the\glsshorttok},%
20037       text={\the\glsshorttok},%
20038       first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
20039       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20040       firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
20041                     {\noexpand\expandonce\noexpand\@glo@shortpl}},%
20042       short={\the\glsshorttok},%

```

```

20043     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20044     long={\the\glslongtok},%
20045     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20046     description={\the\glslongtok},%
20047     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20048     \the\glskeylisttok
20049   }%
20050 }%
20051 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20052 \let\@org@gls@assign@plural\gls@assign@plural
20053 \let\@org@gls@assign@descplural\gls@assign@descplural
20054 \def\gls@assign@firstpl##1##2{%
20055   \@@gls@expand@field{##1}{firstpl}{##2}%
20056 }%
20057 \def\gls@assign@plural##1##2{%
20058   \@@gls@expand@field{##1}{plural}{##2}%
20059 }%
20060 \def\gls@assign@descplural##1##2{%
20061   \@@gls@expand@field{##1}{descplural}{##2}%
20062 }%
20063 \do@newglossaryentry
20064 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20065 \let\gls@assign@plural\@org@gls@assign@plural
20066 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20067 }
20068 \newcommand*{\SetDefaultAcronymStyle}{%
20069   \@for\@gls@type:=\@glsacronymlists\do{%
20070     \SetDefaultAcronymDisplayStyle{\@gls@type}%
20071   }%
20072   \renewcommand{\newacronym}[4][ ]{%
20073     \ifx\@glsacronymlists\@empty
20074       \def\@glo@type{\acronymtype}%
20075       \setkeys{glossentry}{##1}%
20076       \DeclareAcronymList{\@glo@type}%
20077       \SetDefaultAcronymDisplayStyle{\@glo@type}%
20078     \fi
20079     \glskeylisttok{##1}%
20080     \glslabeltok{##2}%
20081     \glsshorttok{##3}%
20082     \glslongtok{##4}%
20083     \newacronymhook
20084     \DefaultNewAcronymDef
20085   }%
20086   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
20087 }
20088 \newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
20089 \newcommand*{\acrlinkfootnote}[3]{%
20090   \footnote{\glslink[#1]{#2}{#3}}%
20091 }

```

```

20092 \newcommand*{\acrnolinkfootnote}[3]{%
20093   \footnote{#3}%
20094 }
20095 \newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
20096   \defglsentryfmt[#1]{%
20097     \ifdefempty\glscustomtext
20098     {%
20099       \ifglsused{\glslabel}%
20100       {%
20101         \acronymfont{\glsgenentryfmt}%
20102       }%
20103     {%
20104       \firstacronymfont{\glsgenentryfmt}%
20105       \ifglshassymbol{\glslabel}%
20106       {%
20107         \expandafter\protect\expandafter\acrfootnote\expandafter
20108         {\@gls@link@opts}{\@gls@link@label}%
20109       {%
20110         \glsifplural
20111         {\glsentrysymbolplural{\glslabel}}%
20112         {\glsentrysymbol{\glslabel}}%
20113       }%
20114     }%
20115   }%
20116 }%
20117 {\glscustomtext\glsinsert}%
20118 }%
20119 }
20120 \newcommand*{\DescriptionFootnoteNewAcronymDef}{%
20121   \edef\@do@newglossaryentry{%
20122     \noexpand\newglossaryentry{\the\glslabeltok}%
20123     {%
20124       type=\acronymtype,%
20125       name={\noexpand\acronymfont{\the\glsshorttok}},%
20126       sort={\the\glsshorttok},%
20127       first={\the\glsshorttok},%
20128       firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20129       text={\the\glsshorttok},%
20130       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20131       short={\the\glsshorttok},%
20132       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20133       long={\the\glslongtok},%
20134       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20135       symbol={\the\glslongtok},%
20136       symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20137       \the\glskeylisttok
20138     }%
20139   }%
20140   \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

20141 \let\org@gl@assign@plural\gl@assign@plural
20142 \let\org@gl@assign@symbolplural\gl@assign@symbolplural
20143 \def\gl@assign@firstpl##1##2{%
20144   \@@gl@expand@field{##1}{firstpl}{##2}%
20145 }%
20146 \def\gl@assign@plural##1##2{%
20147   \@@gl@expand@field{##1}{plural}{##2}%
20148 }%
20149 \def\gl@assign@symbolplural##1##2{%
20150   \@@gl@expand@field{##1}{symbolplural}{##2}%
20151 }%
20152 \do@newglossaryentry
20153 \let\gl@assign@plural\org@gl@assign@plural
20154 \let\gl@assign@firstpl\org@gl@assign@firstpl
20155 \let\gl@assign@symbolplural\org@gl@assign@symbolplural
20156 }
20157 \newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
20158   \renewcommand{\newacronym}[4][]{%
20159     \ifx\@gl@acronymlists\@empty
20160       \def\@glo@type{\acronymtype}%
20161       \setkeys{glossentry}{##1}%
20162       \DeclareAcronymList{\@glo@type}%
20163       \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
20164     \fi
20165     \gl@keylisttok{##1}%
20166     \gl@labeltok{##2}%
20167     \gl@shorttok{##3}%
20168     \gl@longtok{##4}%
20169     \newacronymhook
20170     \DescriptionFootnoteNewAcronymDef
20171   }%
20172   \@for\@gl@type:=\@gl@acronymlists\do{%
20173     \SetDescriptionFootnoteAcronymDisplayStyle{\@gl@type}%
20174   }%
20175   \ifgl@acrsmallcaps
20176     \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
20177     \renewcommand*{\acrpluralsuffix}{\gl@supacrpluralsuffix}%
20178   \else
20179     \ifgl@acrsmaller
20180       \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20181     \fi
20182   \fi
20183   \ifgl@acrdua
20184     \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
20185       can’t both be set}{}%
20186   \fi
20187 }%
20188 \newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
20189   \defgl@entryfmt[##1]{\gl@genentryfmt}%

```

```

20190 }
20191 \newcommand*{\DescriptionDUANewAcronymDef}{%
20192   \edef\@do@newglossaryentry{%
20193     \noexpand\newglossaryentry{\the\glslabeltok}%
20194     {%
20195       type=\acronymtype,%
20196       name={\the\glslongtok},%
20197       sort={\the\glslongtok},
20198       text={\the\glslongtok},%
20199       first={\the\glslongtok},%
20200       plural={\noexpand\expandonce\noexpand\@glo@longpl},%
20201       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20202       short={\the\glsshorttok},%
20203       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20204       long={\the\glslongtok},%
20205       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20206       symbol={\the\glsshorttok},%
20207       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20208       \the\glskeylisttok
20209     }%
20210   }%
20211   \let\@org@gls@assign@firstpl\gls@assign@firstpl
20212   \let\@org@gls@assign@plural\gls@assign@plural
20213   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20214   \def\gls@assign@firstpl##1##2{%
20215     \@gls@expand@field{##1}{firstpl}{##2}%
20216   }%
20217   \def\gls@assign@plural##1##2{%
20218     \@gls@expand@field{##1}{plural}{##2}%
20219   }%
20220   \def\gls@assign@symbolplural##1##2{%
20221     \@gls@expand@field{##1}{symbolplural}{##2}%
20222   }%
20223   \@do@newglossaryentry
20224   \let\gls@assign@firstpl\@org@gls@assign@firstpl
20225   \let\gls@assign@plural\@org@gls@assign@plural
20226   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20227 }
20228 \newcommand*{\SetDescriptionDUAAcronymStyle}{%
20229   \ifglsacrsmallcaps
20230     \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
20231       can’t both be set}{}%
20232   \else
20233     \ifglsacrsmaller
20234       \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
20235         can’t both be set}{}%
20236     \fi
20237   \fi
20238   \renewcommand{\newacronym}[4][\]{%

```

```

20239 \ifx\@glsacronymlists\@empty
20240 \def\@glo@type{\acronymtype}%
20241 \setkeys{glossentry}{##1}%
20242 \DeclareAcronymList{\@glo@type}%
20243 \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
20244 \fi
20245 \glskeylisttok{##1}%
20246 \glslabeltok{##2}%
20247 \glsshorttok{##3}%
20248 \gslongtok{##4}%
20249 \newacronymhook
20250 \DescriptionDUANewAcronymDef
20251 }%
20252 \@for\@gls@type:=\@glsacronymlists\do{%
20253 \SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
20254 }%
20255 }%
20256 \newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
20257 \defglsentryfmt[#1]{%
20258 \ifdefempty\glscustomtext
20259 {%
20260 \ifglsused{\glslabel}%
20261 {%
20262 \let\gls@org@insert\glsinsert
20263 \let\glsinsert\@empty
20264 \acronymfont{\gls@genentryfmt}\gls@org@insert
20265 }%
20266 {%
20267 \gls@genentryfmt
20268 \ifgls@hassymbol{\glslabel}%
20269 {%
20270 \glsifplural
20271 {%
20272 \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
20273 }%
20274 {%
20275 \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
20276 }%
20277 \space(\protect\firstacronymfont
20278 {\gls@scapscase
20279 {\@glo@symbol}
20280 {\@glo@symbol}
20281 {\mfirstucMakeUppercase{\@glo@symbol}}})%
20282 }%
20283 {}%
20284 }%
20285 }%
20286 {\glscustomtext\glsinsert}%
20287 }%

```



```

20288 }
20289 \newcommand*{\DescriptionNewAcronymDef}{%
20290   \edef\@do@newglossaryentry{%
20291     \noexpand\newglossaryentry{\the\glslabeltok}%
20292     {%
20293       type=\acronymtype,%
20294       name={\noexpand
20295         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
20296       sort={\the\glsshorttok},%
20297       first={\the\glslongtok},%
20298       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20299       text={\the\glsshorttok},%
20300       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20301       short={\the\glsshorttok},%
20302       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20303       long={\the\glslongtok},%
20304       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20305       symbol={\noexpand\@glo@text},%
20306       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20307       \the\glskeylisttok}%
20308   }%
20309   \let\@org@gls@assign@firstpl\gls@assign@firstpl
20310   \let\@org@gls@assign@plural\gls@assign@plural
20311   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20312   \def\gls@assign@firstpl##1##2{%
20313     \@gls@expand@field{##1}{firstpl}{##2}%
20314   }%
20315   \def\gls@assign@plural##1##2{%
20316     \@gls@expand@field{##1}{plural}{##2}%
20317   }%
20318   \def\gls@assign@symbolplural##1##2{%
20319     \@gls@expand@field{##1}{symbolplural}{##2}%
20320   }%
20321   \@do@newglossaryentry
20322   \let\gls@assign@firstpl\@org@gls@assign@firstpl
20323   \let\gls@assign@plural\@org@gls@assign@plural
20324   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20325 }
20326 \newcommand*{\SetDescriptionAcronymStyle}{%
20327   \renewcommand{\newacronym}[4][]{%
20328     \ifx\@glsacronymlists\@empty
20329       \def\@glo@type{\acronymtype}%
20330       \setkeys{glossentry}{##1}%
20331       \DeclareAcronymList{\@glo@type}%
20332       \SetDescriptionAcronymDisplayStyle{\@glo@type}%
20333     \fi
20334     \glskeylisttok{##1}%
20335     \glslabeltok{##2}%
20336     \glsshorttok{##3}%

```

```

20337 \glslongtok{##4}%
20338 \newacronymhook
20339 \DescriptionNewAcronymDef
20340 }%
20341 \@for\@gls@type:=\@glsacronymlists\do{%
20342 \SetDescriptionAcronymDisplayStyle{\@gls@type}%
20343 }%
20344 \ifglsacrsmallcaps
20345 \renewcommand{\acronymfont}[1]{\textsc{##1}}
20346 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20347 \else
20348 \ifglsacrsmaller
20349 \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20350 \fi
20351 \fi
20352 }%
20353 \newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
20354 \defglsentryfmt[#1]{%
20355 \ifdefempty\glscustomtext
20356 {%
20357 \let\gls@org@insert\glsinsert
20358 \let\glsinsert\@empty
20359 \ifglsused{\glslabel}%
20360 {%
20361 \acronymfont{\glsgenentryfmt}\gls@org@insert
20362 }%
20363 {%
20364 \firstacronymfont{\glsgenentryfmt}\gls@org@insert
20365 \ifglsashaslong{\glslabel}%
20366 {%
20367 \expandafter\protect\expandafter\acrfootnote\expandafter
20368 {\@gls@link@opts}{\@gls@link@label}%
20369 {%
20370 \glsifplural
20371 {\glsentrylongpl{\glslabel}}%
20372 {\glsentrylong{\glslabel}}%
20373 }%
20374 }%
20375 {}}%
20376 }%
20377 }%
20378 {\glscustomtext\glsinsert}%
20379 }%
20380 }
20381 \newcommand*{\FootnoteNewAcronymDef}{%
20382 \edef\@do@newglossaryentry{%
20383 \noexpand\newglossaryentry{\the\glslabeltok}%
20384 {%
20385 type=\acronymtype,%

```

```

20386     name={\noexpand\acronymfont{\the\glsshorttok}},%
20387     sort={\the\glsshorttok},%
20388     text={\the\glsshorttok},%
20389     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20390     first={\the\glsshorttok},%
20391     firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20392     short={\the\glsshorttok},%
20393     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20394     long={\the\glslongtok},%
20395     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20396     description={\the\glslongtok},%
20397     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20398     \the\glskeylisttok
20399   }%
20400 }%
20401 \let\@org@gls@assign@plural\gls@assign@plural
20402 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20403 \let\@org@gls@assign@descplural\gls@assign@descplural
20404 \def\gls@assign@firstpl##1##2{%
20405   \@gls@expand@field{##1}{firstpl}{##2}%
20406 }%
20407 \def\gls@assign@plural##1##2{%
20408   \@gls@expand@field{##1}{plural}{##2}%
20409 }%
20410 \def\gls@assign@descplural##1##2{%
20411   \@gls@expand@field{##1}{descplural}{##2}%
20412 }%
20413 \do@newglossaryentry
20414 \let\gls@assign@plural\@org@gls@assign@plural
20415 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20416 \let\gls@assign@descplural\@org@gls@assign@descplural
20417 }
20418 \newcommand*\SetFootnoteAcronymStyle{%
20419   \renewcommand{\newacronym}[4][ ]{%
20420     \ifx\@glsacronymlists\@empty
20421       \def\@glo@type{\acronymtype}%
20422       \setkeys{glossentry}{##1}%
20423       \DeclareAcronymList{\@glo@type}%
20424       \SetFootnoteAcronymDisplayStyle{\@glo@type}%
20425     \fi
20426     \glskeylisttok{##1}%
20427     \glslabeltok{##2}%
20428     \glsshorttok{##3}%
20429     \glslongtok{##4}%
20430     \newacronymhook
20431     \FootnoteNewAcronymDef
20432   }%
20433   \@for\@gls@type:=\@glsacronymlists\do{%
20434     \SetFootnoteAcronymDisplayStyle{\@gls@type}%

```

```

20435 }%
20436 \ifglsmallcaps
20437   \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
20438   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20439 \else
20440   \ifglsmaller
20441     \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20442   \fi
20443 \fi
20444 \ifglacrdua
20445   \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
20446   can’t both be set}{}%
20447 \fi
20448 }%
20449 \DeclareRobustCommand*{\glsdoparenifnotempty}[2]{%
20450   \protected@edef\gls@tmp{#1}%
20451   \ifdefempty\gls@tmp
20452   {%
20453   {%
20454     \ifx\gls@tmp\@gls@default@value
20455     \else
20456       \space (#2{#1})%
20457     \fi
20458   }%
20459 }
20460 \newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
20461   \defglentryfmt[#1]{%
20462     \ifdefempty\glscustomtext
20463     {%
20464       \let\gls@org@insert\glsinsert
20465       \let\glsinsert\@empty
20466       \ifglused{\glslabel}%
20467       {%
20468         \acronymfont{\glsgenentryfmt}\gls@org@insert
20469       }%
20470     }%
20471     \glsgenentryfmt
20472     \ifglshassymbol{\glslabel}%
20473     {%
20474       \glsifplural
20475       {%
20476         \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
20477       }%
20478     }%
20479     \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
20480   }%
20481   \space
20482   (\glscapscase
20483     {\firstacronymfont{\@glo@symbol}}%

```

```

20484         {\firstacronymfont{\@glo@symbol}}}%
20485         {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})}%
20486     }%
20487     {}%
20488 }%
20489 }%
20490 {\glscustomtext\glsinsert}%
20491 }%
20492 }
20493 \newcommand*{\SmallNewAcronymDef}{%
20494 \edef\@do@newglossaryentry{%
20495 \noexpand\newglossaryentry{\the\glslabeltok}%
20496 {%
20497     type=\acronymtype,%
20498     name={\noexpand\acronymfont{\the\glsshorttok}},%
20499     sort={\the\glsshorttok},%
20500     text={\the\glsshorttok},%
20501     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20502     first={\the\glslongtok},%
20503     firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20504     short={\the\glsshorttok},%
20505     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20506     long={\the\glslongtok},%
20507     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20508     description={\noexpand\@glo@first},%
20509     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20510     symbol={\the\glsshorttok},%
20511     symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20512     \the\glskeylisttok
20513 }%
20514 }%
20515 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20516 \let\@org@gls@assign@plural\gls@assign@plural
20517 \let\@org@gls@assign@descplural\gls@assign@descplural
20518 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20519 \def\gls@assign@firstpl##1##2{%
20520 \@@gls@expand@field{##1}{firstpl}{##2}%
20521 }%
20522 \def\gls@assign@plural##1##2{%
20523 \@@gls@expand@field{##1}{plural}{##2}%
20524 }%
20525 \def\gls@assign@descplural##1##2{%
20526 \@@gls@expand@field{##1}{descplural}{##2}%
20527 }%
20528 \def\gls@assign@symbolplural##1##2{%
20529 \@@gls@expand@field{##1}{symbolplural}{##2}%
20530 }%
20531 \@do@newglossaryentry
20532 \let\gls@assign@firstpl\@org@gls@assign@firstpl

```

```

20533 \let\gls@assign@plural\@org@gls@assign@plural
20534 \let\gls@assign@descplural\@org@gls@assign@descplural
20535 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20536 }
20537 \newcommand*{\SetSmallAcronymStyle}{%
20538   \renewcommand{\newacronym}[4][]{%
20539     \ifx\@glsacronymlists\@empty
20540       \def\@glo@type{\acronymtype}%
20541       \setkeys{glossentry}{##1}%
20542       \DeclareAcronymList{\@glo@type}%
20543       \SetSmallAcronymDisplayStyle{\@glo@type}%
20544     \fi
20545     \glskeylisttok{##1}%
20546     \glslabeltok{##2}%
20547     \glsshorttok{##3}%
20548     \glslongtok{##4}%
20549     \newacronymhook
20550     \SmallNewAcronymDef
20551   }%
20552   \@for\@gls@type:=\@glsacronymlists\do{%
20553     \SetSmallAcronymDisplayStyle{\@gls@type}%
20554   }%
20555   \ifglsacrsmallcaps
20556     \renewcommand*{\acronymfont}[1]{\textsc{##1}}
20557     \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20558   \else
20559     \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}
20560   \fi
20561   \ifglsacrdua
20562     \ifglsacrsmallcaps
20563       \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
20564         can’t both be set}{}%
20565     \else
20566       \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
20567         can’t both be set}{}%
20568     \fi
20569   \fi
20570 }%
20571 \newcommand*{\SetDUADisplayStyle}[1]{%
20572   \defglsentryfmt[#1]{\glsgenentryfmt}%
20573 }
20574 \newcommand*{\DUANewAcronymDef}{%
20575   \edef\@do@newglossaryentry{%
20576     \noexpand\newglossaryentry{\the\glslabeltok}%
20577     {%
20578       type=\acronymtype,%
20579       name={\the\glsshorttok},%
20580       text={\the\glslongtok},%
20581       first={\the\glslongtok},%

```

```

20582 plural={\noexpand\expandonce\noexpand\@glo@longpl},%
20583 firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20584 short={\the\glsshorttok},%
20585 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20586 long={\the\glslongtok},%
20587 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20588 description={\the\glslongtok},%
20589 descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20590 symbol={\the\glsshorttok},%
20591 symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20592 \the\glskeylisttok
20593 }%
20594 }%
20595 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20596 \let\@org@gls@assign@plural\gls@assign@plural
20597 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20598 \let\@org@gls@assign@descplural\gls@assign@descplural
20599 \def\gls@assign@firstpl##1##2{%
20600   \@gls@expand@field{##1}{firstpl}{##2}%
20601 }%
20602 \def\gls@assign@plural##1##2{%
20603   \@gls@expand@field{##1}{plural}{##2}%
20604 }%
20605 \def\gls@assign@symbolplural##1##2{%
20606   \@gls@expand@field{##1}{symbolplural}{##2}%
20607 }%
20608 \def\gls@assign@descplural##1##2{%
20609   \@gls@expand@field{##1}{descplural}{##2}%
20610 }%
20611 \do@newglossaryentry
20612 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20613 \let\gls@assign@plural\@org@gls@assign@plural
20614 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20615 \let\gls@assign@descplural\@org@gls@assign@descplural
20616 }
20617 \newcommand*\SetDUASstyle{%
20618   \renewcommand{\newacronym}[4][\]{%
20619     \ifx\@glsacronymlists\@empty
20620       \def\@glo@type{\acronymtype}%
20621       \setkeys{glossentry}{##1}%
20622       \DeclareAcronymList{\@glo@type}%
20623       \SetDUADisplayStyle{\@glo@type}%
20624     \fi
20625     \glskeylisttok{##1}%
20626     \glslabeltok{##2}%
20627     \glsshorttok{##3}%
20628     \glslongtok{##4}%
20629     \newacronymhook
20630     \DUANewAcronymDef

```

```

20631 }%
20632 \@for\@gls@type:=\@glsacronymlists\do{%
20633   \SetDUADisplayStyle{\@gls@type}%
20634 }%
20635 }
20636 \newcommand*\SetAcronymStyle{%
20637   \SetDefaultAcronymStyle
20638   \ifglsacrdescription
20639     \ifglsacrfootnote
20640       \SetDescriptionFootnoteAcronymStyle
20641     \else
20642       \ifglsacrdua
20643         \SetDescriptionDUAAcronymStyle
20644       \else
20645         \SetDescriptionAcronymStyle
20646       \fi
20647     \fi
20648   \else
20649     \ifglsacrfootnote
20650       \SetFootnoteAcronymStyle
20651     \else
20652       \ifthenelse{\boolean{glsacrsmalldcaps}\OR
20653         \boolean{glsacrsmaller}}{%
20654         {%
20655           \SetSmallAcronymStyle
20656         }%
20657         {%
20658           \ifglsacrdua
20659             \SetDUASStyle
20660           \fi
20661         }%
20662       \fi
20663     \fi
20664 }
20665 \SetAcronymStyle
20666 \newcommand*\SetCustomDisplayStyle}[1]{%
20667   \defglsentryfmt[#1]{\glsentryfmt}%
20668 }
20669 \newcommand*\CustomAcronymFields{%
20670   name={\the\glsshorttok},%
20671   description={\the\glslongtok},%
20672   first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
20673   firstplural={\acrfullformat
20674     {\noexpand\glsentrylongpl{\the\glslabeltok}}}%
20675     {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
20676   text={\the\glsshorttok},%
20677   plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
20678 }
20679 \newcommand*\CustomNewAcronymDef{%

```



```

20680 \protected@edef\do@newglossaryentry{%
20681   \noexpand\newglossaryentry{\the\glslabeltok}%
20682   {%
20683     type=\acronymtype,%
20684     short={\the\glsshorttok},%
20685     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20686     long={\the\glslongtok},%
20687     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20688     user1={\the\glsshorttok},%
20689     user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
20690     user3={\the\glslongtok},%
20691     user4={\the\glslongtok\noexpand\acrpluralsuffix},%
20692     \CustomAcronymFields,%
20693     \the\glskeylisttok
20694   }%
20695 }%
20696 \do@newglossaryentry
20697 }
20698 \newcommand*\SetCustomStyle{%
20699   \renewcommand{\newacronym}[4][\{
20700     \ifx\@glsacronymlists\empty
20701       \def\@glo@type{\acronymtype}%
20702       \setkeys{glossentry}{##1}%
20703       \DeclareAcronymList{\@glo@type}%
20704       \SetCustomDisplayStyle{\@glo@type}%
20705     \fi
20706     \glskeylisttok{##1}%
20707     \glslabeltok{##2}%
20708     \glsshorttok{##3}%
20709     \glslongtok{##4}%
20710     \newacronymhook
20711     \CustomNewAcronymDef
20712   }%
20713   \@for\@gls@type:=\@glsacronymlists\do{%
20714     \SetCustomDisplayStyle{\@gls@type}%
20715   }%
20716 }
20717 \RequirePackage{glossary-hypernav}
20718 \@gls@loadlist
20719 \@gls@loadlong
20720 \@gls@loadsuper
20721 \@gls@loadtree
20722 \ifx\@glossary@default@style\relax
20723 \else
20724   \setglossarystyle{\@glossary@default@style}
20725 \fi
20726 \newcommand*\showgloparent}[1]{%
20727   \expandafter\show\csname glo@glsdetoklabel{#1}@parent\endcsname
20728 }

```

```

20729 \newcommand*{\showglolevel}[1]{%
20730   \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
20731 }
20732 \newcommand*{\showglotext}[1]{%
20733   \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
20734 }
20735 \newcommand*{\showgloplural}[1]{%
20736   \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
20737 }
20738 \newcommand*{\showglofirst}[1]{%
20739   \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
20740 }
20741 \newcommand*{\showglofirstpl}[1]{%
20742   \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
20743 }
20744 \newcommand*{\showglotype}[1]{%
20745   \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
20746 }
20747 \newcommand*{\showglocounter}[1]{%
20748   \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
20749 }
20750 \newcommand*{\showglouserii}[1]{%
20751   \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
20752 }
20753 \newcommand*{\showglouseriii}[1]{%
20754   \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
20755 }
20756 \newcommand*{\showglouseriv}[1]{%
20757   \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
20758 }
20759 \newcommand*{\showglouseriv}[1]{%
20760   \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
20761 }
20762 \newcommand*{\showglouserv}[1]{%
20763   \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
20764 }
20765 \newcommand*{\showglouservi}[1]{%
20766   \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
20767 }
20768 \newcommand*{\showglongame}[1]{%
20769   \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
20770 }
20771 \newcommand*{\showglodesc}[1]{%
20772   \expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
20773 }
20774 \newcommand*{\showglodescplural}[1]{%
20775   \expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
20776 }
20777 \newcommand*{\showglosort}[1]{%

```

```

20778 \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
20779 }
20780 \newcommand*{\showglosymbol}[1]{%
20781 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
20782 }
20783 \newcommand*{\showglosymbolplural}[1]{%
20784 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
20785 }
20786 \newcommand*{\showgloshort}[1]{%
20787 \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
20788 }
20789 \newcommand*{\showglolong}[1]{%
20790 \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
20791 }
20792 \newcommand*{\showgloindex}[1]{%
20793 \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
20794 }
20795 \newcommand*{\showgloflag}[1]{%
20796 \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
20797 }
20798 \newcommand*{\showgloloclist}[1]{%
20799 \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
20800 }
20801 \newcommand*{\showglofield}[2]{%
20802 \csshow{glo@\glsdetoklabel{#1}@#2}%
20803 }
20804 \newcommand*{\showacronymlists}{%
20805 \show@glsacronymlists
20806 }
20807 \newcommand*{\showglossaries}{%
20808 \show@glo@types
20809 }
20810 \newcommand*{\showglossaryin}[1]{%
20811 \expandafter\show\csname @glotype@#1@in\endcsname
20812 }
20813 \newcommand*{\showglossaryout}[1]{%
20814 \expandafter\show\csname @glotype@#1@out\endcsname
20815 }
20816 \newcommand*{\showglossarytitle}[1]{%
20817 \expandafter\show\csname @glotype@#1@title\endcsname
20818 }
20819 \newcommand*{\showglossarycounter}[1]{%
20820 \expandafter\show\csname @glotype@#1@counter\endcsname
20821 }
20822 \newcommand*{\showglossaryentries}[1]{%
20823 \expandafter\show\csname glolist@#1\endcsname
20824 }
20825 \csname ifglcompatible-2.07\endcsname
20826 \RequirePackage{glossaries-compatible-207}

```

20827\fi

# Glossary

`makeindex` An indexing application [9](#), [13](#), [29](#), [32](#), [189](#), [199](#)

`xindy` An flexible indexing application with multilingual support written in Perl [9](#), [13](#), [29](#), [32](#), [189](#), [199](#)

# Change History

1.01 (2007-05-17)	numberline: numberline option added .. 7
General: Added range facility in format key ..... 120	1.12 (2008-03-08)
\writeist: Added spaces after \delimN and \delimR in ist file ..... 169	\@GLSpl: now uses
1.04 (2007-08-03)	\glentrydescplural and
General: Added \glstextformat ..... 105	\glentrysymbolplural instead of
1.05 (2007-08-10)	\glentrydesc and
\glossarysection: added \@mkboth to \glossarysection ..... 46	\glentrysymbol ..... 135
\gls@defglossaryentry: Changed the default value of the sort key to just the value of the name key ..... 88	\@Glspl@: now uses
1.07 (2007-09-13)	\glentrydescplural and
\@gls@link: fixed bug caused by \theglentrycounter setting the page number too soon ..... 118	\glentrysymbolplural instead of
\glsadd: fixed bug caused by \theglentrycounter setting the page number too soon ..... 165	\glentrydesc and
1.08 (2007-10-13)	\glentrysymbol ..... 134
General: Added babel support ..... 40	\@glspl@: now uses
listgroup: changed listgroup style to use \glsgetgrouptitle ..... 287	\glentrydescplural and
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle ..... 288	\glentrysymbolplural instead of
1.1 (2008-02-22)	\glentrydesc and
\@glossarysection: numbered sections and auto label added ..... 47	\glentrysymbol ..... 133
\@gls@tmpb: changed \toksdef to \newtoks ..... 122	General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget) ..... 129
\@gls@toc: numberline added ..... 48	descriptionplural: new ..... 69
\@p@glossarysection: numbered sections and auto label added ..... 47	\gls@defglossaryentry: Changed default first plural to be first key with s appended (was text key with s appended) ..... 88
General: amsgen now loaded (\new@ifnextchar needed) ..... 3	descriptionplural support added ..... 88
translate: translate option added ..... 27	symbolplural support added ..... 88
\setglossarysection: new ..... 47	\Glsentrydescplural: New ..... 158
numberedsection: numberedsection package option added ..... 8	\glentrydescplural: New ..... 158
	\Glsentrysymbolplural: New ..... 159
	\glentrysymbolplural: New ..... 159
	\SetDescriptionFootnoteAcronymStyle: Added \protect before \footnote and \glslink ..... 251
	\SetFootnoteAcronymStyle: Added \protect before \footnote and \glslink ..... 257
	symbolplural: new ..... 70

1.13 (2008-05-10)	
General: fixed bug that ignored 3rd parameter .....	136–143
\ACRfullpl: new .....	231
\Acrfullpl: new .....	230
\acrfullpl: new .....	230
\acrpluralsuffix: New .....	228
\gls@defglossaryentry: Changed default first value .....	88
Changed default firstplural value .....	88
Removed restriction on only using \newglossaryentry in the preamble	93
\newacronym: Removed restriction on only using \newacronym in the preamble .....	228
1.14 (2008-06-17)	
\@gls@hypergroup: new .....	280
General: added nonumberlist key to \printglossary .....	215
added numberedsection key to \printglossary .....	214
\firstacronymfont: new .....	231
\glsautoprefix: new .....	8
\glsnavhyperlink: changed \edef to \protected@edef .....	279
\glsnavhypertarget: added write to aux file .....	279
\glsnavigation: changed to only use labels for groups that are present ..	281
1.15 (2008-08-15)	
\@gls@link: added \glslabel .....	118
\gls@defglossaryentry: check for \@glo@first in description .....	92
check for \@glo@text in symbol .....	92
\gls@hypergroup: new .....	280
\glsnavhypertarget: added check if rerun required .....	279
\glssettoctitle: new .....	39
\printglossary: changed the way the TOC title is set .....	199
1.16 (2008-08-27)	
\@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 .....	131
\@GLspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used .....	134
\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used .....	131
\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used .....	135
\@glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used .....	133
\@glstarget: raised the hypertarget so the target text doesn't scroll off the top of the page .....	129
\gls@defglossaryentry: Changed def to let .....	88
1.17 (2008-12-26)	
\@do@esc@wrglossary: new .....	193
\@do@seeglossary: new .....	196
\@glo@storeentry: new .....	94
\@gls@glossary: changed definition to use \index instead of \@index ....	189
\@glsdefaultplural: new .....	74
\@glsdefaultsort: new .....	74
\@gls@hypernumber: new .....	225
\@glsnoname: new .....	73
\@glsnonextpages: new .....	215
General: added xindy support .....	29
parent: new .....	71
see: new .....	71
\gls@defglossaryentry: added nonumberlist key .....	88
added parent key .....	88
added see key .....	88
Stored main part of entry format when entry is defined .....	93
\gls@suffixF: new .....	44
\gls@suffixFF: new .....	44
\gls@wrglossary: modified to allow for xindy support .....	190

\glshyperlink: new .....	165	\SetDescriptionFootnoteAcronymStyle:	
\glshypernumber: modified to allow		changed \acronymfont to use	
material to be attached to location .	225	\textsmaller instead of \smaller	251
\glshnavhyperlink: replaced		\SetFootnoteAcronymStyle: changed	
\hyperlink to \@glslink .....	279	\acronymfont to use \textsmaller	
\glshnavhypertarget: replaced		instead of \smaller .....	257
\hypertarget to \@glstarget ...	279	\SetSmallAcronymStyle: changed	
\glsssee: new .....	197	\acronymfont to use \textsmaller	
\glssseeformat: new .....	197	instead of \smaller .....	260
\glssSetSuffixF: new .....	44	1.46 (2021-09-20)	
\glssSetSuffixFF: new .....	44	\@printglossary: changed \xdef to	
\ifglsxindy: new .....	29	\protected\xdef .....	202
\istfilename: added xindy support ...	43	General: changed \edef to	
\newglossarystyle: made		\protected@edef .....	395
\newglossarystyle long .....	224	\glss@defglossaryentry: changed	
\nopostdesc: new .....	42	\xdef to \protected\xdef .....	90, 93
nonumberlist: new .....	71	\glssaddprotectedpagefmt: changed	
\printglossary: added check to		\eappto to \protected@eappto ..	191
determine if \printglossary is		2.01 (2009 May 30)	
already defined .....	199	\@glss@link: moved \@do@wrglossary	
added print language to aux file .....	199	before term is displayed to prevent	
order: order package option added ...	29	unwanted whatsit .....	119
\writeist: added xindy support .....	169	\forallglossaries: replaced	
1.18 (2009-01-14)		\ifthenelse with \ifx .....	58
\@glss@loadlist: new .....	10	\forallglsentries: replaced \ifthenelse	
\@glss@loadlong: new .....	9	with \ifx .....	58
\@glss@loadsuper: new .....	10	\glssdefmain: new .....	16
\@glss@loadtree: new .....	10	\glssdescwidth: changed \linewidth to	
\glss@defglossaryentry: Changed		\hsize .....	289, 311
default value of sort to		\glsslistdottedwidth: changed	
\@glssdefaultsort .....	88	\linewidth to \hsize .....	289
moved sort sanitization to		\glsspagelistwidth: changed	
\newglossaryentry .....	92	\linewidth to \hsize .....	290, 312
\glstarget: new .....	218	nomain: added nomain package option .	16
\oldacronym: new .....	227	\writeist: removed item_02 - no such	
nolist: new .....	10	makeindex key .....	173
nolong: new .....	10	2.02 (2007-07-13)	
sort: moved sanitization to		\@printglossary: suppressed warning	
\newglossaryentry .....	69	globally rather than locally .....	202
nostyles: new .....	10	2.02 (2009-07-13)	
nosuper: new .....	10	\glossarysection: changed \@mkboth	
notree: new .....	10	to \glossarymark .....	46
1.19 (2009-03-02)		\glsglossarymark: New .....	46
\glsclearpage: new .....	48	2.03 (2009-09-23)	
\glssdisp: new .....	135	\@GLS@: Added check for hyperfirst ....	132
\SetDescriptionAcronymStyle:		\@GLSpl: Added check for hyperfirst ...	135
changed \acronymfont to use		\@Gls@: Added check for hyperfirst ....	131
\textsmaller instead of \smaller	255	\@Glspl@: Added check for hyperfirst ..	134
		\@Gls@: Added check for hyperfirst ....	131



\@gls@link: new .....	117	\glsentryuseriv: new .....	162
\@gls@link: added \leavevmode ....	118	\Glsentryuseriv: new .....	162
Moved entry existence check to avoid		\glsentryuseriv: new .....	162
duplicate code .....	118	\Glsentryuseriv: new .....	162
\@glsdisp: Added check for hyperfirst .	135	\glsentryuseriv: new .....	162
\@glspl0: Added check for hyperfirst ..	133	\ns@newglossary: added check to	
\gls glossarymark: Added check to see		determine if \@gls@<type>@display	
if it's already defined .....	46	and \@gls@<type>@displayfirst	
hyperfirst: new .....	28	have been defined. ....	66
2.04 (2009-11-10)		\SetAcronymLists: new .....	19
\@GLS@: Changed test to check if glossary		\SetDefaultAcronymDisplayStyle:	
type has been identified as a list of		new .....	246
acronyms .....	132	\SetDefaultAcronymStyle: new ....	248
\@GLSp1@: Changed test to check if		\SetDescriptionAcronymDisplayStyle:	
glossary type has been identified as a		new .....	253
list of acronyms .....	135	\SetDescriptionDUAAcronymDisplayStyle:	
\@GLS@: Changed test to check if glossary		new .....	251
type has been identified as a list of		\SetDescriptionFootnoteAcronymDisplayStyle:	
acronyms .....	131	new .....	249
\@GLSp1@: Changed test to check if		\SetDUADisplayStyle: new .....	261
glossary type has been identified as a		\SetFootnoteAcronymDisplayStyle:	
list of acronyms .....	134	new .....	255
\@glossaryentryfield: new .....	94	\SetSmallAcronymDisplayStyle: new	258
\@glossarysubentryfield: new ....	94	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 .....	135
acronyms .....	131	Removed spurious brace. Patch	
\@glsacronymlists: new .....	18	provided by Sergiu Dotenco .....	136
\@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 .....	135	provided by Segiu Dotenco .....	174
\@GLSp1@: Changed test to check if		2.06 (2010-06-14)	
glossary type has been identified as a		\altnewglossary: new .....	67
list of acronyms .....	133	\CustomAcronymFields: new .....	263
\@newglossaryentryposthook: new ..	94	\CustomNewAcronymDef: new .....	264
\@newglossaryentryprehook: new ...	94	\SetCustomDisplayStyle: new ....	263
acronymlists: new .....	19	\SetCustomStyle: new .....	264
\DeclareAcronymList: new .....	18	2.07 (2010-07-10)	
\DefineAcronymSynonyms: new ....	244	General: glsadd format key stored in	
\@gls@defglossaryentry: added user1-6		\@glsnumberformat (was mistakenly	
keys .....	89	stored in \@glo@format) .....	165
\glsadd: fixed bug that ignored counter	165	3.0 (2010-07-12)	
\Glsentryuseri: new .....	161	\@makeglossary: Added check for	
\glsentryuseri: new .....	161	savewrites .....	178
\Glsentryuserii: new .....	161	\gls@wrglossary: modified to take into	
\glsentryuserii: new .....	161	account savewrites .....	190
\Glsentryuseriii: new .....	161	3.0 (2010/03/31)	
\glsentryuseriii: new .....	161	\@set@glo@numformat: added 4th	
\Glsentryuseriv: new .....	162	argument .....	120

### 3.0 (2011-04-02)

\@do@esc@wrglossary: added check for hyper location prefix .....	195	\gls@codepage: replaced \ifundefined with \ifcsundef ..	30
modified to use new format .....	193	\gls@defglossaryentry: added \@gls@defsort .....	92
\@glossarysec: replaced \ifundefined with \ifcsundef ...	7	added short and long keys .....	89
\@do@seeglossary: Sanitize and escape cross-referencing information ....	196	replaced \@ifundefined with \ifcsundef .....	89
\@gls@counterwithin: new .....	12	\gls@doclearpage: replaced \ifundefined with \ifcsundef ..	48
\@gls@ifinlist: new .....	49	\glsadd: added \@gls@saveentrycounter .....	166
\@gls@link: added \@gls@saveentrycounter .....	119	\GlsAddXdyCounters: new .....	49
added \@gls@setsort .....	119	\glsentrycounterlabel: new .....	217
\@gls@saveentrycounter: new .....	119	\glsentryitem: new .....	217
\@gls@setupsort@def: new .....	14	\Glsentrylong: new .....	163
\@gls@setupsort@standard: new ....	13	\glsentrylong: new .....	163
\@gls@setupsort@use: new .....	14	\Glsentrylongpl: new .....	163
\@gls@xdy@locationlist: new .....	52	\glsentrylongpl: new .....	163
\@glslink: replaced \@ifundefined with \ifcsundef .....	129	\Glsentryshort: new .....	162
\@glsnextpages: new .....	215	\glsentryshort: new .....	162
\@print@glossary: replaced \ifundefined with \ifcsundef ..	203	\Glsentryshortpl: new .....	163
\@printglossary: added \currentglossary .....	201	\glsentryshortpl: new .....	162
added \glsnextpages .....	201	\glsgetgrouptitle: replaced \ifundefined with \ifcsundef ..	221
make toctitle default to title .....	201	\gls glossarymark: replaced \ifundefined with \ifcsundef ..	46
\@xdy@attributelist: new .....	49	\gls hyperlink: changed default from \glsentryname to \glsentrytext	165
General: added prefix to hyperlink ....	226	\gls hypernumber: replaced \ifundefined with \ifcsundef ..	225
etoolbox now loaded .....	3	\glsnumberformat: replaced \ifundefined with \ifcsundef ..	44
replaced \@ifundefined with \ifcsundef .....	38, 40, 115, 213	\glsrefentry: new .....	217
\acrfootnote: new .....	248	\glsresetsubentrycounter: new ...	216
\ACRfull: added starred version ....	230	\glsseeitem: hyperlink uses \glsseeitemformat instead of	
\Acrfull: added starred version ....	229	\glsentryname .....	198
\acrfull: added starred version ....	228	\glsseeitemformat: new .....	198
\ACRfullpl: added starred version ...	231	\gls sortnumberfmt: new .....	14
\Acrfullpl: added starred version ...	230	\glsstepentry: new .....	216
\acrfullpl: added starred version ...	230	\glsstepsubentry: new .....	216
\acrlinkfootnote: new .....	248	\gls subentrycounterlabel: new ...	217
\acrnolinkfootnote: new .....	248	\gls subentryitem: new .....	217
savewrites: new .....	33	theglossary: replaced \@ifundefined with \ifcsundef .....	217
see: added \@glo@seeautonumberlist	71	short: new .....	73
seeautonumberlist: new .....	9	shortplural: new .....	73
\glossarysection: replaced \ifundefined with \ifcsundef ..	46	\ifglossaryexists: replaced \ifundefined with \ifcsundef ..	59
\glossarystyle: replaced \ifundefined with \ifcsundef ..	223		

<code>\ifglentryexists</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	59
<code>\istfile</code> : deprecated .....	188
<code>glossaryentry</code> : new .....	11
<code>glossarysubentry</code> : new .....	12
<code>\newglossaryentry</code> : replaced	
<code>\DeclareRobustCommand</code> with	
<code>\newrobustcmd</code> .....	76
<code>\newglossarystyle</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	224
<code>\ns@newglossary</code> : added	
<code>\@gls@defsortcount</code> .....	67
replaced <code>\@ifundefined</code> with	
<code>\ifcsundef</code> .....	66
<code>entrycounter</code> : new .....	12
<code>\oldacronym</code> : replaced <code>\@ifundefined</code>	
with <code>\ifcsundef</code> .....	227
compatible-2.07: compatible-2.07	
option added .....	33
<code>long</code> : new .....	73
<code>longplural</code> : new .....	73
<code>nonumberlist</code> : now boolean .....	71
<code>sort</code> : new .....	13
<code>counter</code> : replaced <code>\@ifundefined</code> with	
<code>\ifcsundef</code> .....	70
<code>counterwithin</code> : new .....	12
<code>\printglossary</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	199
<code>\SetDescriptionFootnoteAcronymDisplayStyle</code>	
expanded options link options ....	249
<code>\setentrycounter</code> : added optional	
argument .....	222
<code>\showacronymlists</code> : new .....	269
<code>\showglocounter</code> : new .....	266
<code>\showglodesc</code> : new .....	268
<code>\showglodescplural</code> : new .....	268
<code>\showglofirst</code> : new .....	266
<code>\showglofirstpl</code> : new .....	266
<code>\showgloflag</code> : new .....	269
<code>\showgloindex</code> : new .....	269
<code>\showglolevel</code> : new .....	265
<code>\showglongame</code> : new .....	267
<code>\showgloparent</code> : new .....	265
<code>\showgloplural</code> : new .....	266
<code>\showglosort</code> : new .....	268
<code>\showglossaries</code> : new .....	269
<code>\showglossarycounter</code> : new .....	270
<code>\showglossaryentries</code> : new .....	270
<code>\showglossaryin</code> : new .....	270
<code>\showglossaryout</code> : new .....	270
<code>\showglossarytitle</code> : new .....	270
<code>\showglosymbol</code> : new .....	268
<code>\showglosymbolplural</code> : new .....	268
<code>\showglotext</code> : new .....	265
<code>\showglotype</code> : new .....	266
<code>\showglouserii</code> : new .....	266
<code>\showglouseriii</code> : new .....	267
<code>\showglouseriv</code> : new .....	267
<code>\showglouserv</code> : new .....	267
<code>\showglouservi</code> : new .....	267
<code>subentrycounter</code> : new .....	12
<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 .....	188
General: made robust .....	132
<code>\ACRfull</code> : made robust .....	230
<code>\Acrfull</code> : made robust .....	229
<code>\acrfull</code> : made robust .....	228
<code>\acrfullformat</code> : removed	
<code>\acronymfont</code> as it should already be	
set in the second argument. ....	229
<code>\ACRfullpl</code> : made robust .....	231
<code>\Acrfullpl</code> : made robust .....	230
<code>\acrfullpl</code> : made robust .....	230
<code>\ACRlong</code> : made robust .....	153
<code>\Acrlong</code> : made robust .....	152
<code>\acrlong</code> : made robust .....	152
<code>\ACRlongpl</code> : made robust .....	155
<code>\Acrlongpl</code> : made robust .....	154
<code>\acrlongpl</code> : made robust .....	154
<code>\ACRshort</code> : made robust .....	149
<code>\Acrshort</code> : made robust .....	149
<code>\acrshort</code> : made robust .....	148
<code>\ACRshortpl</code> : made robust .....	151
<code>\Acrshortpl</code> : made robust .....	151
<code>\acrshortpl</code> : made robust .....	150
<code>\Gls</code> : made robust .....	131
<code>\glsadd</code> : made robust .....	165
<code>\glsaddall</code> : made robust .....	166
<code>\GLSdesc</code> : made robust .....	141
<code>\Glsdesc</code> : made robust .....	140
<code>\glsdesc</code> : made robust .....	140
<code>\GLSdescplural</code> : made robust .....	142
<code>\Glsdescplural</code> : made robust .....	141

\glsdescplural: made robust .....	141	\do@wr@glossary code into	
\glsfirst: made robust .....	137	\@do@wrglossary .....	190
\GLSfirstplural: made robust .....	139	\@gls@missingnumberlist: new .....	74
\Glsfirstplural: made robust .....	139	\@gls@writefiles: added check for	
\glsfirstplural: made robust .....	139	existence of token in case	
\glslink: made robust .....	117	\makeglossaries has been	
\GLSname: made robust .....	140	omitted .....	188
\Glsname: made robust .....	140	\@printglossary: add a way to fetch	
\glsname: made robust .....	139	current entry label .....	202
\GLSpl: made robust .....	134	savenumberlist: new .....	9
\Glspl: made robust .....	133	ucmark: new .....	11
\glspl: made robust .....	133	\gls@defglossaryentry: added	
\GLSplural: made robust .....	138	numberlist element .....	92
\GLSsymbol: made robust .....	142	\gls@save@numberlist: new .....	198
\Glsymbol: made robust .....	142	\gls@wrglossary: added check for	
\glssymbol: made robust .....	142	glossary file defined .....	190
\GLSsymbolplural: made robust .....	143	\glsdisplaynumberlist: new .....	164
\Glsymbolplural: made robust .....	143	\glsentrycounter: set default value ..	119
\glssymbolplural: made robust .....	143	\Glsentryfull: fixed bug (replaced	
\Glstext: made robust .....	137	\glsentryshortpl with	
\glstext: made robust .....	136	\glsentryshort) .....	163
\GLSuseri: made robust .....	144	\glsentryfullpl: fixed bug (replaced	
\Glsuseri: made robust .....	144	\glsentryshort with	
\glsuseri: made robust .....	144	\glsentryshortpl) .....	163
\GLSuserii: made robust .....	145	\glsentrynumberlist: new .....	164
\Glsuserii: made robust .....	145	\glsmoveentry: new .....	94
\glsuserii: made robust .....	144	\glsresetsubentrycounter: new ...	216
\GLSuseriii: made robust .....	146	\ifglshaschildren: new .....	61
\Glsuseriii: made robust .....	145	\ifglshasparent: new .....	61
\glsuseriii: made robust .....	145	\makeglossaries: added list parser ..	183
\GLSuseriv: made robust .....	146	indexonlyfirst: new .....	28
\Glsuseriv: made robust .....	146	\renewglossarystyle: new .....	224
\glsuseriv: made robust .....	146	\showglossaryentries: fixed misspelt	
\GLSuserv: made robust .....	147	command .....	270
\Glsuserv: made robust .....	147	\SmallNewAcronymDef: fixed broken	
\glsuserv: made robust .....	147	short and long plural .....	259
\GLSuservi: made robust .....	148	3.03 (2012/09/21)	
\Glsuservi: made robust .....	148	\@gls@sanitizesort: new .....	22
\glsuservi: made robust .....	148	\@gls@setupsort@standard: used	
3.02 (2012-05-19)		\@gls@sanitizesort .....	13
\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 .....	213
\@do@@wrglossary: changed		\glsinlinedescformat: new .....	283
\@glslocref to		\glsinlineemptydescformat: new ..	283
\theglsentrycounter .....	195	\glsinlinenameformat: new .....	283
\@do@wrglossary: changed		\glsinlinepostchild: new .....	283
\@do@wr@glossary to test for		\glsinlinesubdescformat: new ....	283
indexonlyfirst option; put old		\glsinlinesubnameformat: new ....	283

\glspostinline: replaced “.” with	unwanted whatsit occurring here. . .	203
\glspostdescription . . . . .	General: Added check for doc package . . .	3
list: added check for glsnogroupskip .	added datatool-base as a required	
altlongragged4col: added check for	package . . . . .	3
glsnogroupskip . . . . .	added local key . . . . .	115
altsuperragged4col: added check for	\gls@Alphpage: new . . . . .	191
glsnogroupskip . . . . .	\gls@alphpage: new . . . . .	191
alttree: added check for	\gls@disablepagerefexpansion: new	191
glsnogroupskip . . . . .	\gls@numberpage: new . . . . .	191
index: added check for glsnogroupskip	\gls@protected@pagefmts: new . . .	191
326	\gls@romanpage: new . . . . .	191
nogroupskip: new . . . . .	\glsdefmain: added check for doc	
11	package . . . . .	16
long: added check for glsnogroupskip .	\glsorg@endtheglossary: new . . . . .	4
290	\glsorg@theglossary: new . . . . .	4
long3col: added check for	\PrintChanges: new . . . . .	4
glsnogroupskip . . . . .	3.05 (2013-04-21)	
292	\@@do@esc@wrglossary: add Roman	
long4col: added check for	case. Fixed bugs in the else	
glsnogroupskip . . . . .	statements . . . . .	194
293	\@gls@link: added check for	
longragged: added check for	“nohypertypes” . . . . .	118
glsnogroupskip . . . . .	mcolalttree: replaced ‘2’ with	
301	\glsmcols . . . . .	310
longragged3col: added check for	mcolindex: replaced ‘2’ with \glsmcols	306
glsnogroupskip . . . . .	mcolindexspannav: replaced ‘2’ with	
303	\glsmcols . . . . .	307
no postdot: new . . . . .	mcoltree: replaced ‘2’ with \glsmcols	307
11	mcoltreename: replaced ‘2’ with	
tree: added check for glsnogroupskip .	\glsmcols . . . . .	309
327	mcoltreesspannav: replaced ‘2’ with	
treename: added check for	\glsmcols . . . . .	308
glsnogroupskip . . . . .	\gls@protected@pagefmts: added	
329	Roman to list . . . . .	191
super: added check for glsnogroupskip	\gls@Romanpage: new . . . . .	191
312	\glsgetgrouplabel: fixed bug (typo in	
super3col: added check for	\equal) . . . . .	222
glsnogroupskip . . . . .	\nopostdesc: made robust . . . . .	42
314	3.05 (2013/04/21)	
super4col: added check for	\@gls@nohyperlist: new . . . . .	19
glsnogroupskip . . . . .	\GlsDeclareNoHyperList: new . . . . .	19
316	nohypertypes: new . . . . .	19
superragged: added check for	3.06 (2013/06/17)	
glsnogroupskip . . . . .	\@xdy@main@language: Changed back to	
319	using \language . . . . .	29
superragged3col: added check for	\findrootlanguage: Obsoleted . . . . .	56
glsnogroupskip . . . . .	3.07 (2013-07-05)	
321	\@gls@link: fixed bug that failed to find	
3.04 (2012-11-11)	entry in list . . . . .	118
altlist: replaced \newline with		
paragraph break . . . . .		
288		
3.04 (2012-11-18)		
\@@do@@wrglossary: changed		
\theglsentrycounter back to		
\@glslocref . . . . .		195
\@@do@esc@wrglossary: modified to		
compensate for possible incorrect		
page number . . . . .		194
\@gls@escbsdq: unsanitize		
\gls@numberpage, \gls@alphpage,		
\gls@Alphpage and		
\gls@romanpage . . . . .		121
\@print@glossary: Moved aux write to		
end of document to prevent		

\glossarypreamble: modified to work with \setglossarypreamble .....	45	\glsaddallunused: new .....	166
\gls@docclearpage: added check for openright .....	48	\glsexpandfields: new .....	76
\glspostdescription: Added spacefactor code .....	11	\glsnoexpandfields: new .....	76
\GlsSetXdyCodePage: Added check for fontspec .....	57	\glssee: made robust .....	197
\SetDescriptionAcronymDisplayStyle: now using \glsdoparenifnotempty	253	\glsseeformat: made robust .....	197
\setglossarypreamble: new .....	45	\glsseeitem: made robust .....	198
3.08a (2013-08-30)		\glsseelist: made robust .....	197
list: updated list style to use \glossentry and \subglossentry	287	\ifglsdescsuppressed: new .....	62
listdotted: updated listdotted style to use \glossentry and \subglossentry .....	289	\ifglshasdesc: new .....	62
altlist: updated altlist style to use \glossentry and \subglossentry	288	\ifglshassymbol: new .....	62
inline: updated inline style to use \glossentry and \subglossentry	282	altlongragged4col: updated to use \glossentry and \subglossentry	304
3.08a (2013-09-28)		alttree: updated to use \glossentry and \subglossentry .....	331
\@glo@storeentry: no longer need to check for special characters in any of the fields other than sort .....	95	index: added paragraph break at end of environment .....	325
updated for \glossentry .....	95	updated to use \glossentry and \subglossentry .....	325
\@glossaryentryfield: switched to \glossentry .....	94	long: updated to use \glossentry and \subglossentry .....	290
\@glossarysubentryfield: switched to \subglossentry .....	94	longragged: updated to use \glossentry and \subglossentry	301
General: added nogroupskip key to \printglossary .....	214	longragged3col: updated to use \glossentry and \subglossentry	302
removed definition of \@glossaryentryfield .....	385	tree: updated to use \glossentry and \subglossentry .....	327
removed definition of \@glossarysubentryfield .....	385	\setglossarystyle: new .....	223
\compatibleglossentry: new .....	218	\setglossentrycompatibility: new	220
\compatiblesubglossentry: new ...	219	superragged: updated to use \glossentry and \subglossentry	319
\glossaryentryfield: deprecated ...	220	3.09a (2013-10-09)	
\Glossentrydesc: new .....	219	\@gls@assign@symbolplural@field: new .....	22
\glossentrydesc: new .....	219	\@gls@default@value: new .....	70
\Glossentryname: new .....	219	\Glsentrydesc: made robust .....	158
\glossentryname: new .....	218	\Glsentrydescplural: made robust ..	158
\Glossentrysymbol: new .....	219	\Glsentryfirst: made robust .....	160
\glossentrysymbol: new .....	219	\Glsentryfirstplural: made robust .	160
\gls@assign@desc@field: new .....	21	\Glsentryfull: made robust .....	163
\gls@assign@descplural@field: new	21	\Glsentryfullpl: made robust .....	163
\gls@assign@field: new .....	76	\Glsentrylong: made robust .....	163
\gls@ifnotmeasuring: new .....	96	\Glsentrylongpl: made robust .....	163
		\Glsentryname: made robust .....	157
		\Glsentryplural: made robust .....	159
		\Glsentryshort: made robust .....	162
		\Glsentryshortpl: made robust .....	163
		\Glsentrysymbol: made robust .....	159
		\Glsentrysymbolplural: made robust	159
		\Glsentrytext: made robust .....	159

\Glsentryuseri: made robust . . . . .	161	\@GLSpl: add \glslabel,	
\Glsentryuserii: made robust . . . . .	161	\glsifplural, \glscapscase,	
\Glsentryuseriii: made robust . . . . .	161	\glscustomtext and \glsinsert	134
\Glsentryuseriv: made robust . . . . .	162	change to using \glsentryfmt style	
\Glsentryuserv: made robust . . . . .	162	commands . . . . .	135
\Glsentryuservi: made robust . . . . .	162	removed \MakeUppercase as now	
\glstextup: new . . . . .	228	dealt with in \glsentryfmt . . . . .	135
\ifglshassymbol: changed test to check		\@Gls@: add \glsifplural,	
for \@gls@default@symbol . . . . .	62	\glscapscase, \glscustomtext and	
3.10a (2013-09-28)		\glsinsert . . . . .	131
\gls@assign@type@field: new . . . . .	21	change to using \glsentryfmt style	
3.10a (2013-10-13)		commands . . . . .	131
\@gls@keymap: new . . . . .	79	removed \makefirstuc (now dealt	
\@gls@provide@newglossary: new . . .	65	with in \glsentryfmt) . . . . .	131
\@gls@writedef: new . . . . .	79	\@Glspl@: add \glsifplural,	
\@glsdefaultplural: Obsolete . . . . .	74	\glscapscase, \glscustomtext and	
\@glsnodelsc: new . . . . .	74	\glsinsert . . . . .	134
\@print@glossary: Added		change to using \glsentryfmt style	
providecommand code to aux file . .	203	commands . . . . .	134
\gls@defglossaryentry: Changed to		removed \makefirstuc (now dealt	
using \@gls@default@value . . . . .	88	with in \glsentryfmt) . . . . .	134
new . . . . .	88	\@acrlong: added \glslabel,	
\glswritedefhook: new . . . . .	87	\glsifplural, \glscapscase,	
\makeglossaries: Added		\glsinsert and \glscustomtext	378
providecommand code to aux file . .	182	\@acrshort: added \glslabel,	
\new@glossaryentry: new . . . . .	77	\glsifplural, \glscapscase,	
\ns@newglossary: added		\glsinsert and \glscustomtext	377
\@gls@provide@newglossary . . . .	66	\@gls@: add \glslabel, \glsifplural,	
3.11a (2013-10-15)		\glscapscase, \glscustomtext and	
\@ACRlong: added \glslabel,		\glsinsert . . . . .	130
\glsifplural, \glscapscase,		change to using \glsentryfmt style	
\glsinsert and \glscustomtext	379	commands . . . . .	131
\@ACRshort: added \glslabel,		\@gls@noexpand@fields: Fixed bug	
\glsifplural, \glscapscase,		expand replaced with noexpand . . . .	75
\glsinsert and \glscustomtext	378	\@glsdisp: add \glslabel,	
\@Acrlong: added \glslabel,		\glsifplural, \glscapscase,	
\glsifplural, \glscapscase,		\glscustomtext and \glsinsert	135
\glsinsert and \glscustomtext	379	change to using \glsentryfmt style	
\@Acrshort: added \glslabel,		commands . . . . .	135
\glsifplural, \glscapscase,		\@glspl@: add \glslabel,	
\glsinsert and \glscustomtext	377	\glsifplural, \glscapscase,	
\@GLS@: add \glslabel, \glsifplural,		\glscustomtext and \glsinsert	133
\glscapscase, \glscustomtext and		change to using \glsentryfmt style	
\glsinsert . . . . .	132	commands . . . . .	133
change to using \glsentryfmt style		General: added \glslabel,	
commands . . . . .	132	\glsifplural, \glscapscase,	
removed \MakeUppercase (now		\glsinsert and	
moved to \glsentryfmt) . . . . .	132	\glscustomtext . . . . .	149–155

changed to just use	
\Glsentrydescplural .....	141
changed to just use	
\glsentrydescplural .....	141, 142
changed to just use \Glsentrydesc .	141
changed to just use	
\glsentrydesc .....	140, 141
changed to just use	
\Glsentryfirstplural .....	139
changed to just use	
\glsentryfirstplural .....	139
changed to just use \Glsentryfirst	137
changed to just use	
\glsentryfirst .....	137, 138
changed to just use \Glsentryname .	140
changed to just use \glsentryname .	140
changed to just use \Glsentryplural	138
changed to just use \glsentryplural	138
changed to just use	
\Glsentrysymbolplural .....	143
changed to just use	
\glsentrysymbolplural .....	143
changed to just use \Glsentrysymbol	142
changed to just use	
\glsentrysymbol .....	142, 143
Changed to just use \Glsentrytext .	137
changed to just use \glsentrytext .	136
changed to just use	
\Glsentryuseriii .....	146
changed to just use	
\glsentryuseriii .....	145, 146
changed to just use \Glsentryuserii	145
changed to just use	
\glsentryuserii .....	144, 145
changed to just use \Glsentryuseriv	146
changed to just use	
\glsentryuseriv .....	146, 147
changed to just use \Glsentryuseri	144
changed to just use \glsentryuseri	144
changed to just use \Glsentryuservi	148
changed to just use \glsentryuservi	148
changed to just use \Glsentryuserv	147
changed to just use \glsentryuserv	147
Now requires textcase .....	3
acronymlists: replaced	
\@addtoacronymlists with	
\DeclareAcronymList .....	19
\defglssdisplay: obsoleted .....	114
\defglssdisplayfirst: obsoleted ....	114
\defglssentryfmt: new .....	65
\forglssentries: replaced \ifx with	
\ifdefempty .....	58
\gls@assign@desc: new .....	87
\gls@defglossaryentry: Fixed default	
counter if none supplied .....	91
\gls@doentryfmt: new .....	65
\glssdisplay: obsoleted .....	114
\glssdisplayfirst: obsoleted .....	114
\glssgenentryfmt: new .....	109
\glssgetgrouptitle: Added check in	
case non-Latin alphabet in use ....	221
\glssglossarymark: replaced	
\MakeUppercase with	
\mfirstucMakeUppercase .....	46
\glssnavigation: switched to using	
\@gls@getgrouptitle .....	281
\ifglshasdesc: replaced \ifdefempty	
with \ifcseempty .....	62
\ifglshaslong: new .....	62
\ifglshasshort: new .....	62
\ifglshassymbol: replaced	
\ifdefempty with \ifcseempty ....	62
\ifglssused: replaced \ifthenelse with	
\ifbool .....	59
\longnewglossaryentry: new .....	87
\ns@newglossary: replaced	
\glssdisplay and	
\glssdisplayfirst with	
\glssentryfmt .....	66
compatible-3.07: cnew .....	33
\SetCustomDisplayStyle: updated to	
use \defglssentryfmt .....	263
\SetDefaultAcronymDisplayStyle:	
changed to use \defglssentryfmt .	246
\SetDescriptionAcronymDisplayStyle:	
updated to use \defglssentryfmt .	253
\SetDescriptionDUAAcronymDisplayStyle:	
updated to use \defglssentryfmt .	251
\SetDescriptionFootnoteAcronymDisplayStyle:	
updated to use \defglssentryfmt .	249
\SetDUADisplayStyle: updated to use	
\defglssentryfmt .....	261
\SetFootnoteAcronymDisplayStyle:	
updated to use \defglssentryfmt .	255
\SetSmallAcronymDisplayStyle:	
updated to use \defglssentryfmt .	258
\setupglossaries: new .....	37
\showglolong: new .....	269



\showgloshort: new .....	268	\glsprestandardsort: new .....	13
numbers: new .....	35	\glsetexpandfield: new .....	21
symbols: new .....	35	\glsetnoexpandfield: new .....	21
3.12a (2013-10-16)		altsuper4colheader: switched to	
\gls@defglossaryentry: added		\tabularnewline .....	317
\glslabel .....	88	altsuper4colheaderborder: switched	
\glsaddkey: new .....	81	to \tabularnewline .....	318
3.13a (2013-11-05)		long: switched to \tabularnewline ..	290
\@gls@assign@symbol@field: changed		long3col: switched to	
to use \glsetnoexpandfield ....	22	\tabularnewline .....	291
\@gls@assign@symbolplural@field:		long3colheader: switched to	
changed to use		\tabularnewline .....	292
\glsetnoexpandfield .....	22	long3colheaderborder: switched to	
\@gls@link: removed \relax .....	119	\tabularnewline .....	293
\@gls@notranslatorhook: new .....	26	long4col: switched to	
\@gls@setupsort@standard: moved		\tabularnewline .....	293
\@gls@santizesort		long4colheader: switched to	
\glsprestandardsort .....	13	\tabularnewline .....	294
ucmark: added check for memoir .....	11	longheader: switched to	
see: added \gls@checkseeallowed ...	71	\tabularnewline .....	291
\glossarysection: changed		longheaderborder: switched to	
\glossarymark to		\tabularnewline .....	291
\gls glossarymark .....	46	\SetFootnoteAcronymDisplayStyle:	
\glossarystyle: fixed bug caused by		fixed missing argument bug .....	256
using \ifdef instead of \ifcsdef .	223	super: switched to \tabularnewline .	312
\gls@assign@desc@field: changed to		super3col: switched to	
use \glsetnoexpandfield .....	21	\tabularnewline .....	314
\gls@assign@descplural@field:		super3colheader: switched to	
changed to use		\tabularnewline .....	314
\glsetnoexpandfield .....	21	super4col: switched to	
\gls@assign@name@field: changed to		\tabularnewline .....	315
use \glsetnoexpandfield .....	22	super4colheader: switched to	
\gls@assign@type@field: changed to		\tabularnewline .....	316
use \glsetexpandfield .....	21	super4colheaderborder: switched to	
\gls@checkseeallowed: new .....	71	\tabularnewline .....	316
\glsaddallunused: set default to		superheader: switched to	
\@glo@types .....	166	\tabularnewline .....	313
\Glsentryfull: changed to use		superheaderborder: switched to	
\acrfullformat .....	163	\tabularnewline .....	313
\Glsentryfull: changed to use		3.14a (2013-11-12)	
\acrfullformat .....	163	\@glswritefiles: renamed	
\Glsentryfullpl: changed to use		\glswritefiles to	
\acrfullformat .....	163	\@glswritefiles and used	
\Glsentryfullpl: changed to use		“savewrites” option to set	
\acrfullformat .....	163	\glswritefiles .....	188
\gls glossarymark: renamed		General: new .....	272
\glossarymark to		acronyms: new .....	17
\gls glossarymark to avoid conflict		\gls@defglossaryentry: added check	
with memoir .....	46	for existence of default glossary ....	89

set the default for firstplural to be the value of plural .....	91	\GlsUseAcrEntryDispStyle: new ...	234
xindygloss: new .....	30	\GlsUseAcrStyleDefs: new .....	234
\longprovideglossaryentry: new ...	88	short-long: new .....	236
compatible-2.07: added check for 2.07 before setting 3.07 compatibility ....	33	short-long-desc: new .....	239
notranslate: new .....	26	xindynoglsnumbers: new .....	30
\provideglossaryentry: new .....	77	sm-short-long: new .....	237
4.0 (2013-11-14)		sm-short-long-desc: new .....	239
\gls@defglossaryentry: added check for first key .....	91	index: new .....	36
super: fixed typo in \subglossentry (\glossentrydesc) .....	312	\newacronymstyle: new .....	234
4.01 (2013-11-16)		long-sc-short: new .....	237
General: fixed non-value options so that they can be passed to document class .	8	long-sc-short-desc: new .....	238
\CustomAcronymFields: inserted missing comma .....	264	long-short: new .....	235
4.02 (2013-12-05)		long-short-desc: new .....	238
\@acrfull: now using \acrfullfmt ..	229	long-sm-short: new .....	237
\@gls@indexdef: new .....	36	long-sm-short-desc: new .....	238
\@gls@numbersdef: new .....	36	long-sp-short-desc: new .....	238
\@gls@symbolsdef: new .....	35	footnote: new .....	242
General: Removed \acronymfont .	152–156	footnote-desc: new .....	244
\ACRfullfmt: new .....	230	footnote-sc: new .....	243
\Acrfullfmt: new .....	229	footnote-sc-desc: new .....	244
\acrfullfmt: new .....	229	footnote-sm: new .....	243
\ACRfullplfmt: new .....	231	footnote-sm-desc: new .....	244
\Acrfullplfmt: new .....	231	\setacronymstyle: new .....	233
\acrfullplfmt: new .....	230	\SetDescriptionAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material .....	253
\acronymentry: new .....	233	\SetDescriptionFootnoteAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material .....	249
sanitize: fixed bug that caused an error here .....	25	\SetFootnoteAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material .....	255
sc-short-long: new .....	237	\SetGenericNewAcronym: new .....	232
sc-short-long-desc: new .....	239	\SetSmallAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material .....	258
\Genacrfullformat: new .....	113	dua: new .....	240
\genacrfullformat: new .....	113	dua-desc: new .....	241
\GenericAcronymFields: new .....	233	numberedsection: added nameref option .....	8
\Genplacrfullformat: new .....	113	4.02 (2013-13-05)	
\Glsentryfull: bug fix: added missing \acronymfont .....	163	\makeglossaries: made preamble only	183
\glsentryfull: bug fix: added missing \acronymfont .....	163	4.03 (2014-01-17)	
\Glsentryfullpl: bug fix: added missing \acronymfont .....	163	General: changed default to \@empty instead of \relax .....	33
\glsentryfullpl: bug fix: added missing \acronymfont .....	163		
\glsgenacfmt: new .....	111		

#### 4.03 (2014-01-20)

\@do@esc@wrglossary: added		\compatiblesubglossentry: added	
\glsdetoklabel .....	195	\glsdetoklabel .....	355
\@do@noesc@wrglossary: added		\Genacrfullformat: redefined to use	
\glsdetoklabel .....	192	accessibility information .....	376
\@ACRlong: removed \glslabel		\genacrfullformat: redefined to use	
(defined in \@gls@link) .....	379	accessibility information .....	376
\@ACRshort: removed \glslabel		\Genplacrfullformat: redefined to use	
(defined in \@gls@link) .....	378	accessibility information .....	377
\@Acrlong: removed \glslabel		\genplacrfullformat: redefined to use	
(defined in \@gls@link) .....	379	accessibility information .....	377
\@Acrshort: removed \glslabel		\glossentryname: added	
(defined in \@gls@link) .....	377	\glsdetoklabel .....	218
\@GLS@: removed \glslabel (defined in		\gls@defglossaryentry: added	
\@gls@link) .....	132	\glsdetoklabel .....	88
\@GLSpl: removed \glslabel (defined		replaced #1 with \@gls@label .....	89
in \@gls@link) .....	134	replaced \ifthenelse with	
\@Gls@: removed \glslabel (defined in		\ifdefequal .....	90
\@gls@link) .....	131	\glsadd: added \glsdetoklabel ....	165
\@Gls@entry@field: new .....	156	\glsaddkey: switched to using	
\@Glspl@: removed \glslabel (defined		\@gls@field@link .....	82
in \@gls@link) .....	134	\glsdetoklabel: new .....	59
\@acrlong: removed \glslabel		\glsdisplaynumberlist: added	
(defined in \@gls@link) .....	378	\glsdetoklabel .....	164
\@acrshort: removed \glslabel		\glsdoifexistsorwarn: new .....	60
(defined in \@gls@link) .....	377	\glsentryaccess: switched to using	
\@gls@: removed \glslabel (defined in		\@gls@entry@field .....	360
\@gls@link) .....	130	\glsentrydescaccess: switched to	
\@gls@access@display: new .....	364	using \@gls@entry@field .....	361
\@gls@entry@field: new .....	156	\glsentrydescpluralaccess: switched	
\@gls@fetchfield: new .....	80	to using \@gls@entry@field .....	361
\@gls@field@link: new .....	136	\glsentryfirstaccess: switched to	
\@gls@link: added \glsdetoklabel .	118	using \@gls@entry@field .....	361
moved \@gls@link@opts and		\glsentryfirstplural: added	
\@gls@link@label to \@gls@link	118	\glsdetoklabel .....	160
\@gls@writedef: added		\glsentrylongaccess: switched to	
\glsdetoklabel .....	79	using \@gls@entry@field .....	362
\@glsdisp: removed \glslabel		\glsentrylongpluralaccess: switched	
(defined in \@gls@link) .....	135	to using \@gls@entry@field .....	362
\@glspl@: removed \glslabel (defined		\glsentrypluralaccess: switched to	
in \@gls@link) .....	133	using \@gls@entry@field .....	361
\@printglossary: added		\glsentryshortaccess: switched to	
\glsdetoklabel .....	202	using \@gls@entry@field .....	361
General: removed \glslabel (defined in		\glsentryshortpluralaccess:	
\@gls@link) .....	149	switched to using	
sc-short-long-desc: redefined to use		\@gls@entry@field .....	361
accessibility information .....	389	\glsentrysymbolaccess: switched to	
\compatibleglossentry: added		using \@gls@entry@field .....	361
\glsdetoklabel .....	354	\glsentrysymbolpluralaccess:	
		switched to using	

\@gls@entry@field .....	361	replaced \ifcsempy with	
\glsentrytextaccess: switched to		\ifdefempty and replaced \ifx with	
using \@gls@entry@field .....	360	\ifdefequal .....	62
\glsgenacfmt: redefined to use		\ifglsused: added \glsdetoklabel ..	59
accessibility information .....	374	sm-short-long-desc: redefined to use	
\glsgenentryfmt: redefined to use		accessibility information .....	389
accessibility information .....	371	long-sc-short-desc: redefined to use	
\glshyperlink: added		accessibility information .....	388
\glsdetoklabel .....	165	long-short: redefined to use	
\glslocalreset: added		accessibility information .....	386
\glsdetoklabel .....	97	long-short-desc: redefined to use	
\glslocalunset: added		accessibility information .....	387
\glsdetoklabel .....	97	long-sm-short-desc: redefined to use	
\glsmoveentry: added		accessibility information .....	388
\glsdetoklabel .....	94	footnote: redefined to use accessibility	
replaced \ifthenelse with		information .....	392
\ifdefequal .....	94	footnote-desc: redefined to use	
\glsrefentry: added \glsdetoklabel	217	accessibility information .....	394
\glsreset: added \glsdetoklabel ...	96	footnote-sc: redefined to use	
\glsseelist: added \expandafter		accessibility information .....	394
commands .....	198	footnote-sc-desc: redefined to use	
\glsstepentry: added		accessibility information .....	395
\glsdetoklabel .....	216	footnote-sm: redefined to use	
\glsstepsubentry: added		accessibility information .....	394
\glsdetoklabel .....	216	footnote-sm-desc: redefined to use	
\glsunset: added \glsdetoklabel ...	97	accessibility information .....	395
short-long: commented spurious EOL	236	\renewacronymstyle: new .....	234
redefined to use accessibility		\showglocounter: added	
information .....	387	\glsdetoklabel .....	266
short-long-desc: redefined to use		\showglodesc: added \glsdetoklabel	268
accessibility information .....	388	\showglodescaccess: added	
\ifglsdescsuppressed: added		\glsdetoklabel .....	401
\glsdetoklabel .....	62	\showglodescplural: added	
fixed typo .....	62	\glsdetoklabel .....	268
\ifglsentryexists: added		\showglodescpluralaccess: added	
\glsdetoklabel .....	59	\glsdetoklabel .....	401
\ifglschaschildren: added		\showglofirst: added	
\glsdetoklabel .....	61	\glsdetoklabel .....	266
\ifglschasdesc: added		\showglofirstaccess: added	
\glsdetoklabel .....	62	\glsdetoklabel .....	401
\ifglschasfield: new .....	63	\showglofirstpl: added	
\ifglschaslong: added		\glsdetoklabel .....	266
\glsdetoklabel .....	62	\showglofirstpluralaccess: added	
\ifglschasparent: added		\glsdetoklabel .....	401
\glsdetoklabel .....	61	\showgloflag: added \glsdetoklabel	269
\ifglschasshort: added		\showgloindex: added	
\glsdetoklabel .....	62	\glsdetoklabel .....	269
\ifglschassymbol: added		\showglolevel: added	
\glsdetoklabel .....	62	\glsdetoklabel .....	265

\showglolong: added \glsdetoklabel	269	redefined to use accessibility	
\showglolongaccess: added		information .....	389
\glsdetoklabel .....	402	dua-desc: commented spurious EOL ..	242
\showglolongpluralaccess: added		redefined to use accessibility	
\glsdetoklabel .....	402	information .....	392
\showglongname: added \glsdetoklabel	267	4.04 (2014-03-04)	
\showglongnameaccess: added		\@gls@getcounterprefix: added	
\glsdetoklabel .....	400	warning if no prefix can be formed ..	196
\showgloparent: added		4.04 (2014-03-06)	
\glsdetoklabel .....	265	\@gls@noidx@nosanitizesort: new ..	23
\showgloplural: added		\@gls@noidx@sanitizesort: new ...	22
\glsdetoklabel .....	266	\@gls@nosanitizesort: new .....	22
\showglopluralaccess: added		\@gls@sanitizesort: new .....	22
\glsdetoklabel .....	401	\@glo@addchildren: new .....	204
\showgloshort: added		\@glo@do@sortentries: new .....	205
\glsdetoklabel .....	268	\@glo@grabfirst: new .....	210
\showgloshortaccess: added		\@glo@sortedinsert: new .....	205
\glsdetoklabel .....	401	\@glo@sortentries: new .....	204
\showgloshortpluralaccess: added		\@glo@sorthandler@case: new .....	206
\glsdetoklabel .....	402	\@glo@sorthandler@letter: new ...	206
\showglosort: added \glsdetoklabel	268	\@glo@sorthandler@nocase: new ...	206
\showglosymbol: added		\@glo@sorthandler@word: new .....	206
\glsdetoklabel .....	268	\@glo@sortmacro@case: new .....	207
\showglosymbolaccess: added		\@glo@sortmacro@def: new .....	208
\glsdetoklabel .....	401	\@glo@sortmacro@def@do: new .....	208
\showglosymbolplural: added		\@glo@sortmacro@letter: new .....	207
\glsdetoklabel .....	268	\@glo@sortmacro@nocase: new .....	208
\showglosymbolpluralaccess: added		\@glo@sortmacro@standard: new ...	207
\glsdetoklabel .....	401	\@glo@sortmacro@use: new .....	209
\showglotext: added \glsdetoklabel	265	\@glo@sortmacro@word: new .....	207
\showglotextaccess: added		\@gls@noidx@do: new .....	210
\glsdetoklabel .....	401	\@gls@noidx@getgrouptitle: new ..	222
\showglotype: added \glsdetoklabel	266	\@gls@noref@warn: new .....	187
\showglouserii: added		\@gls@reference: new .....	213
\glsdetoklabel .....	266	\@gls@warnonglossdefined: new ....	20
\showglouseriii: added		\@gls@warnontheGLOSSdefined: new ..	21
\glsdetoklabel .....	267	\@no@makeglossaries: new .....	187
\showglouseriiii: added		\@print@glossary: new .....	202
\glsdetoklabel .....	267	\@print@noidx@glossary: new .....	209
\showglouseriv: added		\@print@gloss@setsort: new .....	200
\glsdetoklabel .....	267	\@print@glossary: new .....	200
\showglouseriv: added		General: added sort key to printgloss	
\glsdetoklabel .....	267	group .....	215
\showglouseriv: added		\compatibleglossentry: changed	
\glsdetoklabel .....	267	\newcommand to \def as is may or	
dua: fixed bug in \acrfullfmt .....	241	may not be defined .....	354
fixed bug in \Acrfullplfmt .....	241	\compatiblesubglossentry: changed	
fixed bug in \acrfullplfmt .....	241	\newcommand to \def as is may or	
		may not be defined .....	355

\defglsdisplayfirst: fixed unwanted space .....	114	\Acrfullplfmt: fixed no case change bug .....	231
\glo@grabfirst: new .....	210	\glsletentryfield: new .....	156
\gls@defglossaryentry: replaced \ifx with \ifdefvoid .....	93	4.08 (2014-07-30)	
\glsnoidxdisplayloc: new .....	212	\@ACRlong: added	
\glsnoidxdisplaylocclishandler: new .....	212	\do@gls@link@checkfirsthyper .....	379
\glsnoidxloclist: new .....	212	\@ACRshort: added	
\glsnoidxlocclishandler: new ....	212	\do@gls@link@checkfirsthyper .....	378
\glsnoidxstripaccents: new .....	23	\@Acrlong: added	
alttree: moved hangindent and parindent assignments outside level test .....	331	\do@gls@link@checkfirsthyper .....	378
\makeglossaries: Moved definition of \glswrite to \makeglossaries ..	182	\@Acrshort: added	
\makenoidxglossaries: new .....	184	\do@gls@link@checkfirsthyper .....	377
\printglossary: changed to use new \@printglossary .....	200	\@GLS@: moved \glsifhyper .....	132
\printnoidxglossaries: new .....	200	moved check for first use to	
\printnoidxglossary: new .....	200	\@gls@link .....	132
\showgloclolist: new .....	269	\@GLSpl: moved \glsifhyper .....	135
\warn@noprintglossary: Activate warning in \makeglossaries ....	199	moved check for first use to	
\writeist: checked for definition of \glswrite .....	169, 173	\@gls@link .....	135
4.06 (2014-03-12)		\@GLS@: moved \glsifhyper .....	131
\@GLS@: added \glsifhyper .....	132	moved check for first use to	
\@GLSpl: added \glsifhyper .....	135	\@gls@link .....	131
\@Gls@: added \glsifhyper .....	131	\@GLspl@: moved \glsifhyper .....	134
\@Glspl@: added \glsifhyper .....	134	moved check for first use to	
\@gls@: added \glsifhyper .....	131	\@gls@link .....	134
\@gls@numbersdef: added hook to set toc title .....	36	\@acrlong: added	
\@gls@symbolsdef: added hook to set toc title .....	35	\do@gls@link@checkfirsthyper .....	378
\@glsdisp: added \glsifhyper .....	135	\@acrshort: added	
\@glspl@: added \glsifhyper .....	133	\do@gls@link@checkfirsthyper .....	377
General: added \glsifhyper ....	149–156	\@closegls: new .....	179
acronym: added hook to set toc title ....	17	\@gls@: moved \glsifhyper .....	131
acronyms: added hook to set toc title ...	17	moved check for first use to	
\glsdefmain: added hook to set toc title	16	\@gls@link .....	131
4.07 (2014-04-04)		\@gls@automake: new .....	179
\@glossarysection: added optional argument when using unstarred version .....	47	\@gls@doautomake: new .....	33
\@gls@noidx@do: added \global in case it's used in a tabular-like style ....	210	\@gls@field@link: added assignment of	
		\do@gls@link@checkfirsthyper .....	136
		\@gls@forbidtexext: new .....	65
		\@gls@hyp@opt: new .....	116
		\@gls@link: removed redundancy ....	118
		renamed \gls@type to \glstype ...	118
		\@gls@link@checkfirsthyper: new .	117
		\@glsdisp: moved \glsifhyper .....	135
		moved check for first use to	
		\@gls@link .....	135
		\@glspl@: moved \glsifhyper .....	133
		moved check for first use to	
		\@gls@link .....	133
		\@ignored@glossaries: new .....	68

General: added entrycounter option to		removed \@sGLStext .....	136
printgloss family .....	214	removed \@sGlstext .....	137
added nopostdot option to		removed \@sglstext .....	136
printgloss family .....	214	removed \@sGLSuseriii .....	146
added subentrycounter option to		removed \@sGlsuseriii .....	145
printgloss family .....	214	removed \@sglsuseriii .....	145
explicitly initialise hyper key .....	115	removed \@sGLSuserii .....	145
moved \glsifhyper .....	149–156	removed \@sGlsuserii .....	145
removed \@sACRlongpl .....	155	removed \@sglsuserii .....	144
removed \@sAcrlongpl .....	155	removed \@sGLSuseriv .....	147
removed \@sacrlongpl .....	154	removed \@sGlsuseriv .....	146
removed \@sACRlong .....	153	removed \@sglsuseriv .....	146
removed \@sAcrlong .....	153	removed \@sGLSuseri .....	144
removed \@sacrlong .....	152	removed \@sGlsuseri .....	144
removed \@sACRshortpl .....	151	removed \@sglsuseri .....	144
removed \@sAcrshortpl .....	151	removed \@sGLSsuservi .....	148
removed \@sacrshortpl .....	150	removed \@sGlsuservi .....	148
removed \@sACRshort .....	150	removed \@sglsuservi .....	148
removed \@sAcrshort .....	149	removed \@sGLSsuserv .....	147
removed \@sacrshort .....	148	removed \@sGlsuserv .....	147
removed \@sgls@link .....	117	removed \@sglsuserv .....	147
removed \@sGLSdescplural .....	142	removed \@sGLS .....	132
removed \@sGlsdescplural .....	141	removed \@sGls .....	131
removed \@sglsdescplural .....	141	removed \@sgls .....	130
removed \@sGLSdesc .....	141	removed \@thirdofthree (defined in	
removed \@sGlsdesc .....	141	kernel) .....	130
removed \@sglsdesc .....	140	removed sPGLS .....	277
removed \@sglsdisp .....	135	removed sPgls .....	276
removed \@sGLSfirstplural .....	139	removed spgls .....	274
removed \@sGlsfirstplural .....	139	removed sPGLSpl .....	278
removed \@sglsfirstplural .....	139	removed sPglspl .....	276
removed \@sGLSfirst .....	138	removed spglspl .....	275
removed \@sGlsfirst .....	137	\ACRfull: removed \s@ACRfull .....	230
removed \@sglsfirst .....	137	switched to using \@gls@hyp@opt ..	230
removed \@sGLSname .....	140	\Acrfull: removed \@sAcrfull .....	229
removed \@sGlsname .....	140	switched to using \@gls@hyp@opt ..	229
removed \@sglsname .....	139	\acrfull: removed \@sacrfull .....	228
removed \@sGLSplural .....	138	switched to using \@gls@hyp@opt ..	228
removed \@sGlsplural .....	138	\ACRfullpl: removed \s@ACRfullpl ..	231
removed \@sglsplural .....	138	switched to using \@gls@hyp@opt ..	231
removed \@sGLSpl .....	134	\Acrfullpl: removed \s@Acrfullpl ..	230
removed \@sGlspl .....	133	switched to using \@gls@hyp@opt ..	230
removed \@sglspl .....	133	\acrfullpl: removed \s@acrfullpl ..	230
removed \@sGLSsymbolplural ....	143	switched to using \@gls@hyp@opt ..	230
removed \@sGlsymbolplural ....	143	\ACRlong: switched to using	
removed \@sglsymbolplural ....	143	\@gls@hyp@opt .....	153
removed \@sGLSsymbol .....	142	\Acrlong: switched to using	
removed \@sGlsymbol .....	142	\@gls@hyp@opt .....	152
removed \@sglsymbol .....	142		

\acrlong: switched to using \@gls@hyp@opt .....	152	\glsdohyperlink: new .....	129
\ACRlongpl: switched to using \@gls@hyp@opt .....	155	\glsdohypertarget: new .....	129
\Acrlongpl: switched to using \@gls@hyp@opt .....	154	\glsenablehyper: added \KV@glslink@hypertrue to definition .....	130
\acrlongpl: switched to using \@gls@hyp@opt .....	154	\GLSfirst: switched to using \@gls@hyp@opt .....	137
\ACRshort: switched to using \@gls@hyp@opt .....	149	\Glsfirst: switched to using \@gls@hyp@opt .....	137
\Acrshort: switched to using \@gls@hyp@opt .....	149	\glsfirst: switched to using \@gls@hyp@opt .....	137
\acrshort: switched to using \@gls@hyp@opt .....	148	\GLSfirstplural: switched to using \@gls@hyp@opt .....	139
\ACRshortpl: switched to using \@gls@hyp@opt .....	151	\Glsfirstplural: switched to using \@gls@hyp@opt .....	139
\Acrshortpl: switched to using \@gls@hyp@opt .....	151	\glsfirstplural: switched to using \@gls@hyp@opt .....	139
\acrshortpl: switched to using \@gls@hyp@opt .....	150	\glsifhyper: deprecated .....	116
\forallacronyms: new .....	58	\glslink: switched to using \@gls@hyp@opt .....	117
\GLS: switched to using \@gls@hyp@opt	132	\glslinkcheckfirsthyperhook: new	118
\Gls: switched to using \@gls@hyp@opt	131	\glslinkvar: new .....	116
\gls: switched to using \@gls@hyp@opt	130	\GLSname: switched to using \@gls@hyp@opt .....	140
\glsdefglossaryentry: added check for ignored glossary .....	89	\Glsname: switched to using \@gls@hyp@opt .....	140
\gls@istfilebase: new .....	43	\glsname: switched to using \@gls@hyp@opt .....	139
\glsaddkey: removed \@sGLS@user@<key> .....	83	\GLSpl: switched to using \@gls@hyp@opt .....	134
removed \@sGls@user@<key> .....	83	\Glspl: switched to using \@gls@hyp@opt .....	133
removed \@sgls@user@<key> .....	82	\glspl: switched to using \@gls@hyp@opt .....	133
switched to using \@gls@hyp@opt	82, 83	\GLSplural: switched to using \@gls@hyp@opt .....	138
\GLSdesc: switched to using \@gls@hyp@opt .....	141	\Glsplural: switched to using \@gls@hyp@opt .....	138
\Glsdesc: switched to using \@gls@hyp@opt .....	140	\glsplural: switched to using \@gls@hyp@opt .....	138
\glsdesc: switched to using \@gls@hyp@opt .....	140	\glsspace: new .....	229
\GLSdescplural: switched to using \@gls@hyp@opt .....	142	\GLSsymbol: switched to using \@gls@hyp@opt .....	142
\Glsdescplural: switched to using \@gls@hyp@opt .....	141	\Glsymbol: switched to using \@gls@hyp@opt .....	142
\glsdescplural: switched to using \@gls@hyp@opt .....	141	\glssymbol: switched to using \@gls@hyp@opt .....	142
\glsdisablehyper: added \KV@glslink@hyperfalse to definition .....	129	\GLSsymbolplural: switched to using \@gls@hyp@opt .....	143
\glsdisp: switched to using \@gls@hyp@opt .....	135		



\Glsymbolplural: switched to using \@gls@hyp@opt .....	143	altlongragged4col: fixed bug that displayed description instead of symbol .....	304
\glssymbolplural: switched to using \@gls@hyp@opt .....	143	\newglossary: added starred version ..	66
\GLStext: switched to using \@gls@hyp@opt .....	136	\newignoredglossary: new .....	68
\Glstext: switched to using \@gls@hyp@opt .....	137	\ns@newglossary: added \@glotype@<name>@log .....	66
\glstext: switched to using \@gls@hyp@opt .....	136	new .....	66
\glstreenamefmt: new .....	324	\p@gls@hyp@opt: new .....	116
\GLSuseri: switched to using \@gls@hyp@opt .....	144	\PGLS: changed to use \@gls@hyp@opt	277
\Glsuseri: switched to using \@gls@hyp@opt .....	144	\Pgls: changed to use \@gls@hyp@opt	276
\glsuseri: switched to using \@gls@hyp@opt .....	144	\pgls: changed to use \@gls@hyp@opt	274
\GLSuserii: switched to using \@gls@hyp@opt .....	145	\PGLSpl: changed to use \@gls@hyp@opt .....	278
\Glsuserii: switched to using \@gls@hyp@opt .....	144	\Pglspl: changed to use \@gls@hyp@opt .....	276
\glsuserii: switched to using \@gls@hyp@opt .....	144	\pglspl: changed to use \@gls@hyp@opt .....	275
\GLSuseriii: switched to using \@gls@hyp@opt .....	146	\s@gls@hyp@opt: new .....	116
\Glsuseriii: switched to using \@gls@hyp@opt .....	145	\s@newglossary: new .....	66
\glsuseriii: switched to using \@gls@hyp@opt .....	145	automake: new .....	32
\GLSuseriv: switched to using \@gls@hyp@opt .....	146	4.09 (2014-08-12)	
\Glsuseriv: switched to using \@gls@hyp@opt .....	146	\glsaddkey: fixed bug in user commands	82
\glsuseriv: switched to using \@gls@hyp@opt .....	146	4.10 (2014-08-27)	
\GLSuserv: switched to using \@gls@hyp@opt .....	147	\@Gls@acentryname: new .....	157
\Glsuserv: switched to using \@gls@hyp@opt .....	147	\@Gls@entryname: new .....	157
\GLSuservi: switched to using \@gls@hyp@opt .....	148	\@gls@glossary: Renamed \@glossary to \@gls@glossary .....	189
\Glsuservi: switched to using \@gls@hyp@opt .....	148	\glspercentchar: new .....	167
\ifignoredglossary: new .....	68	\glstildechar: new .....	167
		alttree: moved space after symbol	331, 332
		4.11 (2014-09-01)	
		@@do@esc@wrglossary: added hook ..	194
		sanitize: none option .....	25
		\gls@wrglossary: renamed from \@wrglossary to \gls@wrglossary	190
		\glsaddprotectedpagefmt: new ....	191
		\glsbackslash: new .....	167
		4.12 (2014-11-22)	
		\@gls@addpredefinedattributes: Added glsignore attribute .....	52
		\@gls@adjustmode: new .....	166
		\@gls@nottranslatorhook: removed ...	26
		\@gls@toc: added \protect to \numberline .....	48
		\@gls@usetranslator: new .....	26
		\glsacrpluralsuffix: new .....	40
		\glsadd: added check for vertical mode	165

\glsaddallunused: replaced @gobble with glsignore .....	166	\glsunset: switched to \@glsunset ...	97
\glsifusedtranslatordict: new ....	26	4.15 (2015-03-16)	
\glsignore: new .....	166	General: bug fix replaced \@glo@type with \glstype .....	155
\glsupacrpluralsuffix: new .....	40	4.16 (2015-06-18)	
\ProvidesGlossariesLang: new .....	40	\glsaddstoragekey: new .....	81
\RequireGlossariesLang: new .....	40	4.16 (2015-07-08)	
4.13 (2015-02-03)		\@ACRlong: added \glspostlinkhook	379
\indexspace: new .....	285, 306, 324	\@ACRshort: added \glspostlinkhook	378
4.14 (2015-02-28)		\@Acrlong: added \glspostlinkhook	379
\@@glslocalreset: new .....	98	\@Acrshort: added \glspostlinkhook	378
\@@glslocalunset: new .....	98	\@GLS@: added \glspostlinkhook ...	132
\@@glsreset: new .....	98	\@GLSpl: added \glspostlinkhook ..	135
\@@glsunset: new .....	98	\@Gls@: added \glspostlinkhook ...	132
\@@newglossaryentry@defcounters: new .....	99	\@Glspl@: added \glspostlinkhook .	134
\@cGls: new .....	102	\@acrlong: added \glspostlinkhook	378
\@cGls@: new .....	103	\@acrshort: added \glspostlinkhook	377
\@cGlspl@: new .....	104	\@gls@: added \glspostlinkhook ...	131
\@cgls: new .....	102	\@gls@@link: added \glspostlinkhook .....	117
\@cgls@: new .....	102	\@gls@field@link: added \glspostlinkhook .....	136
\@cglspl: new .....	103	\@gls@link: moved definition of \glsifhyperon outside of this macro .....	119
\@cglspl@: new .....	103	\@glsdisp: added \glspostlinkhook	136
\@gls@entry@count: new .....	102	\@glspl@: added \glspostlinkhook .	133
\@gls@increment@currcount: new ..	101	General: added \glspostlinkhook	149–156
\@gls@local@increment@currcount: new .....	101	\glsacspace: new .....	236
\@gls@write@entrycounts: new ....	102	\glsadd: changed \@do@wrglossary to \@do@wrglossary .....	166
\@glslocalreset: new .....	98	\glsfielddef: new .....	85
\@glslocalunset: new .....	97	\glsfieldedef: new .....	84
\@glsreset: new .....	98	\glsfieldfetch: new .....	85
\@glsunset: new .....	98	\glsfieldgdef: new .....	84
\@newglossaryentry@defcounters: new .....	94	\glsfieldxdef: new .....	84
\@cGls: new .....	102	\glsifhyperon: moved definition of \glsifhyperon .....	118
\@cgls: new .....	102	\glslinkpostsetkeys: new .....	118
\@cGlsformat: new .....	103	\glspostlinkhook: new .....	117
\@cglsformat: new .....	102	\glswriteentry: new .....	190
\@cGlspl: new .....	103	\ifglsfieldcseq: new .....	87
\@cglspl: new .....	103	\ifglsfielddefeq: new .....	86
\@cGlsplformat: new .....	104	\ifglsfieldeq: new .....	86
\@cglsplformat: new .....	103	long-sp-short: new .....	235
\@gls@defdocnewglossaryentry: new .	76	\showglofield: new .....	269
\@glsenableentrycount: new .....	99	4.18 (2015-09-09)	
\@glslocalreset: switched to \@glslocalreset .....	97	General: split mfirstuc into separate bundle .....	3
\@glslocalunset: switched to \@glslocalunset .....	97		
\@glsreset: switched to \@glsreset ...	96		

4.19 (2015-10-31)		\gls@arabicpage: new .....	191
\glstreenamebox: new .....	330	\gls@protected@pagefmts: added	
4.19 (2015-11-22)		arabic to list .....	191
\@gls@link@nocheckfirsthyper: new	136	\glstrytitlecase: new .....	160
\@gls@preglossaryhook: new .....	200	\glsfindwidesttoplevelname: new	330
\@printglossary: added		\glslistgroupheaderfmt: new .....	286
\@gls@preglossaryhook .....	202	\glslistnavigationitem: new .....	286
\do@glsglisablehyperinlist: new	118	\glstreegroupheaderfmt: new .....	324
\doifglossarynoexistsordo: new	61	\glstreenavigationfmt: new .....	324
\gls@gobbleopt: new .....	65	\ifglswrallowprimitivemods: new	192
\glsdoifexistsordo: new .....	60	list: fixed missing space before	
4.20 (2015-11-30)		description .....	287
\@gls@link: added		long: fixed typo in \glossentrydesc	290
\@gls@setdefault@glslink@opts	118	super4col: fixed bug in \glossentry	315
added \glsdonohyperlink when		4.23 (2016-04-30)	
hyperlink is suppressed .....	119	\glscurrentfieldvalue: new .....	64
\@gls@setdefault@glslink@opts:		\ifglshasfield: added	
new .....	118	\glscurrentfieldvalue .....	63, 64
\gls@checkseeallowed@preambleonly:		altlongragged4col: check for	
new .....	71	nogroupskip changed .....	304
\glsdonohyperlink: new .....	129	altsuperragged4col: check for	
4.21 (2016-01-24)		nogroupskip changed .....	323
\@printglossary: warn if no style has		long: check for nogroupskip changed	290
been set .....	201	long-booktabs: check for nogroupskip	
General: changed checkfirsthyper		changed .....	296
assignment .....	149–155	long3col: check for nogroupskip	
\glossarystyle: set default style if not		changed .....	292
already set .....	223	long3col-booktabs: check for	
\glsltpenaltycheck: new .....	299	nogroupskip changed .....	297
\glspatchLToutput: new .....	299	long4col: check for nogroupskip	
\glspenaltygroupskip: new .....	299	changed .....	293
altlong4col-booktabs: new .....	298	long4col-booktabs: check for	
altlongragged4col-booktabs: new	299	nogroupskip changed .....	297
long-booktabs: new .....	296	longragged: check for nogroupskip	
long3col-booktabs: new .....	296	changed .....	301
long4col-booktabs: new .....	297	longragged3col: check for nogroupskip	
longragged-booktabs: new .....	298	changed .....	303
longragged3col-booktabs: new	298	super: check for nogroupskip changed	312
\setglossarystyle: set default style if		super3col: check for nogroupskip	
not already set .....	223	changed .....	314
4.22 (2016-04-19)		super4col: check for nogroupskip	
\@do@esc@wrglossary: added check		changed .....	316
for \@arabic .....	194	superragged: check for nogroupskip	
added test to allow temporary primitive		changed .....	319
modifications and added arabic case	194	superragged3col: check for	
mcolalttreespannav: new .....	311	nogroupskip changed .....	321
mcolindexspannav: new .....	307	4.24 (2016-05-27)	
mcoltreenonamespannav: new .....	309	\@gls@extramakeindexopts: new	178
mcoltreespannav: new .....	308		

\@gls@glossary: added check for debug mode .....	189	\@gls@setupsort@none: new .....	15
\@gls@see@noindex: new .....	6	\@xdycrossrefhook: new .....	54
debug: new .....	4	\@xdylocationclassorder: bug fix: changed \edef to \def .....	55
seenoinindex: new .....	7	\glosortentrieswarning: new .....	20
\glsnomakeindexwarning: new .....	49	\gls@set@xr@key: new .....	71
\GlsSetQuote: new .....	174	\gls@xr@key: new .....	71
\GlsSetWriteIstHook: new .....	174	\GlsAddXdyLocation: bug fix: changed #1 to #2 .....	54
4.25 (2016-06-09)		\glsnoidxstripaccents: added \a ...	23
\@gls@enablesavenonumberlist: new	72	added \TH, \dh and \DH .....	24
\@gls@initnonumberlist: new .....	72	4.31 (2017-08-10)	
\@gls@savenonumberlist: new .....	72	nolist: added check for “list” style .....	10
4.25 (??)		4.31 (2017-09-10)	
General: changed		style: changed \renewcommand to \def .	8
\DeclareRobustCommand to		4.32 (2017-08-24)	
\newrobustcmd and changed		\@glsnavhypertarget: new .....	280
\@ifundefined to \ifcsundef ...	367	\@glsshowtarget: new .....	6
4.26 (2016-10-12)		\glsshowtarget: new .....	5
\@glossary@default@style: added		4.33 (2017-09-20)	
check for classicthesis .....	8	\@do@esc@wrglossary: added	
mcolindex: replaced \@idxitem with		\gls@the and \gls@number .....	194
\glstreeitem .....	306	renamed from	
mcolindexspannav: replaced \@idxitem		\@do@esc@wrglossary .....	193
with \glstreeitem .....	307	\@do@noesc@wrglossary: new .....	192
\glstreechildpredesc: new .....	325	\@do@wrglossary: changed to check	
\glstreeitem: new .....	325	for esclocations .....	192
\glstreepredesc: new .....	325	\@gls@missinglang@warn: new .....	20
\glstreesubitem: new .....	325	\GlsSetXdyFirstLetterAfterDigits:	
\glstreesubsubitem: new .....	325	added starred version .....	167
4.28 (2017-01-07)		\GlsSetXdyNumberGroupOrder: new .	168
\glspatchtabularx: new .....	96	esclocations: new .....	9
4.29 (2017-01-19)		4.34 (2017-11-03)	
\@gls@noidx@do: current letter group		mcolalttreespannav: removed spurious	
assignment made global .....	211	space .....	311
\@print@noidx@glossary: moved		\glsshowtarget: modified to check for	
definition of		math mode and inner .....	5
\@gls@currentlettergroup outside		4.35 (2017-11-14)	
of the glossary environment .....	209	\glsadd: added \@gls@setsort (in case	
General: added check for		of sort=use) .....	166
\@glsxtr@doaccsupp .....	354	4.36 (2018-03-07)	
\glsnavhyperlinkname: new .....	279	\@gls@glossary: removed \index ...	189
4.30 (2017-06-11)		4.37 (2018-04-07)	
\@glo@autosee: new .....	93	\gls@begindocdefs: new .....	78
\@glo@autoseehook: new .....	93	4.38 (2018-05-10)	
\@glo@check@sortallowed: new .....	13	\@gls@define@glossaryentrycounter:	
\@gls@noidx@do: letter group		added check for existence of	
assignment made global .....	211	glossaryentry counter .....	11
\@gls@setupsort@def: added check for		new .....	11
register .....	14		

\@gls@define@glossarysubentrycounter: new ..... 12 prepended \currentglossary. to \theHglossarysubentry and removed spurious eol space ..... 12	nonumberlist: changed \val and \nr to \gls@nonumberlist@val and \gls@nonumberlist@nr ..... 71
\glsaccsupp: added braces around actual text argument ..... 364	translate: changed \val and \nr to \gls@translate@val and \gls@translate@nr ..... 27
\glsentrycounterlabel: bug fix: move conditional inside command ..... 217	numberedsection: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr .... 8
\GlsEntryCounterLabelPrefix: new 214	4.42 (2019-01-06)
\glsentryitem: bug fix: move conditional inside command ..... 217	\@gls@@automake@immediate: new .. 181
\glsrefentry: bug fix: move conditional inside command ..... 217	\@gls@automake@immediate: new ... 180
\glsresetsubentrycounter: bug fix: move conditional inside command . 216	\gls@automake@nr: new ..... 32
\glsstepentry: bug fix: move conditional inside command ..... 216	\glsfieldedef: changed from \edef to \protected@csedef ..... 84
\glsstepsubentry: bug fix: move conditional inside command ..... 216	\glsfieldxdef: changed from \edef to \protected@csxdef ..... 84
\glsesubentrycounterlabel: bug fix: move conditional inside command . 217	\ifglsautomake: now defined explicitly instead of through boolean key ..... 32
\glsesubentryitem: bug fix: move conditional inside command ..... 217	noglossaryindex: new ..... 36
\showglongnameaccess: bug fix: corrected field (was showing text access field) 400	automake: switch from boolean to choice 32
4.40 (2018-06-01)	4.42 (??)
\istfile: changed \def to \providecommand ..... 188	altlong4col-booktabs: removed superfluous \glspatchLToutput . 298
\makenoidxglossaries: false ..... 184	4.43 (2019-09-28)
4.41 (2018-07-23)	\glsnoidxstripaccents: add check for LaTeX version 2019/10/01 ..... 24
\@gls@override@glossary: new ..... 34	4.44 (2019-12-06)
General: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr ..... 214	\@glsprefix@record@hook: new .... 274
debug: changed \val and \nr to \gls@debug@val and \gls@debug@nr ..... 4	4.45 (2020-02-13)
seenoindex: changed \val and \nr to \gls@seenoindex@val and \gls@seenoindex@nr ..... 7	\@do@write@glslabels: new ..... 31
kernelglossredefs: new ..... 34	\@glsshows@target: new ..... 6
\glossary: added warning ..... 34	\@GLSdesc@: added accessibility support 381
\gls@original@glossary: new ..... 33	\@GLSdescplural@: added accessibility support ..... 382
\gls@original@makeglossary: new .. 33	\@GLSfirst@: added accessibility support ..... 380
\makeglossaries: removed redefinition of \makeglossary ..... 182	\@GLSfirstplural@: added accessibility support ..... 380
\makeglossary: added warning ..... 34	\@GLSname@: added accessibility support 381
	\@GLSplural@: added accessibility support ..... 380
	\@GLSsymbol@: added accessibility support ..... 382
	\@GLSsymbolplural@: added accessibility support ..... 382
	\@GLStext@: added accessibility support 379
	\@GLSuseri@: added accessibility support ..... 383

\@GLSuserii@: added accessibility support .....	383	\@glsname@: added accessibility support	381
\@GLSuseriii@: added accessibility support .....	383	\@glsplural@: added accessibility support .....	380
\@GLSuseriv@: added accessibility support .....	384	\@glsshowaccsupp: new .....	6
\@GLSuserv@: added accessibility support .....	384	\@glssymbol@: added accessibility support .....	382
\@GLSuservi@: added accessibility support .....	384	\@glssymbolplural@: added accessibility support .....	382
\@Gls@acentryname: added check for \glsshortaccessdisplay .....	158	\@glstext@: added accessibility support	379
\@Glsdesc@: added accessibility support	381	\@glsuseri@: added accessibility support .....	382
\@Glsdescplural@: added accessibility support .....	381	\@glsuserii@: added accessibility support .....	383
\@Glsfirst@: added accessibility support .....	380	\@glsuseriii@: added accessibility support .....	383
\@Glsfirstplural@: added accessibility support .....	380	\@glsuseriv@: added accessibility support .....	384
\@Glsname@: added accessibility support	381	\@glsuserv@: added accessibility support .....	384
\@Glsplural@: added accessibility support .....	380	\@glsuservi@: added accessibility support .....	384
\@Glsymbol@: added accessibility support .....	382	General: removed backward compatibility use of symbol key .....	358
\@Glsymbolplural@: added accessibility support .....	382	acronyms: changed \renewcommand to \def .....	17
\@Glstext@: added accessibility support	379	debug: showaccsupp .....	4
\@Glsuseri@: added accessibility support .....	383	restoremakeloss: new .....	31
\@Glsuserii@: added accessibility support .....	383	\@gls@accessibility: new .....	355
\@Glsuseriii@: added accessibility support .....	383	\@gls@accsupp@engine: new .....	355
\@Glsuseriv@: added accessibility support .....	384	\@gls@accessibility: new .....	355
\@Glsuserv@: added accessibility support .....	384	\@glsdefaultshortaccess: new .....	395
\@Glsuservi@: added accessibility support .....	384	\@glsdescriptionaccessdisplay: added check for existence .....	365
\@do@write@glslabels: new .....	31	\@glsdescriptionpluralaccessdisplay: added check for existence .....	365
\@domakeglossaries: new .....	30	\@glsentrydescpluralaccess: corrected field reference .....	361
\@gls@fieldaccess@display: new ..	364	\@glsentryfirstpluralaccess: switched to using \@gls@entry@field .....	361
\@glsdesc@: added accessibility support	381	\@glsentryparent: new .....	161
\@glsdescplural@: added accessibility support .....	381	\@glsentryprefix: added \@glsdetoklabel .....	273
\@glsfirst@: added accessibility support .....	380	\@glsentryprefix: added \@glsdetoklabel .....	273
\@glsfirstplural@: added accessibility support .....	380	\@glsentryprefixfirst: added \@glsdetoklabel .....	273
		\@glsentryprefixfirst: added \@glsdetoklabel .....	273

\Glsentryprefixfirstplural: added		\glssymbolpluralaccessdisplay:	
\glstoklabel ..... 273		added check for existence ..... 365	
\Glsentryprefixfirstplural: added		\glstextaccessdisplay: added check	
\glstoklabel ..... 273		for existence ..... 364	
\Glsentryprefixplural: added		\glsuseriaccessdisplay: new ..... 366	
\glstoklabel ..... 273		\glsuseriiaccessdisplay: new .... 366	
\Glsentryprefixplural: added		\glsuseriiiaccessdisplay: new ... 366	
\glstoklabel ..... 273		\glsuserivaccessdisplay: new .... 367	
\Glsentrytitlecase: added existence		\glsuservaccessdisplay: new ..... 367	
check ..... 160		\glsuserviaccessdisplay: new .... 367	
\Glsentryuseriaccess: new ..... 362		\ifglshaschildren: made robust ..... 61	
\Glsentryuseriiaccess: new ..... 362		\ifglshasfield: made robust ..... 63	
\Glsentryuseriiiaccess: new ..... 362		\ifglshaslong: made robust ..... 62	
\Glsentryuserivaccess: new ..... 362		\ifglshasprefix: added	
\Glsentryuservaccess: new ..... 362		\glstoklabel ..... 274	
\Glsentryuserviaccess: new ..... 362		\ifglshasprefixfirst: added	
\glstoklabel ..... 273		\glstoklabel ..... 274	
\Glsfieldaccsupp: new ..... 363		\ifglshasprefixfirstplural: added	
\Glsfirstaccessdisplay: added check		\glstoklabel ..... 274	
for existence ..... 365		\ifglshasprefixplural: added	
\Glsfirstpluralaccessdisplay:		\glstoklabel ..... 274	
added check for existence ..... 365		\ifglshashshort: made robust ..... 62	
\Glslongaccessdisplay: added check		\ifglshassymbol: made robust ..... 62	
for existence ..... 366		disablemakegloss: new ..... 30	
\Glslongpluralaccessdisplay: added		\makeglossaries: let \@makeglossary	
check for existence ..... 366		to \@gobble instead of \relax .... 182	
\Glsnameaccessdisplay: added check		writetoklabels: new ..... 32	
for existence ..... 364		user1access: new ..... 357	
\Glspluralaccessdisplay: added		user2access: new ..... 357	
check for existence ..... 365		user3access: new ..... 357	
\Glsprefixsep: new ..... 274		user4access: new ..... 358	
\Glssee: switched to \newrobustcmd . 197		user5access: new ..... 358	
\Glsseeformat: switched to		user6access: new ..... 358	
\newrobustcmd ..... 197		\xglstoklabel ..... 363	
\Glsseelitem: switched to		4.46 (2020-03-19)	
\newrobustcmd ..... 198		\@ifglossaryexists: new ..... 59	
\Glsseelist: switched to		\@printgloss@checkexists: new ... 199	
\newrobustcmd ..... 197		\@printgloss@checkexists@allowignored:	
\Glsshortaccessdisplay: added check		new ..... 199	
for existence ..... 366		\@printgloss@checkexists@noignored:	
\Glsshortaccsupp: new ..... 363		new ..... 199	
\Glsshortplaccsupp: new ..... 364		\@printglossary: replaced	
\Glsshortpluralaccessdisplay:		\@ifglossaryexists with	
added check for existence ..... 366		\@printgloss@checkexists .... 201	
\Glsshowaccsupp: new ..... 6		\doifglossarynoexistsordo: switched	
\Glsshowtargetfont: new ..... 6		to starred form of	
\Glsshowtargetouter: new ..... 5		\@ifglossaryexists ..... 61	
\Glsshowtargetsymbol: new ..... 5		\ifglossaryexists: added starred form 59	
\Glsymbolaccessdisplay: added		\s@ifglossaryexists: new ..... 59	
check for existence ..... 365			

\setglossarypreamble: switched to starred form of \ifglossaryexists	45	\glsdefs@newdocentry: new	79
4.47 (2021-09-20)		\glsdisplaynumberlist: changed \edef to \protected@edef	164
\@do@esc@wrglossary: changed \edef to \protected@edef	195	\glsfielddef: changed \edef to \protected@edef	85
\@do@noesc@wrglossary: changed \edef to \protected@edef	192	\glsfieldedef: changed \edef to \protected@edef	84
\@do@write@glslabels: fully expand list	31	\glsfieldfetch: changed \edef to \protected@edef	85
\@gls@@default@acr@descplural: new	248	\glsfieldgdef: changed \edef to \protected@edef	84
\@gls@@default@acr@firstplural: new	248	\glsfieldxdef: changed \edef to \protected@edef	84
\@gls@@default@acr@plural: new	247	\glsIfListOfAcronyms: changed \edef to \protected@edef	18
\@gls@noexpand@assign@default@plurals: new	247	\glsmoveentry: changed \edef to \protected@edef	94
\@glo@do@sortentries: changed \edef to \protected@edef	205	\glsnavhyperlink: changed \edef to \protected@edef	279
\@glo@sortentries: changed \edef to \protected@edef	204	\GlsSetQuote: added check for backslash	176
\@glo@storeentry: changed \edef to \protected@edef	95	\glsstepsubentry: changed \edef to \protected@edef	216
\@gls@checkesc: new	127	\ifglsfieldcseq: changed \edef to \protected@edef	87
\@gls@deprecated@acr@warning: new	262	\ifglshaschildren: changed \edef to \protected@edef	61
\@gls@link: changed \edef to \protected@edef	118	\ifignoredglossary: changed \edef to \protected@edef	68
\@gls@setupsort@use: changed \edef to \protected@edef	15	\makenoidxglossaries: changed \edef to \protected@edef	184
\@gls@writedef: added a warning if entry has already been defined	79	\new@glossaryentry: allow overwrite if defined in .glsdefs	77
\@write@glslabelsname: new	32	\newignoredglossary: changed \edef to \protected@edef	68
General: changed \x to \@gls@x	226	\ns@newglossary: changed \edef to \protected@edef	66
\do@write@glslabelsname: new	32	writeglslabelnames: new	32
\forglsentries: changed \edef to \protected@edef	58	\SetDescriptionDUAAcronymStyle: added deprecated warning	252
\gls@defdocnewglossaryentry: removed see warning	76	\SetDescriptionFootnoteAcronymStyle: added deprecated warnings	250
\gls@defglossaryentry: changed \edef to \protected@edef	88, 89	\SetDUASyle: added deprecated warnings	262
\gls@docdef@addlabel: new	77	\SetFootnoteAcronymStyle: added deprecated warnings	257
\gls@docdef@labellist: new	77	\SetSmallAcronymStyle: added deprecated warnings	260
\gls@glsdefs@addlabel: new	77		
\gls@glsdefs@labellist: new	77		
\gls@save@numberlist: changed \edef to \protected@edef	199		
\glsadd: changed \edef to \protected@edef	166		
\glsaddall: changed \edef to \protected@edef	166		



4.47 (??)		\glslistexpandedname:new .....	286
\@makeglossaries@warn@noprintglossary:		\glslistinit:new .....	286
new .....	183	\glsunexpandedfieldvalue:new ....	85
4.48 (2021-10-19)		4.49 (2021-11-01)	
\glscapitalisewords:new .....	160	\glslistinit:added\glsnavigation	286

# 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	
\!	122, 125, 176, 480
\"	23, 122–125, 127, 128, 416, 477–480, 482
\#	127, 171, 506
\%	167, 173, 338, 339, 503, 507, 508
\&	40, 165, 426, 502
\'	23, 416
\.	11, 23, 408, 416
\=	23, 416
\?	122, 124, 176, 477, 479, 510
\@	78, 449
\@@delimN	225, 541
\@do@wrglossary	184, 192, 195, 516, 521, 522
\@do@esc@wrglossary	192, 521
\@do@noesc@wrglossary	192, 521
\@do@wrglossary	166, 190, 502, 520, 521
\@do@write@glslabels	31, 422
\@glo@assign@sortkey	185, 516, 535
\@glo@list	58, 438
\@glo@sort	23, 415
\@glo@type	200, 525
\@glossarysec	7, 47, 48, 407, 430, 431
\@glossaryseclabel	8, 47, 48, 214, 407, 430, 431, 534
\@glossarysecstar	8, 47, 214, 407, 430, 431, 534
\@gls@@default@acr@descplural	247
\@gls@@default@acr@firstplural	247
\@gls@@default@acr@plural	247
\@gls@checkactual	126, 127, 481, 482
\@gls@checkbar	125, 126, 480, 481
\@gls@checkescactual	124, 479
\@gls@checkescbar	124, 125, 479, 480
\@gls@checkesclevel	125, 480
\@gls@checkescquote	123, 177, 478, 479, 510, 511
\@gls@checklevel	126, 481
\@gls@checkquote	123, 175, 478, 509
\@gls@default@entryfmt	105, 114, 466, 472, 473
\@gls@expand@field	21, 75, 76, 81, 82, 247, 250, 252, 254, 256, 257, 259–262, 396–400, 414, 447, 448, 451, 555, 557, 558, 560, 562, 564, 566
\@gls@extramakeindexopts	175, 182, 509, 514
\@gls@fixbraces	197, 523
\@gls@noexpand@assign@default@plurals	247, 396
\@gls@noexpand@field	21, 74, 75, 414, 447
\@gls@noidx@no@sanitizesort	23, 415
\@gls@noidx@nosanitizesort	187, 415, 518
\@gls@noidx@sanitizesort	415
\@gls@nosanitizesort	22, 187, 415, 518
\@gls@sanitizesort	22, 187, 415, 518
\@gls@xdycheckbackslash	128, 482, 483
\@gls@xdycheckquote	127, 128, 482
\@glslocalreset	98, 100, 462, 464
\@glslocalunset	97, 100, 462, 464
\@glsreset	98, 100, 462–464
\@glsshowtarget	5, 6, 405, 406
\@glsunset	98, 100, 462, 464
\@newglossaryentry@defcounters	99, 463
\@this@glo@	58, 438
\@ACRfull	230, 543
\@ACRfullpl	231, 543
\@ACRlong	153, 230, 496, 543
\@ACRlongpl	155, 231, 497, 543
\@ACRshort	150, 230, 493, 543
\@ACRshortpl	151, 231, 494, 543
\@Acrfull	229, 542, 543
\@Acrfullpl	230, 231, 543
\@Acrlong	153, 229, 495, 543
\@Acrlongpl	155, 231, 496, 543

\@Acrshort	149, 493	\@Glsfirst	137, 487
\@Acrshortpl	151, 494	\@Glsfirst@	137, 487
\@Alpha	191, 194, 520–522	\@Glsfirstplural	139, 488
\@GLS	132, 485	\@Glsfirstplural@	139, 488
\@GLS@	132, 278, 485	\@Glsname	140, 488, 489
\@GLSdesc	141, 489	\@Glsname@	140, 489
\@GLSdesc@	141, 489	\@Glspl	133, 485
\@GLSdescplural	142, 489	\@Glspl@	101, 104, 133, 277, 464, 466, 485, 486
\@GLSdescplural@	142, 489	\@Glsplural	138, 488
\@GLSfirst	137, 138, 488	\@Glsplural@	138, 488
\@GLSfirst@	138, 488	\@Glsymbol	142, 490
\@GLSfirstplural	139, 488	\@Glsymbol@	142, 490
\@GLSfirstplural@	139, 488	\@Glsymbolplural	143, 490
\@GLSname	140, 489	\@Glsymbolplural@	143, 490
\@GLSname@	140, 489	\@Glstext	137, 487
\@GLSpl	134, 486	\@Glstext@	137, 487
\@GLSpl@	134, 278, 486	\@Glsuser@i	383, 384
\@GLSplural	138, 488	\@Glsuseri	144, 490
\@GLSplural@	138, 488	\@Glsuseri@	144, 490
\@GLSsymbol	142, 490	\@Glsuserii	145, 491
\@GLSsymbol@	142, 143, 490	\@Glsuserii@	145, 491
\@GLSsymbolplural	143, 490	\@Glsuseriii	145, 491
\@GLSsymbolplural@	143, 490	\@Glsuseriii@	145, 146, 491
\@GLStext	136, 487	\@Glsuseriv	146, 491
\@GLStext@	136, 137, 487	\@Glsuseriv@	146, 491
\@GLSuseri	144, 490	\@Glsuserv	147, 492
\@GLSuseri@	144, 490, 491	\@Glsuserv@	147, 492
\@GLSuserii	145, 491	\@Glsuservi	148, 492
\@GLSuserii@	145, 491	\@Glsuservi@	148, 492
\@GLSuseriii	146, 491	\@Mi	299
\@GLSuseriii@	146, 491	\@PGLS	277
\@GLSuseriv	146, 147, 492	\@PGLS@	277
\@GLSuseriv@	147, 492	\@PGLSpl	278
\@GLSuserv	147, 492	\@PGLSpl@	278
\@GLSuserv@	147, 492	\@Pgl	276
\@GLSuservi	148, 492	\@Pgl@	276
\@GLSuservi@	148, 492	\@Pglspl	276
\@Gls	131, 484	\@Pglspl@	276
\@Gls@	101, 103, 131, 276, 464, 465, 484	\@Roman	191, 194, 520, 521
\@Gls@acrentryname	232, 498, 544	\@acrfull	228, 542
\@Gls@entry@field	82, 157–163, 452, 497–501	\@acrfullpl	230, 543
\@Gls@entryname	157, 232, 498, 544	\@acrlong	152, 229, 495, 542
\@GlsSetXdyFirstLetterAfterDigits	167, 168, 503	\@acrlongpl	154, 230, 496, 543
\@GlsSetXdyNumberGroupOrder	168, 503, 504	\@acrshort	148, 229, 492, 542, 543
\@Glsdesc	140, 141, 489	\@acrshortpl	150, 230, 231, 494, 543
\@Glsdesc@	141, 489	\@addtoacronymlists	18, 412, 413
\@Glsdescplural	141, 489	\@afterheading	284, 288, 342
\@Glsdescplural@	141, 489	\@alpha	191, 194, 520–522
		\@arabic	191, 194, 520, 521

\@auxout .....	65, 66, 102, 182, 184, 185, 187, 199, 203, 280, 441, 442, 465, 514, 516, 518, 524, 526, 527	\@empty .....	11, 15, 18, 33, 36–38, 49, 51, 54, 57, 58, 90, 95, 119, 120, 130–134, 149–155, 169, 172, 174, 179–181, 188, 190, 196, 222, 248, 250, 252, 253, 255, 257, 258, 260, 262, 264, 336, 338, 340, 377–379, 409, 411, 412, 422, 424–426, 432, 433, 435, 437, 438, 457, 460, 461, 475, 476, 484–486, 493–497, 504, 506, 508, 512, 513, 519, 523, 539, 555, 557, 559–563, 565, 566, 568
\@backslashchar .....	121, 122, 128, 176, 477, 482, 483	\@end@fixbraces .....	197, 523
\@bsphack .....	190, 519	\@endfortrue ...	28, 61, 80, 280, 419, 439, 450
\@cGls .....	102, 465	\@esphack .....	190, 520
\@cGls@ .....	101, 103, 464, 465	\@expandtwoargs .....	36, 425
\@cGlspl .....	103, 466	\@firstofone .....	23, 31, 286, 415, 416, 421
\@cGlspl@ .....	101, 103, 464, 466	\@firstofthree	116, 130, 133, 135, 149, 150, 152, 154, 377–379, 474, 484–486, 493–496
\@cclv .....	299, 300	\@firstoftwo .....	26, 27, 79, 80, 116, 133, 134, 150–152, 154, 155, 418, 419, 449, 450, 474, 485, 486, 494–497
\@cgls .....	102, 465	\@for .....	28, 36, 38, 50, 51, 58, 79, 80, 121, 169–171, 181– 183, 191, 198, 204, 234, 248, 251, 253, 255, 257, 260, 262, 264, 280, 281, 336, 419, 425, 426, 432, 433, 438, 449, 450, 477, 504, 505, 514, 515, 520, 524, 527, 545, 555, 557, 559, 561, 562, 565, 567, 568
\@cgls@ .....	100, 102, 464, 465	\@glo@desc .....	92, 459
\@cglspl .....	103, 466	\@glo@symbol .....	92, 459
\@cglspl@ .....	100, 103, 464, 466	\@glo@access .....	356, 359, 364
\@chapter .....	38, 426	\@glo@addchildren ..	204, 209, 527, 528, 531
\@classoptionslist .....	36, 425	\@glo@assign@sortkey .....	183, 185, 215, 515, 516, 535
\@closegls .....	179, 512	\@glo@autosee .....	93, 459
\@colht .....	300	\@glo@autoseehook .....	93, 460
\@colroom .....	300	\@glo@check@mkidxrangechar .....	120, 121, 195, 335, 336, 476, 522
\@currentlabelname .....	8, 214, 407, 534	\@glo@check@sortallowed .....	13–16, 183, 187, 410, 411, 515, 518
\@curroptions .....	36, 425	\@glo@childlist .....	204, 527
\@declaredoptions .....	36, 425	\@glo@counter .....	71, 88, 92, 444, 456, 458
\@delimN .....	225, 541	\@glo@counterprefix ...	188, 192, 195, 196, 222, 223, 226, 518, 521–523, 539, 541
\@delimR .....	225, 540, 541	\@glo@default@sorttype .....	13, 185, 207, 208, 409, 410, 516, 529, 530
\@disable@onlypremakeg .....	182, 426, 515	\@glo@defaultcounter .....	91, 92, 458
\@disable@premakecs .....	38, 426	\@glo@desc	69, 87, 88, 90, 92, 444, 455–457, 459
\@disabled@gl saddx dy counters	51, 432, 433	\@glo@descaccess .....	357, 359, 360
\@do@addcounter .....	50, 432	\@glo@descplural ...	69, 87, 88, 444, 455, 456
\@do@auxoutstuff .....	203, 526, 527		
\@do@glossentry .....	218, 354, 355, 537		
\@do@gls@checkesc .....	122		
\@do@gls@getcounterprefix .....	192, 195, 521, 522		
\@do@gls@islistofacronyms .....	18, 413		
\@do@glssee .....	93, 460		
\@do@ifinlist .....	49, 432		
\@do@newglossaryentry ..	232, 246, 247, 249–254, 256, 257, 259–262, 264, 396– 400, 544, 554–558, 560–562, 564–566, 568		
\@do@seeglossary ...	184, 197, 516, 523, 524		
\@do@subglossentry .....	220, 355, 538		
\@do@wrglossary .....	119, 475, 520		
\@do@write@glslabels .....	32, 422		
\@do@writeaux@info .....	199, 524		
\@domakeglossaries .....	30, 31, 181, 184, 421, 514, 515		
\@ehc .....	299		

\@glo@descpluralaccess .....	357, 359, 360	\@glo@sort .....	13, 16, 22, 23, 69, 88, 91, 95, 96, 410, 411, 415, 444, 456, 458, 460, 461
\@glo@do@sortentries .....	204, 527, 528	\@glo@sortedinsert .....	205, 528
\@glo@entry .....	166, 503	\@glo@sortentries ..	207, 208, 527, 529, 530
\@glo@entryprefix .....	272	\@glo@sorthandler@case .....	208, 529, 530
\@glo@entryprefixfirst .....	272	\@glo@sorthandler@letter ...	207, 528, 529
\@glo@entryprefixfirstplural ..	272, 273	\@glo@sorthandler@nocase ...	208, 529, 530
\@glo@entryprefixplural .....	272	\@glo@sorthandler@word .....	207, 528, 529
\@glo@esclabel .....	95, 96, 460, 461	\@glo@sortinghandler ...	204, 205, 527, 528
\@glo@etext .....	106–108, 467, 468	\@glo@sortinglist .....	204, 205, 208, 527, 528, 530, 531
\@glo@first .....	70, 88, 91, 92, 259, 399, 444, 456, 458, 459, 564	\@glo@sortmacro@case .....	530
\@glo@firstaccess .....	356, 359	\@glo@sortmacro@def .....	530
\@glo@firstplural .....	70, 88, 91, 400, 444, 456, 458	\@glo@sortmacro@def@do .....	531
\@glo@firstpluralaccess .....	356, 359	\@glo@sortmacro@letter .....	529
\@glo@grabfirst .....	210, 532	\@glo@sortmacro@nocase .....	530
\@glo@label .....	73, 74, 81, 82, 84–93, 99, 164, 165, 272, 273, 330, 359, 360, 446, 451–460, 463, 502	\@glo@sortmacro@standard .....	530
\@glo@list .....	93, 460	\@glo@sortmacro@use .....	531
\@glo@long ..	62, 73, 89, 92, 440, 446, 456, 458	\@glo@sortmacro@word .....	529
\@glo@longaccess .....	357, 359, 360	\@glo@sorttype .....	185, 209, 210, 215, 516, 531, 532, 535
\@glo@longpl .....	73, 89, 92, 247– 249, 251, 254, 256, 259, 261, 396, 446, 456, 458, 554–556, 558, 560, 562, 564, 566	\@glo@storeentry .....	13–15, 410, 411, 460
\@glo@longpluralaccess .....	357, 359, 360	\@glo@suffix ..	120, 121, 195, 336, 476, 477, 522
\@glo@name .....	13, 69, 74, 88, 90–92, 410, 444, 447, 456–458	\@glo@symbol .....	62, 70, 88, 92, 253, 258, 259, 440, 444, 456, 459, 559, 563, 564
\@glo@no@assign@sortkey ....	183, 515, 535	\@glo@symbolaccess .....	356, 359, 399
\@glo@nonumberlist .....	72, 445	\@glo@symbolplural ..	70, 88, 93, 444, 456, 459
\@glo@numfmt .....	196, 336, 522	\@glo@symbolpluralaccess .....	356, 359
\@glo@parent .....	15, 71, 88, 90, 91, 95, 96, 205, 411, 445, 456–458, 460, 461, 528	\@glo@text .....	69, 88, 91, 92, 131–135, 156–158, 254, 273, 398, 444, 456, 458, 459, 484–487, 497, 498, 560
\@glo@plural ..	69, 88, 90, 91, 398, 444, 456–458	\@glo@textaccess .....	356, 359, 396–398
\@glo@pluralaccess .....	356, 359	\@glo@thislabel .....	94, 460
\@glo@prefix .....	9, 71, 72, 88, 95, 96, 120, 121, 195, 335, 336, 408, 445, 456, 460, 461, 476, 477, 522	\@glo@thislettergrp ....	210, 211, 531–533
\@glo@range .....	195, 335, 336, 522	\@glo@thisvalue .....	63, 64, 440, 441
\@glo@see .....	71, 88, 93, 445, 456, 460	\@glo@tmp .....	81, 82, 196, 451, 452, 523
\@glo@seeautonumberlist ...	9, 71, 408, 445	\@glo@type ....	8, 15, 70, 88–93, 165, 166, 182, 188, 200–205, 208–210, 213, 214, 232, 248, 250, 252, 255, 257, 260, 262, 264, 279, 281, 407, 411, 444, 456–459, 502, 503, 514, 519, 525–528, 531, 534, 544, 555, 557, 559, 560, 562, 565, 566, 568
\@glo@short .....	63, 73, 89, 92, 399, 440, 446, 456, 458	\@glo@types .....	31, 58, 66, 98, 99, 166, 181–183, 270, 330, 422, 438, 441, 442, 463, 503, 514, 515, 570
\@glo@shortaccess ..	357, 359, 360, 396–399	\@glo@useri .....	72, 89, 92, 446, 456, 458
\@glo@shortpl .....	73, 89, 92, 246–249, 251, 254, 256, 259, 261, 396, 399, 446, 456, 458, 554, 556, 558, 560, 562, 564, 566	\@glo@useriaccess .....	357, 359, 360
\@glo@shortpluralaccess ....	357, 359, 360	\@glo@userii .....	72, 89, 92, 446, 456, 458
		\@glo@useriaccess .....	357, 359, 360

<code>\@glo@useriii</code> .....	73, 89, 92, 446, 456, 458	<code>\@gls@classI</code> .....	169, 170, 504, 505
<code>\@glo@useriiiaccess</code> .....	357, 359, 360	<code>\@gls@classII</code> .....	170, 505
<code>\@glo@useriv</code> .....	73, 89, 92, 446, 456, 458	<code>\@gls@codepage</code> .....	203, 437, 527
<code>\@glo@userivaccess</code> .....	358–360	<code>\@gls@counter</code> .....	
<code>\@glo@userv</code> .....	73, 89, 92, 446, 456, 458	.....	115, 118–120, 165, 166, 188, 195,
<code>\@glo@uservaccess</code> .....	358–360		196, 336, 473, 475, 476, 502, 518, 522, 523
<code>\@glo@uservi</code> .....	73, 89, 92, 446, 456, 458	<code>\@gls@counterwithin</code> .....	11, 12, 409
<code>\@glo@userviaccess</code> .....	358–360	<code>\@gls@ctr</code> .....	50, 432
<code>\@glodesc</code> .....	92, 459	<code>\@gls@currentlettergroup</code>	209, 211, 531–533
<code>\@glolist@</code> .....	90, 457	<code>\@gls@debugfalse</code> .....	4, 405
<code>\@gloname</code> .....	92, 458	<code>\@gls@debugtrue</code> .....	4, 5, 405
<code>\@glossary@default@style</code> .....	8, 10,	<code>\@gls@declareoption</code> .....	
	201, 223, 265, 407, 408, 525, 539, 540, 568	.....	9, 10, 16, 17, 20, 21, 26, 29–
<code>\@glossaryentryfield</code> .....	95, 460, 461		32, 35, 36, 408, 412, 414, 418, 420–422, 424
<code>\@glossarysection</code> .....	46, 429, 430	<code>\@gls@default</code> .....	104, 466
<code>\@glossarystyle</code> ..	201, 202, 213, 525, 526, 534	<code>\@gls@default@value</code> .....	
<code>\@glossarysubentryfield</code> ..	95, 96, 460, 461	.....	62–64, 74, 75, 88, 90–92, 258,
<code>\@gls</code> .....	130, 484		272, 440, 441, 444, 447, 448, 456–459, 563
<code>\@gls@</code> ...	100, 102, 130, 275, 276, 464, 465, 484	<code>\@gls@deffile</code> .....	77–79, 449, 450
<code>\@gls@@automake@immediate</code> .....	182, 514	<code>\@gls@define@glossaryentrycounter</code> ..	
<code>\@gls@@link</code> .....	117, 474	.....	12, 37, 214, 216, 409, 425, 535
<code>\@gls@Hcounter</code> .....	119, 120, 475, 476	<code>\@gls@define@glossarysubentrycounter</code>	
<code>\@gls@ReturnAfterFi</code> .....	226, 541	.....	37, 215, 216, 409, 425, 535
<code>\@gls@actualchar</code> .....	95,	<code>\@gls@defsort</code> .....	13–15, 92, 410, 411, 459
	96, 124, 126, 173, 339, 461, 479, 481, 507	<code>\@gls@defsortcount</code> ..	13–15, 67, 410, 411, 442
<code>\@gls@addpredefinedattributes</code> .....		<code>\@gls@deprecated@acr@warning</code> .....	
	169, 178, 433, 434, 504, 511	.....	250, 252, 254, 257, 260, 262
<code>\@gls@adjustmode</code> .....	165, 502, 503	<code>\@gls@do@acronymsdef</code>	17, 37, 68, 412, 425, 443
<code>\@gls@after</code> .....	18, 413	<code>\@gls@do@indexdef</code> ..	36–38, 68, 424–426, 443
<code>\@gls@automake</code> .....	183, 512, 515	<code>\@gls@do@numbersdef</code>	35–37, 68, 424, 425, 443
<code>\@gls@automake@immediate</code> ...	181, 513, 514	<code>\@gls@do@symbolsdef</code> .....	35, 68, 424, 443
<code>\@gls@before</code> .....	18, 413	<code>\@gls@do@symbolssdef</code> .....	37, 38, 425, 426
<code>\@gls@between</code> .....	281	<code>\@gls@doautomake</code> .....	
<code>\@gls@body</code> .....	157, 158, 498	.....	32, 33, 181, 183, 422, 514, 515
<code>\@gls@checkactual</code>	122, 176, 478, 481, 482, 510	<code>\@gls@docheckquotedef</code> ..	175–177, 509–511
<code>\@gls@checkbar</code> ..	122, 176, 478, 480, 481, 510	<code>\@gls@docloadedfalse</code> .....	4, 404
<code>\@gls@checkedmkidx</code> .....		<code>\@gls@docloadedtrue</code> .....	3, 4, 404
	121–128, 175–177, 477–483, 509–511	<code>\@gls@dodeflistparser</code> .....	183, 515
<code>\@gls@checkesc</code> .....	122, 176	<code>\@gls@doentrycounterdef</code> ..	37, 38, 425, 426
<code>\@gls@checkescactual</code>	122, 176, 477, 479, 510	<code>\@gls@doentrydef</code> .....	114, 115, 472, 473
<code>\@gls@checkescbar</code> ..	122, 176, 478–480, 510	<code>\@gls@dolast</code> .....	197, 198, 524
<code>\@gls@checkesclevel</code> .....	122, 176, 480	<code>\@gls@donext</code> .....	198, 524
<code>\@gls@checkescquote</code> .....		<code>\@gls@donext@def</code> .....	164, 502
	122, 176, 177, 477–479, 509–511	<code>\@gls@dosubentrycounterdef</code>	37, 38, 425, 426
<code>\@gls@checklevel</code> ...	122, 176, 478, 481, 510	<code>\@gls@dothiswrite</code> .....	179–181, 512–514
<code>\@gls@checkmkidxchars</code> .....		<code>\@gls@elem</code> .....	280
	95, 120, 176, 184, 195, 197,	<code>\@gls@enablesavenonumberlist</code>	78, 445, 449
	335, 336, 460, 476, 477, 509, 516, 522, 523	<code>\@gls@encapchar</code> .....	124, 125, 173,
<code>\@gls@checkquote</code>	122, 175, 176, 477, 478, 509		196, 197, 336, 339, 479–481, 507, 522, 523

<code>\@gls@entry@count</code> .....	101, 102, 465	<code>\@gls@link@nocheckfirsthyper</code> .....	
<code>\@gls@entry@field</code> .....	81, 82, 100,		136, 149–155, 487, 493–497
	157–164, 360–362, 451, 452, 463, 497–501	<code>\@gls@link@opts</code> .	118, 249, 256, 475, 556, 561
<code>\@gls@escbsdq</code> .....	122, 174, 340, 477, 508	<code>\@gls@list</code> .....	280, 281
<code>\@gls@expand@fields</code> .....	75, 76, 447, 448	<code>\@gls@listsuffix</code> .....	49, 432
<code>\@gls@expandonce</code> .....	76, 448	<code>\@gls@loadlist</code> .....	10, 265, 408, 568
<code>\@gls@extramakeindexopts</code> ...	182, 511, 514	<code>\@gls@loadlong</code> .....	10, 265, 408, 568
<code>\@gls@fetchfield</code> .....	63, 441, 450	<code>\@gls@loadsuper</code> .....	10, 265, 408, 568
<code>\@gls@field@link</code> .....		<code>\@gls@loadtree</code> .....	10, 265, 408, 568
	. 83, 136–148, 379–384, 452, 453, 487–492	<code>\@gls@local@increment@currcount</code> ...	
<code>\@gls@fieldaccess@display</code> .....	364–367		100, 464, 465
<code>\@gls@firsttok</code> .....	210, 531, 532	<code>\@gls@loclist</code> .....	
<code>\@gls@fixbraces</code> .....	93, 460, 523		185, 186, 210, 211, 516, 517, 532, 533
<code>\@gls@forbidtexext</code> .....	66, 442	<code>\@gls@map</code> .....	79, 80, 449, 450
<code>\@gls@fulllist</code> .....	31	<code>\@gls@missinglang@warn</code> ...	20, 41, 414, 427
<code>\@gls@get@counterprefix</code> .....	196, 523	<code>\@gls@missingnumberlist</code> .....	92, 447, 459
<code>\@gls@getbody</code> .....	157, 498	<code>\@gls@noaccess</code> .....	364
<code>\@gls@getcounter</code> .....	443	<code>\@gls@noexpand@fields</code> .....	76, 447, 448
<code>\@gls@getcounterprefix</code> .	192, 195, 521–523	<code>\@gls@nohyperlist</code> .	19, 68, 118, 413, 443, 475
<code>\@gls@getgrouptitle</code> .....		<code>\@gls@noidx@do</code> .....	209, 531, 532
	184, 221, 281, 516, 538, 539	<code>\@gls@noidx@getgrouptitle</code> ..	184, 516, 539
<code>\@gls@glossary</code> .....	189, 190, 519	<code>\@gls@noidx@sanitizesort</code> .	22, 187, 415, 518
<code>\@gls@gobbleopt</code> .....	65, 442	<code>\@gls@noidx@setsanitizesort</code> .....	
<code>\@gls@grptitle</code> .....	221, 279, 281, 538		25, 187, 417, 518
<code>\@gls@hyp@opt</code> .....	83, 102,	<code>\@gls@noidx@loclist@finalsep</code> ...	186, 517
	103, 117, 130–155, 228–231, 274–278,	<code>\@gls@noidx@loclist@prev</code> .	186, 212, 517, 533
	452, 453, 465, 466, 474, 484–497, 542, 543	<code>\@gls@noidx@loclist@sep</code> .	186, 212, 517, 533
<code>\@gls@hyp@opt@cs</code> .....	116, 474	<code>\@gls@noref@warn</code> ...	184, 210, 516, 518, 531
<code>\@gls@hypergroup</code> .....	280	<code>\@gls@numberlink</code> .....	225, 226, 541
<code>\@gls@ifinlist</code> .....	50, 432	<code>\@gls@numbersdef</code> .....	35, 424
<code>\@gls@ifnotmeasuring</code> .....	96, 461	<code>\@gls@numlist@lastsep</code> .....	164, 165, 502
<code>\@gls@igtype</code> .....	68, 444	<code>\@gls@numlist@nextsep</code> .....	164, 502
<code>\@gls@increment@currcount</code> ..	100, 464, 465	<code>\@gls@numlist@sep</code> .....	164, 165, 502
<code>\@gls@indexdef</code> .....	36, 424	<code>\@gls@old@chapter</code> .....	38, 426
<code>\@gls@initnonumberlist</code> ...	72, 89, 445, 456	<code>\@gls@oldnewglossaryentryposthook</code> .	359
<code>\@gls@islistofacronyms</code> .....	18, 413	<code>\@gls@oldnewglossaryentryprehook</code> ..	
<code>\@gls@keylist</code> .....	395		358, 359
<code>\@gls@keymap</code> .	72, 79–82, 272, 358, 446, 449–451	<code>\@gls@onlypremakeg</code> .....	38, 426
<code>\@gls@label</code> .....		<code>\@gls@order</code> .....	179–181, 512, 513
	184, 188, 192, 195, 196, 516, 518, 521, 522	<code>\@gls@org@LT@output</code> .....	299
<code>\@gls@labelsfile</code> .....	31, 32, 422	<code>\@gls@org@glsnoidx@displayloc</code> .	186, 517, 518
<code>\@gls@langmod</code> .....	179, 180, 512, 513	<code>\@gls@org@glseeformat</code> ....	186, 517, 518
<code>\@gls@levelchar</code> .....		<code>\@gls@override@glossary</code> ..	34, 35, 423, 424
	. 96, 125, 126, 173, 339, 461, 480, 481, 507	<code>\@gls@patchtabularx</code> .....	96, 461
<code>\@gls@link</code> .....	117, 131–136, 149–	<code>\@gls@preglossaryhook</code> .....	202, 525, 526
	156, 377–379, 474, 475, 484–487, 493–497	<code>\@gls@prevlevel</code> .	310, 311, 330–333, 348, 349
<code>\@gls@link@checkfirsthyper</code> .....		<code>\@gls@provide@newglossary</code> ...	66, 441, 442
	130–135, 474, 484–486	<code>\@gls@quotechar</code> .....	123–
<code>\@gls@link@label</code> .	118, 249, 256, 475, 556, 561		126, 173, 175, 177, 339, 478–481, 507–511



<code>\@gls@reference</code> .	184, 185, 188, 516, 518, 533	<code>\@gls@usetranslator</code> ..	27, 40, 418, 419, 427
<code>\@gls@removespaces</code> .....	226, 541	<code>\@gls@value</code> .....	75, 160, 447, 448, 499
<code>\@gls@renewglossary</code> .....	178, 511, 519	<code>\@gls@warnonglossdefined</code>	21, 199, 414, 524
<code>\@gls@replacementtext</code> .....	363, 364	<code>\@gls@warnontheGLOSSdefined</code> .....	21, 218, 414, 537
<code>\@gls@rest</code> .....	157, 498	<code>\@gls@write@entrycounts</code> .....	101, 465
<code>\@gls@restoreat</code> .....	78, 449	<code>\@gls@writedef</code> .....	78, 449
<code>\@gls@roman</code> .....	53, 336, 337, 434	<code>\@gls@writeisthook</code> .....	173, 174, 507, 508
<code>\@gls@sanitized@tmp</code> .....	121, 477	<code>\@gls@x</code> .....	226
<code>\@gls@sanitizedesc</code> .....	28, 414, 420	<code>\@gls@xdy@locationlist</code> .....	169, 170, 434, 504, 505
<code>\@gls@sanitizename</code> .....	415	<code>\@gls@xdycheckbackslash</code>	121, 477, 482, 483
<code>\@gls@sanitizesort</code> .....	13, 410, 415	<code>\@gls@xdycheckquote</code> .....	121, 477, 482
<code>\@gls@sanitizesymbol</code> .....	28, 29, 415, 420	<code>\@gls@xref</code> .....	197, 523
<code>\@gls@saveentrycounter</code> .	119, 166, 475, 502	<code>\@gls@Alphacompositor</code> .	44, 53, 337, 429, 435
<code>\@gls@savenonumberlist</code> .....	72, 445	<code>\@gls@Hlocref</code> .....	192, 195, 521, 522
<code>\@gls@see@noindex</code> .....	7, 71, 406, 407, 445	<code>\@gls@acronymlists</code> .....	18, 19, 58, 232–234, 248, 250–253, 255, 257, 260, 262, 264, 269, 412, 413, 438, 544, 545, 555, 557, 559–562, 565–568, 570
<code>\@gls@setacronymstyle</code> .....	28, 29, 37, 420, 425	<code>\@gls@addkey</code> .....	82, 451
<code>\@gls@setcounter</code> .....	67, 442, 443	<code>\@gls@addstoragekey</code> .....	81, 451
<code>\@gls@setdefault@glslink@opts</code> .	118, 475	<code>\@gls@addxdyattribute</code> .....	50, 51, 432, 433
<code>\@gls@setsort</code> .....	13–16, 119, 166, 410, 411, 475, 502	<code>\@gls@defaultplural</code> .....	446
<code>\@gls@setupshortcuts</code> .....	37, 425	<code>\@gls@defaultsort</code> .....	13, 410, 447
<code>\@gls@setupsort@def</code> .....	410	<code>\@gls@desc</code> .....	140, 489
<code>\@gls@setupsort@none</code> .....	411	<code>\@gls@desc@</code> .....	140, 489
<code>\@gls@setupsort@standard</code> .....	410	<code>\@gls@descplural</code> .....	141, 489
<code>\@gls@setupsort@use</code> .....	411	<code>\@gls@descplural@</code> .....	141, 489
<code>\@gls@sort</code> .....	211, 532	<code>\@gls@disp</code> .....	135, 486
<code>\@gls@sort@A</code> .....	206, 528, 529	<code>\@gls@entry</code> ..	31, 32, 98, 99, 102, 422, 463, 465
<code>\@gls@sort@B</code> .....	206, 528, 529	<code>\@gls@entrytitlecase</code> .....	160, 499, 500
<code>\@gls@startswithexpandonce</code> .....	75, 448	<code>\@gls@first</code> .....	137, 487
<code>\@gls@storenonumberlist</code> ..	72, 92, 445, 459	<code>\@gls@first@</code> .....	137, 487
<code>\@gls@symbolsdef</code> .....	35, 424	<code>\@gls@firstletter</code> .....	168, 503
<code>\@gls@this</code> .....	191, 520	<code>\@gls@firstplural</code> .....	139, 488
<code>\@gls@thisHloc</code> .....	196, 523	<code>\@gls@firstplural@</code> .....	139, 380, 488
<code>\@gls@thisentryname</code> .....	31, 32	<code>\@gls@hypernumber</code> .....	225, 540, 541
<code>\@gls@thisfield</code> .....	63, 64, 441	<code>\@gls@isacronymlistfalse</code> .....	19, 413
<code>\@gls@thislabel</code> ..	61, 198, 208, 439, 524, 530	<code>\@gls@isacronymlisttrue</code> .....	19, 413
<code>\@gls@thislist</code> .....	164, 165, 502	<code>\@gls@link</code> .....	119, 129, 130, 165, 279, 475, 483, 484, 502
<code>\@gls@thisloc</code> .....	196, 523	<code>\@gls@localreset</code> .....	97, 100, 462, 464
<code>\@gls@thisval</code> .....	80, 450	<code>\@gls@localunset</code> .....	97, 100, 462, 464
<code>\@gls@title</code> .....	46, 429	<code>\@gls@locref</code> .....	188, 192, 195, 196, 335, 336, 518, 521, 522
<code>\@gls@tmp</code> ...	6, 15, 41, 53, 54, 76, 121, 190, 280, 281, 406, 411, 427, 435, 448, 477, 519	<code>\@gls@minrange</code> .....	168, 170, 337, 504, 505
<code>\@gls@tmpb</code>	123–128, 175, 177, 478–483, 509–511	<code>\@gls@name</code> .....	139, 488
<code>\@gls@toc</code> .....	47, 430, 431	<code>\@gls@name@</code> .....	139, 140, 488
<code>\@gls@type</code> ...	181, 183, 234, 248, 251, 253, 255, 257, 260, 262, 264, 330, 514, 515, 545, 555, 557, 559, 561, 562, 565, 567, 568		
<code>\@gls@updatechecked</code> .....	121, 122, 176, 477, 478, 509, 510		



\@glshnavhypertarget	279	\@ifglossaryexists	59, 199, 438, 524
\@glshnextpages	201, 526, 535	\@ifl@t@r	24, 416
\@glshnodesc	88, 90, 92, 446, 456, 457, 459	\@ifnextchar	67, 116, 442, 474
\@glshnoname	88, 90, 92, 446, 456–458	\@ifpackageloaded	...
\@glshnonextpages	201, 526, 535	..	3, 8, 26–28, 40, 57, 96, 164, 175, 354, 404, 407, 418, 419, 427, 437, 461, 501, 508
\@glshnumberformat	115, 118, 165, 188, 195, 196, 335, 336, 473, 475, 502, 518, 522	\@ifstar	59, 66, 81, 82, 116, 167, 168, 227, 438, 442, 451, 474, 503, 542
\@glshopenfile	178, 188, 511, 512, 519	\@ifundefined	40, 280, 289, 290, 300, 311, 312, 318, 331, 332, 348, 427
\@glshorder	182, 420, 514	\@ignored@glossaries	31, 68, 422, 443, 444
\@glshspl	133, 485	\@input@	202, 526
\@glshspl@	101, 103, 133, 275, 277, 464, 466, 485	\@istfilename	182, 429, 514
\@glshplural	138, 488	\@makecol	299, 300
\@glshplural@	138, 488	\@makeglossaries@warn@noprintglossary	183
\@glshprefix@record@hook	275–278	\@makeglossary	30, 31, 182, 421, 511, 514, 515
\@glshreset	97, 100, 462, 464	\@minus	284, 285, 306, 324
\@glshsee	93, 197, 460, 524	\@mkboth	46, 47, 430
\@glshshowaccsupp	5, 356, 405, 406	\@newglossary	65, 66, 441–443
\@glshshowtarget	4, 5, 129, 405, 406, 483	\@newglossaryentry@defcounters	93, 99, 459, 460, 463
\@glshsymbol	142, 489	\@newglossaryentryposthook	81, 82, 93, 272, 359, 451, 459, 460
\@glshsymbol@	142, 490	\@newglossaryentryprehook	81, 82, 87, 89, 272, 358, 451, 455, 456, 460
\@glshsymbolplural	143, 490	\@nil	18, 93, 120–122, 157, 176, 195, 197, 210, 211, 225, 226, 335, 336, 413, 460, 476–478, 498, 509, 510, 522, 523, 531, 532, 540, 541
\@glshsymbolplural@	143, 490	\@nnil	18, 198, 413, 524
\@glshtarget	129, 130, 218, 280, 483, 484, 537	\@no@makeglossaries	30, 31, 183, 185, 421, 515, 516, 518
\@glshtext	136, 487	\@no@post@desc	341, 428
\@glshtext@	136, 487	\@nopostdesc	201, 428, 526
\@glshunset	97, 100, 462, 464	\@onelevel@sanitize	6, 22, 32, 53, 79, 95, 121, 168, 172, 197, 199, 210, 337, 338, 406, 415, 434, 449, 461, 477, 503, 504, 506, 523, 524, 531
\@glshuseri	144, 490	\@onlypreamble	67, 77, 87, 88, 101, 104, 183, 187, 443, 449, 455, 456, 465, 466, 515, 518
\@glshuseri@	144, 490	\@onlypremakeg	43, 44, 50, 51, 55, 67, 174, 426, 428, 429, 432, 433, 436, 443, 508
\@glshuserii	144, 491	\@org@glossaryentrynumbers	201, 202, 525, 526
\@glshuseriii	144, 491	\@org@glsh@assign@descplural	247, 256, 257, 259–262, 396, 399, 400, 555, 562, 564–566
\@glshuseriii@	145, 491		
\@glshuseriv	146, 491		
\@glshuseriv@	146, 491		
\@glshuserv	147, 492		
\@glshuserv@	147, 492		
\@glshuservi	148, 492		
\@glshuservi@	148, 492		
\@glshwidestname	330–332, 348		
\@glshwritefiles	33, 422, 519		
\@glshxtr@doaccsupp	354		
\@glshxtr@record	274		
\@gobble	4, 5, 13–15, 30–32, 79, 96, 121, 167, 171, 182, 184, 286, 334, 338, 339, 405, 410, 411, 421, 449, 450, 461, 477, 503, 506, 515, 516		
\@gobblethree	5, 405		
\@idxitem	325		
\@ifclassloaded	4, 11, 46, 404, 409, 430		

<code>\@org@gl@assign@firstpl</code> .....	<code>\@tmp</code> .....
..... 247, 250, 252, 254, 256, 257, 259–	337, 434
262, 396–400, 555–558, 560, 562, 564, 566	<code>\@use@ption</code> .....
<code>\@org@gl@assign@plural</code> .....	36, 425
. 247, 250, 252, 254, 256, 257, 259–262,	<code>\@warn@nomakeglossaries</code> 181, 203, 514, 527
396–400, 555, 557, 558, 560, 562, 564–566	<code>\@wrglossary@pageformat</code> .... 191, 520, 521
<code>\@org@gl@assign@symbolplural</code> .....	<code>\@wrglossarynumberhook</code> . 191, 194, 520–522
..... 247, 250, 252, 254, 259–262,	<code>\@write@gl@labelsname</code> .....
396–398, 400, 555, 557, 558, 560, 564–566	32
<code>\@org@gl@numberformat</code> .....	<code>\@xdy@main@language</code> .....
164, 502	..... 29, 179, 180, 203, 420, 512, 513, 526
<code>\@org@newglossaryentryprehook</code> .. 87, 455	<code>\@xdy@attributelist</code> .. 51, 171, 432, 433, 505
<code>\@outputpage</code> .....	<code>\@xdy@attributes</code> .....
299, 300	..... 50, 169, 334, 336, 432, 433, 504
<code>\@p@glossarysection</code> .....	<code>\@xdy@counters</code> .... 49–51, 171, 432, 433, 505
46, 429, 430	<code>\@xdy@crossrefhook</code> .....
<code>\@pgls</code> .....	170, 435, 505
274	<code>\@xdy@language</code> .....
<code>\@pgls@</code> .....	203, 437, 526, 527
274, 275	<code>\@xdy@lettergroups</code> 57, 172, 339, 437, 438, 507
<code>\@pglspl</code> .....	<code>\@xdy@locationclassorder</code> .....
275	..... 55, 171, 338, 436, 505
<code>\@pglspl@</code> .....	<code>\@xdy@locref</code> .. 50, 172, 334, 338, 432, 433, 506
275	<code>\@xdynumbergrouporder</code> 57, 168, 437, 503, 504
<code>\@plus</code> .....	<code>\@xdy@requiredstyles</code> .....
284, 285, 306, 324	..... 56, 169, 336, 436, 437, 504
<code>\@print@glossary</code> .....	<code>\@xdysortrules</code> .... 55, 173, 339, 436, 507
200, 525, 526	<code>\@xdystyle</code> .....
<code>\@print@noidx@glossary</code> .... 200, 525, 531	169, 336, 504
<code>\@print@gloss@checkexists</code> ... 200, 201, 525	<code>\@xdy@useralphabets</code> .. 52, 169, 336, 434, 504
<code>\@print@gloss@checkexists@allowignored</code>	<code>\@xdy@userlocationdefs</code> .....
..... 525	..... 54, 170, 335, 337, 435, 505
<code>\@print@gloss@checkexists@noignored</code>	<code>\@xdy@userlocationnames</code> 54, 55, 335, 435, 436
..... 199, 200, 524, 525	<code>\@xfor@nextelement</code> .....
<code>\@print@gloss@setsort</code> .....	198, 524
..... 183, 185, 201, 515, 516, 525	<code>\@</code> .....
<code>\@print@glossary</code> .....	94, 121, 127, 167,
200, 525	173, 174, 225, 226, 339, 340, 342–344,
<code>\@roman</code> .....	352, 353, 460, 477, 503, 507, 508, 540, 541
53, 336, 434	<code>\{</code> .... 79, 167, 174, 334, 339, 340, 449, 503, 508
<code>\@secondofthree</code> . 116, 130, 131, 134, 149,	<code>\}</code> .... 79, 167, 174, 334, 340, 449, 450, 503, 508
151, 153, 155, 377, 474, 484, 486, 493–496	<code>\~</code> .....
<code>\@secondoftwo</code> .....	23, 32, 416
23, 26, 27,	<code>\‘</code> .....
41, 79, 80, 129–132, 135, 149, 150, 152,	23, 416
153, 286, 377–379, 403, 415, 416, 418,	<code>\ </code> .....
419, 427, 449, 450, 483–486, 493, 495, 496	122, 124, 127, 176, 478–480, 510
<code>\@set@glo@numformat</code> .... 196, 336, 476, 522	<code>\~</code> .....
<code>\@sglsaddkey</code> .....	23, 416
82, 451	
<code>\@sglsaddstoragekey</code> .....	
81, 451	
<code>\@tabacckludge</code> .....	
23, 416	
<code>\@text@composite@x</code> .....	
23, 416	
<code>\@thirdofthree</code> .. 116, 132, 134, 150, 152,	
153, 155, 378, 474, 485, 486, 493, 495–497	
<code>\@this@attr</code> .....	
171, 505, 506	
<code>\@this@childlabel</code> .....	
204, 205, 527, 528	
<code>\@this@counter</code> .....	
51, 433	
<code>\@this@ctr</code> .....	
171, 505, 506	
<code>\@this@key</code> .....	
80, 450	
<code>\@this@label</code> .....	
204, 527	
<code>\@this@cs</code> .....	
38, 426	

## A

<code>\a</code> .....	23, 416
<code>\AA</code> .....	24, 416
<code>\aa</code> .....	24, 416
<code>\Ac</code> .....	554
<code>\ac</code> .....	554
<code>accsupp package</code> .....	354, 355
<code>\accsuppglossaryentryfield</code> .....	354
<code>\accsuppglossarysubentryfield</code> ....	355
<code>\Acf</code> .....	554
<code>\acf</code> .....	554

<code>\Acfp</code> .....	554	<code>\acronymtype</code> .....	17, 232, 234, 246, 248–252, 254–257, 259–262, 264, 396– 399, 412, 544, 545, 554–562, 564–566, 568
<code>\acfp</code> .....	554	<code>\acrpluralsuffix</code> .....	232, 235–237, 241–244, 247–249, 251, 254–257, 259– 261, 264, 387, 392–394, 396–400, 542, 544, 546–548, 551–553, 555–558, 560–568
<code>\Acl</code> .....	554	<code>\ACRshort</code> .....	493
<code>\acl</code> .....	554	<code>\Acrshort</code> .....	245, 493, 554
<code>\Aclp</code> .....	554	<code>\acrshort</code> .....	245, 492, 554
<code>\aclp</code> .....	554	<code>\ACRshorttpl</code> .....	494
<code>\Acp</code> .....	554	<code>\Acrshorttpl</code> .....	245, 494, 554
<code>\acp</code> .....	554	<code>\acrshorttpl</code> .....	245, 494, 554
<code>\acrfootnote</code> .....	249, 256, 555, 556, 561	<code>\Acs</code> .....	554
<code>\ACRfull</code> .....	543	<code>\acs</code> .....	554
<code>\Acrfull</code> .....	245, 542, 554	<code>\Acsp</code> .....	554
<code>\acrfull</code> .....	245, 542, 554	<code>\acsp</code> .....	554
<code>\ACRfullfmt</code> .....	230, 233, 241, 243, 391, 393, 543, 544, 551, 552	<code>\add@accent@</code> .....	23, 415
<code>\Acrfullfmt</code> .....	229, 233, 241, 243, 391, 393, 543, 544, 551, 552	<code>\addcontentsline</code> .....	48, 431
<code>\acrfullfmt</code> .....	229, 233, 241, 242, 391, 393, 542, 544, 551, 552	<code>\addglossarytocaptions</code> .....	41, 427
<code>\acrfullformat</code> .....	163, 229, 246, 247, 263, 501, 542, 554, 567	<code>\addtolength</code> .....	332, 348
<code>\ACRfullpl</code> .....	543	<code>\advance</code> .....	14, 15, 90, 119, 299, 410, 411, 457, 476
<code>\Acrfullpl</code> .....	246, 543, 554	<code>\AE</code> .....	23, 416
<code>\acrfullpl</code> .....	245, 543, 554	<code>\ae</code> .....	23, 416
<code>\ACRfullplfmt</code> .....	231, 233, 241, 243, 391, 393, 543, 545, 551, 552	<code>\altnewglossary</code> .....	443
<code>\Acrfullplfmt</code> .....	231, 233, 241, 243, 391, 393, 543, 545, 551, 552	<code>amsgen package</code> .....	<u>3</u> , <u>115</u>
<code>\acrfullplfmt</code> .....	230, 233, 241, 243, 391, 393, 543, 544, 551, 552	<code>amsmath package</code> .....	<u>96</u>
<code>\acrlinkfootnote</code> .....	248, 555	<code>\andname</code> .....	198, 426, 524
<code>\acrlinkfullformat</code> .....	229–231, 542, 543	<code>\AnyTrackedLanguages</code> .....	41, 403, 427
<code>\ACRlong</code> .....	495	<code>\appto</code> .....	19, 24, 72, 81, 82, 272, 358, 413, 416, 446, 451
<code>\Acrlong</code> .....	245, 495, 554	<code>array package</code> .....	<u>296</u> , <u>300</u> , <u>318</u>
<code>\acrlong</code> .....	245, 495, 554	<code>article class</code> .....	<u>196</u>
<code>\ACRlongpl</code> .....	497	<code>\AtBeginDocument</code> .....	17, 57, 78, 96, 166, 185, 412, 437, 449, 461, 503, 516
<code>\Acrlongpl</code> .....	245, 496, 554	<code>\AtEndDocument</code> .....	31, 33, 78, 101, 184, 188, 203, 280, 422, 449, 465, 515, 518, 526, 527
<code>\acrlongpl</code> .....	245, 496, 554		
<code>\acrnameformat</code> .....	254, 397, 544, 560		
<code>\acrno linkfootnote</code> .....	556		
<code>\acronymentry</code> .....	232, 235, 236, 238, 239, 241–244, 386–389, 392–395, 544–554		
<code>\acronymfont</code> .....	111, 112, 149–152, 157, 163, 231, 233, 235–239, 241–244, 249, 251, 253, 255–260, 374, 375, 377–379, 387– 389, 391–395, 397–399, 471, 493–495, 498, 501, 544–554, 556, 557, 559, 561–565		
<code>\acronymname</code> .....	17, 42, 412, 426, 428		
<code>\acronymsort</code> .....	232, 235, 236, 238, 239, 241, 242, 244, 387–389, 392–395, 544–554		

## B

<code>\b</code> .....	23, 416
<code>babel package</code> .....	<u>26</u> , <u>39</u> , <u>41</u> , <u>56</u>
<code>\begin</code> .....	171, 209, 284, 286, 290–295, 298, 299, 301–324, 338, 506, 531
<code>\BeginAccSupp</code> .....	355
<code>\begingroup</code> .....	4, 6, 32, 190, 194, 226, 404, 406, 519, 521, 541
<code>\bfseries</code> .....	291– 294, 296, 297, 302–305, 313–318, 320–324
<code>\bgroup</code> .....	23, 87, 127, 164, 201, 204, 415, 455, 501, 525, 527

bib2gls .....	23, 61, 193		
booktabs package .....	296–299		
\boolean .....	263, 567		
\boolfalse .....	33, 422, 423		
\booltrue .....	33, 423		
\bottomrule .....	296, 297		
\box .....	300		
<b>C</b>			
\c .....	23, 416		
\c@equation .....	119, 476		
\c@glossaryentry .....	11, 409		
\c@glossarysubentry .....	12, 409		
\c@page .....	191, 192, 194, 520–522		
\capitalisewords .....	160		
\catcode .....	32, 78, 127, 449		
\cGls .....	103, 465		
\cglS .....	102, 465		
\cGlsformat .....	101, 464, 465		
\cglSformat .....	100, 464, 465		
\cGlspl .....	104, 466		
\cglSpl .....	103, 466		
\cGlsplformat .....	101, 464, 466		
\cglSplformat .....	100, 464, 466		
\char .....	222, 539		
classicthesis package .....	8		
\cleardoublepage .....	48, 431		
\clearpage .....	48, 431		
\closeout .....	32, 78, 173, 174, 179, 189, 422, 449, 507, 508, 512, 519		
\compatglossarystyle .....	341–353		
\compatibleglossentry .....	220, 537, 538		
\compatiblesubglossentry .....	220, 538		
\copy .....	299, 300		
\count@ .....	210, 532		
\csdef .....	21, 81–83, 93, 94, 99, 100, 204, 205, 224, 234, 340, 414, 451– 453, 459, 460, 463, 464, 527, 528, 540, 545		
\csedef .....	101, 191, 465, 520		
\csgdef .....	45, 65, 68, 100, 101, 199, 213, 429, 442, 443, 464, 465, 524, 534		
\cslet .....	72, 87, 94, 208, 445, 455, 460, 531		
\csname .....	13–16, 23, 37, 39, 41, 42, 47, 50, 51, 53, 54, 56, 58, 61, 66, 67, 74, 75, 81–83, 85, 86, 90, 91, 93–96, 98, 114, 118–120, 131–135, 149–156, 164, 166, 169–171, 176–179, 184, 188, 190, 191, 195–197, 200–203, 205, 213, 218, 220, 223, 224, 227, 247, 265–273, 280, 281, 330–332, 334–336, 348, 354, 355, 359, 360, 363, 367, 377–379, 401, 402, 410, 411, 416, 425–428, 430, 431, 433–435, 437–439, 442, 443, 447, 451–455, 457–463, 472, 473, 475, 476, 484–487, 493–497, 502, 505, 506, 509–512, 516, 518–520, 522, 523, 525–528, 534, 537–540, 542, 568–570		
\csshow .....	269, 570		
\csuse .....	42, 45, 65, 75, 82, 83, 114, 179–181, 205, 207, 209, 211, 212, 214, 215, 223, 234, 273, 341– 353, 428, 429, 442, 447, 448, 452, 453, 473, 512–514, 528, 530–533, 535, 540, 546		
\csxdef .....	92, 101, 247, 459, 465		
\currentglossary .....	11, 12, 45, 201, 409, 429, 526		
\currentglssubentry .....	12, 216, 409, 536		
\CurrentOption .....	36, 37, 272, 354, 425		
\CurrentTrackedLanguage .....	41, 403, 404, 427		
\CurrentTrackedTag .....	41, 403, 404, 427		
\CustomAcronymFields .....	264, 567, 568		
\CustomNewAcronymDef .....	264, 567, 568		
<b>D</b>			
\d .....	23, 416		
datatool package .....	206		
datatool-base package .....	3		
\day .....	169, 173, 336, 339, 504, 507		
\DeclareAcronymList .....	17, 19, 232, 234, 248, 250, 252, 255, 257, 260, 262, 264, 412, 413, 544, 545, 555, 557, 559, 560, 562, 565, 566, 568		
\DeclareCurrentRelease .....	3, 285		
\DeclareListParser .....	183, 515		
\DeclareOption .....	9, 272, 354, 408		
\DeclareOptionX .....	9, 408		
\DeclareRelease .....	3, 285		
\DeclareRobustCommand .....	42, 258, 428, 563		
\def .....	8, 9, 11–18, 22–24, 29–31, 34, 35, 37–39, 42, 43, 46, 49, 52–57, 61, 65– 67, 69–73, 76, 85, 87–92, 94, 96, 100– 104, 114, 115, 118–128, 130–156, 164, 165, 169, 173, 175–177, 179–181, 185, 186, 188, 191, 192, 194–197, 200, 201, 204, 208–210, 212, 213, 215, 222–227, 229–232, 247, 248, 250, 252–262, 264, 272, 275–278, 281–283, 310, 311, 330– 333, 335, 336, 339, 341, 348, 349, 354– 359, 377–384, 396–400, 408–413, 415,		

416, 420, 423–426, 428, 429, 432, 434–437, 439, 442–446, 448, 454–461, 464–466, 472, 473, 475–497, 502, 504, 507, 509–513, 515–517, 519–525, 527, 530–535, 539–544, 555, 557–560, 562–566, 568	\do@gl@xdycheckbackslash . . . . . 121, 477
\def@gl@xdycheckbackslash . . . . . 128, 129, 482, 483	\do@gl@disablehyperinlist . . . . . 118, 475
\DefaultNewAcronymDef . . . . . 248, 554, 555	\do@gl@haschildren . . . . . 61, 439
\defgl@display . . . . . 472	\do@write@gl@labelsname . . . . . 31, 32
\defgl@displayfirst . . . . . 473	doc package . . . . . 3, 4, 16
\defgl@sentryfmt . . . . . 66, 68, 114, 234, 246, 249, 251, 253, 255, 258, 261, 263, 442, 443, 472, 473, 545, 554, 556, 557, 559, 561, 563, 565, 567	\doifglossarynoexistsordo . . . 66, 439, 442
\define@boolkey . . . . . 7, 9, 11, 12, 17, 24, 25, 28, 29, 33, 115, 215, 407–409, 412, 416, 417, 420, 422, 423, 474, 535	\dtl@ifsingle . . . . . 222, 539
\define@choicekey . . . . . 4, 7, 8, 13, 25, 27, 29, 32, 35, 71, 214, 215, 405–407, 410, 417, 418, 420, 422, 423, 445, 534, 535	\dtl@insertinto . . . . . 205, 528
\define@key . . . . . 8, 12, 19, 25, 29, 30, 69–73, 81, 82, 115, 165, 213, 215, 272, 356–358, 408, 409, 413, 418, 420, 444–446, 451, 473, 502, 534, 535	\dtl@sortresult . . . . . 206, 528, 529
\DefineAcronymSynonyms . . 37, 246, 425, 554	\dtl@compare . . . . . 206, 529
\delimN . . . . . 172, 183, 212, 225, 338, 429, 506, 515, 533, 541	\dtl@iccompare . . . . . 206, 529
\delimR . . . . . 172, 225, 338, 429, 506, 540, 541	\DTLifinlist . . . . . 68, 118, 444, 475
\DescriptionDUANewAcronymDef 252, 558, 559	\DTLifint . . . . . 222, 539
\DescriptionFootnoteNewAcronymDef . . . . . 250, 556, 557	\dtl@letterindexcompare . . . . . 206, 529
\descriptionname . 42, 291–294, 296, 297, 302, 303, 305, 313–318, 320–324, 426, 428	\DTLsubstituteall . . . . . 121, 477
\DescriptionNewAcronymDef . . 255, 560, 561	\dtl@wordindexcompare . . . . . 206, 528
\DH . . . . . 24, 416	\DUANewAcronymDef . . . . . 262, 565, 566
\dh . . . . . 24, 416	
\dimen@ . . . . . 236, 299, 330, 547	
\disable@keys . . . . . 37, 425	
\do . . . . . 28, 36, 38, 50, 51, 58, 79, 80, 121, 164, 169–171, 181–183, 191, 198, 204, 234, 248, 251, 253, 255, 257, 260, 262, 264, 280, 281, 336, 419, 425, 426, 432, 433, 438, 449, 450, 477, 502, 504, 505, 514, 515, 520, 524, 527, 545, 555, 557, 559, 561, 562, 565, 567, 568	
\do@gl@storeentry . 13–15, 93, 410, 411, 459	
\do@gl@link@checkfirsthyper . . . . . 117, 118, 130–136, 149–155, 377–379, 474, 475, 484–487, 493–497	
	<b>E</b>
	\eappto . . . . . 443, 460, 520
	\edef . . . . . 38, 41, 49–54, 56, 66, 75, 78, 80, 114, 119–128, 167, 173, 175–177, 179–181, 183, 188, 195, 196, 203, 206, 210, 222, 226, 227, 246, 249, 251, 253, 256, 259, 261, 334, 335, 337, 339, 396–399, 411, 413, 426, 427, 432–439, 442–444, 447–450, 453–456, 460, 472, 473, 475–483, 502, 503, 507–513, 515, 516, 519, 521–524, 526–529, 532, 536, 539, 541, 542, 554, 556, 558, 560, 561, 564, 565
	\egroup . . . . . 23, 87, 165, 202, 205, 415, 455, 502, 526, 528
	\else . . . . . 5, 11, 12, 15–18, 20, 22, 24, 25, 31–34, 36–38, 42–44, 46–58, 72, 74, 90, 91, 94–96, 100, 101, 117–119, 121–128, 131–136, 157, 158, 168, 169, 172–175, 177–180, 188, 190, 192, 194–198, 201, 210, 215, 217, 222, 225, 226, 236, 250–252, 254, 255, 257, 258, 260, 261, 263, 265, 280, 284, 287, 290, 292, 293, 296, 297, 299–301, 303, 304, 312, 314, 316, 319, 321, 323, 326, 327, 329, 331, 332, 335–341, 346–349, 364, 406, 408, 409, 411–415, 417, 422, 423, 425, 426, 428–438, 445, 447, 457, 458, 460, 461, 464, 474–487, 498, 503, 504, 506–513, 519–525, 532, 535, 536, 539–541, 547, 557, 558, 561, 563, 565, 567, 568
	\emph . . . . . 197, 227, 524, 542



## G

- garamondx package ..... [228](#)
- \gdef ... [15](#), [32](#), [51](#), [66](#), [85](#), [90](#), [91](#), [190](#), [215](#),  
[280](#), [411](#), [433](#), [442](#), [454](#), [457](#), [458](#), [519](#), [535](#)
- \Genacrfullformat ... [112](#), [233](#), [235](#), [236](#),  
[242](#), [376](#), [386](#), [387](#), [392](#), [472](#), [544–547](#), [552](#)
- \genacrfullformat [112](#), [113](#), [233](#), [235](#), [236](#),  
[242](#), [376](#), [386](#), [387](#), [392](#), [472](#), [544–547](#), [552](#)
- \GenericAcronymFields .....  
..... [232](#), [235](#), [236](#), [238](#), [239](#), [241](#), [242](#),  
[244](#), [386–389](#), [391](#), [392](#), [394](#), [395](#), [544–554](#)
- \Genplacrfullformat .....  
..... [112](#), [233](#), [235](#), [236](#), [242](#),  
[376](#), [386](#), [387](#), [392](#), [471](#), [472](#), [545–547](#), [552](#)
- \genplacrfullformat .....  
..... [112](#), [113](#), [233](#), [235](#), [236](#), [242](#), [375](#),  
[376](#), [386](#), [387](#), [392](#), [471](#), [472](#), [545–547](#), [552](#)
- getttitlestring package ..... [286](#)
- \GetTitleStringDisableCommands .... [286](#)
- \GetTitleStringSetup ..... [286](#)
- \glo@desc ..... [341](#)
- \glo@do@compare ..... [206](#), [207](#), [528](#), [529](#)
- \glo@grabfirst ..... [211](#), [531](#), [532](#)
- \glo@label ..... [61](#), [94](#), [439](#), [460](#)
- \glo@list ..... [94](#), [460](#)
- \glo@name ..... [218](#), [537](#)
- \glo@parent ..... [61](#), [439](#)
- \glo@type ..... [94](#), [460](#)
- \glo@value ..... [79](#), [449](#)
- \global . [14–16](#), [74](#), [77](#), [87](#), [93](#), [98](#), [190](#), [202](#),  
[210](#), [211](#), [216](#), [300](#), [410](#), [411](#), [447](#), [449](#),  
[455](#), [459](#), [462](#), [463](#), [519](#), [526](#), [532](#), [533](#), [535](#)
- \glo@linkprefix .. [119](#), [165](#), [218](#), [475](#), [502](#), [537](#)
- \glosortentrieswarning .. [20](#), [204](#), [414](#), [527](#)
- glossaries package .....  
[34](#), [36](#), [56](#), [57](#), [168](#), [265](#), [272](#), [286](#), [334](#), [354](#)
- glossaries-accsupp package ..... [94](#), [354](#), [355](#)
- glossaries-extra package .....  
..... [78](#), [170](#), [178](#), [200](#), [274](#), [354](#), [363](#)
- \GlossariesWarning ..... [4](#), [5](#), [7](#), [20](#),  
[21](#), [24](#), [25](#), [30](#), [31](#), [34](#), [35](#), [45](#), [49](#), [60](#), [64](#),  
[71](#), [74](#), [77–79](#), [102–104](#), [114](#), [116](#), [164](#),  
[179–181](#), [184–187](#), [190](#), [196](#), [199](#), [201](#),  
[220](#), [223](#), [262](#), [334](#), [354](#), [405](#), [407](#), [413](#),  
[414](#), [416–418](#), [421](#), [423](#), [429](#), [431](#), [439](#),  
[441](#), [445](#), [447](#), [465](#), [466](#), [472–474](#), [501](#),  
[512–514](#), [516–518](#), [520](#), [523–525](#), [538](#), [540](#)
- \GlossariesWarningNoLine .....  
..... [4](#), [5](#), [20](#), [183](#), [185](#),  
[188](#), [203](#), [280](#), [405](#), [414](#), [515](#), [516](#), [519](#), [527](#)
- \glossary ..... [33–35](#), [335](#), [336](#), [423](#), [424](#)
- glossary package ..... [1](#), [34](#), [227](#)
- glossary styles:
  - altlist ..... [287](#), [288](#), [342](#)
  - altlistgroup ..... [288](#), [342](#)
  - altlisthypergroup ..... [288](#), [342](#)
  - altlong4col ..... [294](#), [295](#), [304](#), [344](#)
  - altlong4col-booktabs ..... [298](#), [299](#)
  - altlong4colborder ..... [295](#), [344](#)
  - altlong4colheader ..... [295](#), [298](#), [344](#)
  - altlong4colheaderborder .... [295](#), [344](#)
  - altlongragged4col ... [299](#), [304](#), [305](#), [345](#)
  - altlongragged4col-booktabs .... [299](#)
  - altlongragged4colborder .... [305](#), [345](#)
  - altlongragged4colheader .... [304](#), [345](#)
  - altlongragged4colheaderborder [305](#), [346](#)
  - altsuper4col ..... [317](#), [318](#), [322](#), [353](#)
  - altsuper4colborder ..... [317](#), [353](#)
  - altsuper4colheader ..... [317](#), [353](#)
  - altsuper4colheaderborder ... [318](#), [353](#)
  - altsuperragged4col ..... [322](#), [323](#), [351](#)
  - altsuperragged4colborder ... [323](#), [351](#)
  - altsuperragged4colheader ... [323](#), [351](#)
  - altsuperragged4colheaderborder .  
..... [323](#), [351](#)
  - alttree ..... [310](#), [325](#), [330](#), [332](#), [348](#)
  - alttreegroup ..... [333](#), [349](#)
  - alttreehypergroup ..... [333](#), [349](#)
  - index ..... [8](#), [306](#), [324–327](#), [346](#)
  - indexgroup ..... [326](#), [346](#)
  - indexhypergroup ..... [326](#), [346](#)
  - inline ..... [341](#)
  - list ..... [8](#), [10](#), [286–288](#), [341](#)
  - listdotted ..... [288](#), [289](#), [342](#)
  - listgroup ..... [287](#), [341](#)
  - listhypergroup ..... [287](#), [342](#)
  - long ..... [290](#), [291](#), [296](#), [300](#), [342](#), [344](#)
  - long-booktabs ..... [296](#), [298](#)
  - long3col ..... [291](#), [292](#), [297](#), [343](#)
  - long3col-booktabs ..... [296](#), [298](#)
  - long3colborder ..... [292](#), [343](#)
  - long3colheader ..... [292](#), [296](#), [343](#)
  - long3colheaderborder ..... [292](#), [343](#)
  - long4col ..... [293](#), [294](#), [297](#), [343](#)
  - long4col-booktabs ..... [297](#), [298](#)
  - long4colborder ..... [294](#), [344](#)
  - long4colheader ..... [294](#), [297](#), [344](#)



long4colheaderborder	294, 344	treehypergroup	328, 347
longborder	290, 343	treenoname	309, 325, 328, 329, 347
longheader	291, 296, 343	treenonamegroup	329, 348
longheaderborder	291, 343	treenonamehypergroup	329, 348
longragged	298, 300, 302	glossary-hypernav package	167
longragged-booktabs	298	glossary-list package	8, 10, 285
longragged3col	298, 302, 303, 345	glossary-long package	10, 289, 304, 311, 312
longragged3col-booktabs	298	glossary-longragged package	300
longragged3colborder	303, 345	glossary-mcols package	306
longragged3colheader	303, 345	glossary-super package	10, 289, 311, 318, 322
longragged3colheaderborder	303, 345	glossary-superragged package	318
longraggedborder	301, 344	glossary-tree package	10, 324
longraggedheader	302, 345	\glossaryentry	196, 197, 336, 522, 523
longraggedheaderborder	302, 345	glossaryentry (counter)	11, 12, 216, 217
mcolalmtree	310, 350	\glossaryentryfield	218, 341–348, 350–353, 386, 537, 538
mcolalmtreegroup	310, 350	\glossaryentrynumbers	9, 172, 201, 202, 211, 215, 216, 338, 408, 506, 525, 526, 532, 533, 535
mcolindex	306, 349	\glossaryheader	171, 209, 282, 284–288, 290–294, 296, 297, 301–305, 307–310, 312, 313, 315, 319, 320, 322, 325–330, 333, 338, 506, 531, 537
mcolindexgroup	306, 349	\glossarymark	46, 430
mcolindexhypergroup	306, 307, 349	\glossaryname	16, 41, 42, 411, 426–428
mcoltree	308, 349	\glossarypostamble	171, 209, 338, 429, 506, 531
mcoltreegroup	349	\glossary preamble	171, 209, 338, 429, 505, 531
mcoltreehypergroup	308, 349	\glossarysection	171, 209, 338, 429, 505, 531
mcoltreenoname	309, 350	\glossarystyle	539, 540
mcoltreenonamegroup	309, 350	glossarysubentry (counter)	12, 216, 217
mcoltreenonamehypergroup	309, 350	\glossarysubentryfield	220, 341–348, 350–353, 386, 538
sublistdotted	342	\glossarytitle	171, 200, 201, 209, 213, 338, 505, 525, 526, 531, 534
super	312, 313, 320, 352	\glossarytoctitle	8, 16, 17, 35, 36, 39, 42, 46, 171, 201, 209, 213, 214, 338, 407, 411, 412, 424, 426, 428, 430, 505, 525, 531, 534
super3col	313–315, 352	\glossentry	94, 202, 211, 220, 282, 284, 285, 287–291, 293, 301, 302, 304, 312, 314, 315, 319, 321, 322, 325, 327, 328, 331, 460, 526, 533, 538
super3colborder	314, 352	\Glossentrydesc	385, 537
super3colheader	314, 352	\glossentrydesc	282–285, 287–293, 301–304, 312, 314, 315, 319, 321–323, 326, 327, 329, 331, 332, 385, 537
super3colheaderborder	315, 352	\Glossentryname	537
super4col	315–317, 353		
super4colborder	316, 353		
super4colheader	316, 353		
super4colheaderborder	316, 353		
superborder	312, 352		
superheader	313, 352		
superheaderborder	313, 352		
superragged	319, 320, 350		
superragged3col	320–322, 351		
superragged3colborder	321, 351		
superragged3colheader	321, 351		
superragged3colheaderborder	322, 351		
superraggedborder	319, 350		
superraggedheader	320, 350		
superraggedheaderborder	320, 351		
tree	307, 327, 328, 330, 346		
treegroup	308, 328, 347		



<code>\glossentryname</code> .....	282–291,	<code>\gls@docdef@addlabel</code> .....	78
293, 301, 302, 304, 312, 314, 315, 319,		<code>\gls@docdef@labellist</code> .....	77
321, 322, 326, 327, 329, 331, 332, 385, 537		<code>\gls@docclearpage</code> .....	48, 431
<code>\Glossentrysymbol</code> .....	385, 538	<code>\gls@doentryfmt</code> .....	442
<code>\glossentrysymbol</code> .....		<code>\gls@dosubst</code> .....	121, 477
..... 282, 283, 293, 304, 315, 316,		<code>\gls@dotocitle</code> .	201, 202, 213, 525, 526, 534
322, 323, 326, 327, 329, 331, 332, 385, 538		<code>\gls@end@sanitizesort</code> .....	22, 23, 415
<code>\GLS</code> .....	485	<code>\gls@endcheck</code> .....	75, 76, 448
<code>\Gls</code> .....	103, 227, 246, 465, 484, 542, 554	<code>\gls@glossary</code> .....	
<code>\gls</code> .....	34, 102, 184, 217,	..	34, 35, 195–197, 423, 424, 519, 522, 523
227, 246, 423, 465, 484, 516, 536, 542, 554		<code>\gls@glsdefs@addlabel</code> .....	79
<code>\gls@accessibility</code> .....	356	<code>\gls@glsdefs@labellist</code> .....	77, 78
<code>\gls@accsupp@engine</code> .....	355	<code>\gls@gobbleopt</code> .....	67, 442, 443
<code>\gls@Alphpage</code> .....	191, 194, 520, 522	<code>\gls@grplabel</code> .....	279
<code>\gls@alphpage</code> .....	191, 194, 520, 522	<code>\gls@hypergroup prerun</code> .....	280
<code>\gls@arabicpage</code> .....	191, 194, 520, 521	<code>\gls@ifnotmeasuring</code> .....	97, 461, 462
<code>\gls@assign@desc</code> .....	87, 92, 455, 459	<code>\gls@inlinepostchild</code> .....	282, 283, 341
<code>\gls@assign@descplural</code> .	247, 256, 257,	<code>\gls@inlinesep</code> .....	282, 341
259–262, 396, 399, 400, 555, 562, 564–566		<code>\gls@inlinesubsep</code> .....	282, 283, 341
<code>\gls@assign@field</code> 76, 81, 82, 87, 89, 91–93,		<code>\gls@islistofacronyms</code> .....	18, 413
272, 273, 448, 451, 452, 455, 456, 458, 459		<code>\gls@istfilebase</code> .....	
<code>\gls@assign@firstpl</code> .....		.....	42, 43, 179, 180, 428, 512, 513
.... 247, 250, 252, 254, 256, 257, 259–		<code>\gls@label</code> .....	227, 542
262, 396–400, 555–558, 560, 562, 564, 566		<code>\gls@level</code> ....	90, 91, 211, 447, 457, 458, 532
<code>\gls@assign@plural</code> .....		<code>\gls@noidxglossary</code> .....	184, 516, 518
. 247, 250, 252, 254, 256, 257, 259–262,		<code>\gls@nonumberlist@nr</code> .....	71, 445
396–400, 555, 557, 558, 560, 562, 564–566		<code>\gls@nonumberlist@val</code> .....	71, 445
<code>\gls@assign@symbolplural</code> .....		<code>\gls@nosetquote</code> .....	
..... 247, 250, 252, 254, 259–262,		.....	88, 173, 175, 177, 456, 507, 508, 511
396–398, 400, 555, 557, 558, 560, 564–566		<code>\gls@number</code> .....	194, 521
<code>\gls@automake@nr</code> .....	32, 181, 422, 514	<code>\gls@numberedsection@nr</code> ..	8, 214, 407, 534
<code>\gls@automake@val</code> .....	32, 422	<code>\gls@numberedsection@val</code> .	8, 214, 407, 534
<code>\gls@begindocdefs</code> .....	78, 449	<code>\gls@numberpage</code> .....	191, 194, 520, 521
<code>\gls@checkisacronymlist</code> ....	117, 413, 474	<code>\gls@org@glossaryentryfield</code> ...	202, 526
<code>\gls@checkseeallowed</code> .....		<code>\gls@org@glossarysubentryfield</code>	202, 526
..... 71, 182, 184, 445, 448, 515, 516		<code>\gls@org@insert</code> .....	
<code>\gls@checkseeallowed@preambleonly</code> .		.....	253, 255, 256, 258, 559, 561, 563
..... 445, 448		<code>\gls@orgAlph</code> .....	194, 521, 522
<code>\gls@codepage</code> .....		<code>\gls@orgalph</code> .....	194, 521, 522
. 57, 179, 180, 203, 420, 437, 512, 513, 527		<code>\gls@orgarabic</code> .....	194, 521
<code>\gls@debug@nr</code> .....	4, 35, 405, 423	<code>\gls@orgnumber</code> .....	194, 521
<code>\gls@debug@val</code> .....	4, 35, 405, 423	<code>\gls@orgRoman</code> .....	194, 521, 522
<code>\gls@defdocnewglossaryentry</code> .....		<code>\gls@orgromannumeral</code> .....	194, 521
..... 78, 99, 448, 449, 463		<code>\gls@orgthe</code> .....	194, 521
<code>\gls@defglossaryentry</code> .....		<code>\gls@original@glossary</code> .....	35, 423, 424
..... 76–79, 87, 448, 449, 455, 456		<code>\gls@original@makeglossary</code> ..	35, 423, 424
<code>\gls@disablepagerefexpansion</code> .....		<code>\gls@protected@pagefmts</code>	121, 191, 477, 520
..... 190, 194, 520, 522		<code>\gls@Romanpage</code> .....	191, 194, 520, 522
<code>\gls@do@addxdyattribute</code> .....	51, 433	<code>\gls@romanpage</code> .....	191, 194, 520, 521

<code>\gls@save@numberlist</code> .....	9, 408, 524	<code>\GlsAddXdyStyle</code> .....	436, 437
<code>\gls@seen@index@nr</code> .....	7, 406	<code>\glsautomakefalse</code> ....	32, 33, 181, 422, 514
<code>\gls@seen@index@val</code> .....	7, 406	<code>\glsautomaketrue</code> .....	32, 422
<code>\gls@set@xr@key</code> .....	71, 445	<code>\glsautoprefix</code> .....	8, 214, 407, 534
<code>\gls@suffixF</code> .....		<code>\glsbackslash</code> .....	503
.....	44, 172, 174, 338, 340, 429, 506, 508	<code>\glscapitalisewords</code> .....	160
<code>\gls@suffixFF</code> .....		<code>\glscapspace</code> .....	105,
.....	44, 172, 174, 338, 340, 429, 506, 508		107, 109–112, 130–135, 149–155, 240,
<code>\gls@text</code> .....	113, 472		253, 259, 367, 369, 372–379, 390, 466,
<code>\gls@the</code> .....	194, 521		468–472, 484–486, 493–497, 550, 559, 563
<code>\gls@thissty</code> .....	28, 419	<code>\glscategory</code> .....	363
<code>\gls@tmp</code> .....	188, 258, 519, 563	<code>\glsclearpage</code> .....	47, 430, 431
<code>\gls@tmplen</code> ....	129, 330–332, 348, 349, 483	<code>\glsclsebrace</code> .....	
<code>\gls@tr@set@acronym@toctitle</code> ...	17, 412	.....	54, 172, 338, 339, 435, 503, 506, 507
<code>\gls@tr@set@main@toctitle</code> ....	16, 411	<code>\glscompositor</code> .....	
<code>\gls@tr@set@numbers@toctitle</code> ...	36, 424	...	43, 44, 54, 174, 337, 340, 429, 435, 508
<code>\gls@tr@set@symbols@toctitle</code> ...	35, 424	<code>\glscounter</code> .....	19, 38, 49,
<code>\gls@translate@nr</code> .....	27, 419		67, 91, 119, 334, 413, 426, 432, 442, 458, 475
<code>\gls@translate@val</code> .....	27, 419	<code>\glscurrententrylabel</code> ..	199, 202, 524, 526
<code>\gls@wrglossary</code> .....	190, 519	<code>\glscurrentfieldvalue</code> .....	63, 64, 441
<code>\gls@xdystring</code> .....	121, 477	<code>\glscustomtext</code> ..	105, 108, 109, 111, 113,
<code>\gls@xindy@glsnumbersfalse</code> ....	30, 421		130–135, 149–156, 240, 241, 249, 253,
<code>\gls@xindy@glsnumberstrue</code> .....	29, 420		255, 256, 258, 259, 367, 371, 374, 376–
<code>\gls@xr@key</code> .....	6, 7, 71, 406, 407, 445		379, 389, 391, 466, 469, 471, 472, 484–
<code>\glsaccessibility</code> .....	363, 364		486, 493–497, 550, 556, 559, 561, 563, 564
<code>\glsaccsupp</code> .....	363, 364	<code>\GlsDeclareNoHyperList</code> .....	19, 413
<code>\glsacronymtrue</code> .....	17, 412	<code>\glsdefaultshortaccess</code> .....	395–399
<code>\glsacrpluralsuffix</code> ..	40, 228, 237, 241,	<code>\glsdefaulttype</code> .....	16, 45, 56, 58,
	242, 244, 248, 426, 542, 548, 551–553, 555		65, 66, 89, 104, 114, 188, 200, 412, 429,
<code>\glsacrshortcutsfalse</code> .....	37, 425		437, 438, 442, 456, 466, 472, 473, 519, 525
<code>\glsacrshortcutstrue</code> .....	37, 425	<code>\glsdefmain</code> .....	16, 68, 411, 412, 443
<code>\glsacspace</code> ....	235, 236, 238, 546, 547, 549	<code>\glsdefs@newdocentry</code> .....	79
<code>\glsadd</code> .....	34, 166, 423, 502, 503	<code>\GLSdesc</code> .....	489
<code>\glsadd options</code>		<code>\Glsdesc</code> .....	489
counter .....	165	<code>\glsdesc</code> .....	489
format .....	165, 224	<code>\GLSdescplural</code> .....	489
<code>\glsaddall</code> .....	503	<code>\Glsdescplural</code> .....	489
<code>\glsaddall options</code>		<code>\glsdescplural</code> .....	489
types .....	165, 166	<code>\glsdescriptionaccessdisplay</code> .....	
<code>\glsaddallunused</code> .....	503	.....	369–371, 381, 385, 386
<code>\glsaddkey</code> .....	451	<code>\glsdescriptionpluralaccessdisplay</code>	
<code>\GlsAddLetterGroup</code> .....	438	.....	368, 369, 381, 382
<code>\glsaddprotectedpagefmt</code> .....	520	<code>\glsdescwidth</code> .....	290–292,
<code>\GlsAddSortRule</code> .....	436		295, 298, 299, 301–305, 312–315, 317–324
<code>\glsaddstoragekey</code> .....	451	<code>\glsdetoklabel</code> ..	31, 59–64, 72, 79, 84–88,
<code>\GlsAddXdyAlphabet</code> .....	434		94, 98, 100, 101, 118, 156, 157, 164–166,
<code>\GlsAddXdyAttribute</code> ..	50–52, 334, 335, 432–434		184–186, 192, 195, 202, 205, 206, 210,
<code>\GlsAddXdyCounters</code> ..	50, 51, 67, 432, 433, 443		211, 213, 216–218, 265–269, 273, 274,
<code>\GlsAddXdyLocation</code> .....	435, 436		

330, 354, 355, 364–367, 401, 402, 438–	\glentryfirstplural .....
441, 445, 449, 453–456, 460, 462–465,	. 103, 105, 106, 109, 110, 139, 368, 369,
475, 497, 498, 502, 516, 517, 521, 522,	372, 373, 380, 466, 467, 469, 470, 488, 499
526, 528, 529, 532, 534, 536, 537, 568–570	\glentryfirstpluralaccess .....
\gldisablehyper .....	365
\gldisp .....	\glentryfmt .....
486	66, 68, 442, 443, 466
\gldisplay .....	\Glsentryfull .....
105, 114, 466, 472, 473	233, 241, 243, 391, 393, 501, 545, 551, 553
\gldisplayfirst ...	\glentryfull .....
105, 114, 466, 472, 473	233, 241, 243, 391, 393, 501, 545, 551, 552
\gldisplaynumberlist .....	\Glsentryfullpl .....
186, 501, 517	233, 241, 243, 391, 394, 501, 545, 551, 553
\glsdohyperlink .....	\glentryfullpl .....
129, 130, 483, 484	233, 241, 243, 391, 393, 501, 545, 551, 553
\glsdohypertarget .....	\glentryitem ...
129, 130, 483, 484	282, 284–291, 293, 301,
\glsdoidexists 61–63, 84–87, 97, 130–136,	302, 304, 312, 314, 315, 319, 321, 322,
148–155, 160, 164, 165, 185, 186, 275–	326, 327, 329, 331, 341–348, 350–353, 536
278, 377–379, 385, 438–440, 453–455,	\Glsentrylong .....
462, 484–487, 493–497, 499, 501, 502, 517	103, 153,
\glsdoidexistsordo .	157, 163, 235, 240, 241, 376, 379, 386,
117, 156, 439, 474, 497	390, 391, 465, 495, 498, 501, 546, 550, 551
\glsdoidexistsorwarn .....	\glentrylong .....
213, 218, 219, 439, 533, 537, 538	. 102, 113, 152, 154, 157, 163, 235, 236,
\glsdoidnoexists .....	238–244, 256, 376, 378, 379, 386–395,
76, 87, 439, 448, 455	465, 472, 495, 496, 498, 501, 546–554, 561
\glsdonohyperlink .....	\glentrylongaccess .....
119, 129, 475, 483	366
\glsdoparenifnotempty .....	\Glsentrylongpl .....
563	104,
\glsdosanitizesort .....	155, 163, 235, 236, 240, 241, 377, 386,
13, 410	390, 391, 466, 497, 501, 546, 547, 550, 551
\glsenableentrycount .....	\glentrylongpl .....
463, 465	103, 113,
\glsenablehyper .....	154, 156, 163, 235, 236, 240–243, 256,
484	263, 377, 386, 387, 390–394, 466, 472,
\glentryaccess .....	496, 497, 501, 546, 547, 550–553, 561, 567
364	\glentrylongpluralaccess .....
\glentrycounter ...	366
223, 226, 475, 539, 541	\Glsentryname 140, 219, 381, 385, 489, 498, 537
\glentrycounterfalse .....	\glentryname 140, 330, 381, 385, 488, 489, 498
12, 409	\glentrynumberlist ....
\glentrycounterlabel .....	164, 185, 501, 516
217, 536	\glentryparent .....
\GlsEntryCounterLabelPrefix .....	500
216, 217, 535, 536	\Glsentryplural .....
\glentrycountertrue .....	106,
12, 409	109, 138, 368, 372, 380, 467, 470, 488, 499
\glentrycurrcount .....	\glentryplural .
100–102, 463, 465	105, 106, 109, 138, 368,
\Glsentrydesc 141, 219, 381, 385, 489, 498, 537	369, 372, 380, 466, 467, 469, 470, 488, 499
\glentrydesc ...	\glentrypluralaccess .....
107, 108, 140, 141, 219,	365
369–371, 381, 385, 468, 469, 489, 498, 537	\Glsentryprefix .....
\glentrydescaccess .....	276
365	\glentryprefix .....
\Glsentrydescplural ....	275, 277
141, 381, 489, 498	\Glsentryprefixfirst .....
\glentrydescplural ....	276
105, 106, 141,	\glentryprefixfirst .....
142, 368, 369, 381, 382, 466, 467, 489, 498	275, 278
\glentrydescpluralaccess .....	\Glsentryprefixfirstplural .....
366	277
\Glsentryfirst .....	\glentryprefixfirstplural ....
103, 108, 110,	275, 278
137, 370, 373, 380, 465, 468, 470, 487, 499	\Glsentryprefixplural .....
\glentryfirst .....	277
. 102, 107, 108, 110, 137, 138, 370, 371,	\glentryprefixplural .....
373, 374, 380, 465, 468–470, 487, 488, 499	275, 278
\glentryfirstaccess .....	
365	
\Glsentryfirstplural ...	
104, 106, 109,	
139, 368, 372, 380, 466, 467, 470, 488, 499	

<code>\glsentryprevcount</code> . . . . .	100, 101, 463, 464	<code>\Glsentryuseri</code> . . . . .	144, 383, 490, 500
<code>\Glsentryshort</code> . . . . .	112, 149, 157, 158, 236, 242, 243, 375–377, 387, 392, 393, 471, 493, 498, 501, 547, 552, 553	<code>\glsentryuseri</code> . . . . .	144, 382, 383, 490, 491, 500
<code>\glsentryshort</code> . . . . .	111–113, 149, 150, 157, 158, 163, 233, 235, 236, 238, 239, 241–244, 375–378, 386–389, 391– 395, 471, 472, 493, 498, 500, 501, 545–554	<code>\glsentryuseriaccess</code> . . . . .	366
<code>\glsentryshortaccess</code> . . . . .	366	<code>\Glsentryuserii</code> . . . . .	145, 383, 491, 500
<code>\Glsentryshorttpl</code> . . . . .	111, 151, 236, 242, 243, 374, 387, 393, 394, 471, 494, 501, 547, 552, 553	<code>\glsentryuserii</code> . . . . .	144, 145, 383, 491, 500
<code>\glsentryshorttpl</code> . . . . .	111, 113, 150, 152, 163, 235, 236, 241–243, 263, 374, 375, 377, 386, 387, 391–394, 471, 472, 494, 495, 501, 546, 547, 551–553, 567	<code>\glsentryuseriiaccess</code> . . . . .	366
<code>\glsentryshortpluralaccess</code> . . . . .	366	<code>\Glsentryuseriii</code> . . . . .	146, 383, 491, 500
<code>\glsentrysort</code> . . . . .	500	<code>\glsentryuseriii</code> . . . . .	145, 146, 383, 491, 500
<code>\Glsentrysymbol</code> . . . . .	142, 219, 382, 385, 490, 499, 538	<code>\glsentryuseriiiaccess</code> . . . . .	367
<code>\glsentrysymbol</code> . . . . .	107, 108, 142, 143, 219, 249, 253, 258, 369–371, 382, 385, 468, 469, 490, 499, 538, 556, 559, 563	<code>\Glsentryuseriv</code> . . . . .	146, 384, 492, 500
<code>\glsentrysymbolaccess</code> . . . . .	365	<code>\glsentryuseriv</code> . . . . .	146, 147, 384, 491, 492, 500
<code>\Glsentrysymbolplural</code> . . . . .	143, 382, 490, 499	<code>\glsentryuserivaccess</code> . . . . .	367
<code>\glsentrysymbolplural</code> . . . . .	105, 106, 143, 249, 253, 258, 368, 369, 382, 466, 467, 490, 499, 556, 559, 563		
<code>\glsentrysymbolpluralaccess</code> . . . . .	365	<code>\Glsentryuseri</code> . . . . .	144, 383, 490, 500
<code>\Glsentrytext</code> . . . . .	107, 110, 137, 370, 373, 379, 468, 470, 487, 499	<code>\glsentryuseri</code> . . . . .	144, 382, 383, 490, 491, 500
<code>\glsentrytext</code> . . . . .	107, 108, 110, 136, 137, 165, 198, 369, 370, 373, 374, 379, 468–470, 487, 499, 502, 524	<code>\glsentryuseriaccess</code> . . . . .	366
<code>\glsentrytextaccess</code> . . . . .	365	<code>\Glsentryuserii</code> . . . . .	145, 383, 491, 500
<code>\glsentrytitlecase</code> . . . . .	499, 500	<code>\glsentryuserii</code> . . . . .	144, 145, 383, 491, 500
<code>\glsentrytype</code> . . . . .	89, 456, 500	<code>\glsentryuseriiaccess</code> . . . . .	366
<code>\Glsentryuseri</code> . . . . .	144, 383, 490, 500	<code>\Glsentryuseriii</code> . . . . .	146, 383, 491, 500
<code>\glsentryuseri</code> . . . . .	144, 382, 383, 490, 491, 500	<code>\glsentryuseriii</code> . . . . .	145, 146, 383, 491, 500
<code>\glsentryuseriaccess</code> . . . . .	366	<code>\glsentryuseriiiaccess</code> . . . . .	367
<code>\Glsentryuserii</code> . . . . .	145, 383, 491, 500	<code>\Glsentryuseriv</code> . . . . .	146, 384, 492, 500
<code>\glsentryuserii</code> . . . . .	144, 145, 383, 491, 500	<code>\glsentryuseriv</code> . . . . .	146, 147, 384, 491, 492, 500
<code>\glsentryuseriiaccess</code> . . . . .	366	<code>\glsentryuserivaccess</code> . . . . .	367
<code>\Glsentryuseriii</code> . . . . .	146, 383, 491, 500		
<code>\glsentryuseriii</code> . . . . .	145, 146, 383, 491, 500	<code>\Glsentryuseri</code> . . . . .	147, 384, 492, 500
<code>\glsentryuseriiiaccess</code> . . . . .	367	<code>\glsentryuseri</code> . . . . .	147, 384, 492, 500
<code>\Glsentryuseriv</code> . . . . .	146, 384, 492, 500	<code>\glsentryuseriaccess</code> . . . . .	367
<code>\glsentryuseriv</code> . . . . .	146, 147, 384, 491, 492, 500	<code>\Glsentryuseri</code> . . . . .	148, 384, 492, 500
<code>\glsentryuserivaccess</code> . . . . .	367	<code>\glsentryuseri</code> . . . . .	148, 384, 492, 500
		<code>\glsentryuseriaccess</code> . . . . .	367
		<code>\glsesclocationsfalse</code> . . . . .	184, 516
		<code>\glsesclocationstrue</code> . . . . .	9, 408
		<code>\glsexpandfields</code> . . . . .	448
		<code>\glsfieldaccsupp</code> . . . . .	363, 364
		<code>\glsfielddef</code> . . . . .	454
		<code>\glsfieldedef</code> . . . . .	453
		<code>\glsfieldfetch</code> . . . . .	160, 454, 499
		<code>\glsfieldgdef</code> . . . . .	453
		<code>\glsfieldxdef</code> . . . . .	453
		<code>\GLSfirst</code> . . . . .	488
		<code>\Glsfirst</code> . . . . .	487
		<code>\glsfirst</code> . . . . .	487
		<code>\glsfirstaccessdisplay</code> . . . . .	370, 371, 373, 374, 380
		<code>\GLSfirstplural</code> . . . . .	488
		<code>\Glsfirstplural</code> . . . . .	488
		<code>\glsfirstplural</code> . . . . .	488
		<code>\glsfirstpluralaccessdisplay</code> . . . . .	368, 369, 372, 380
		<code>\glsfirstpluralacesdisplay</code> . . . . .	373
		<code>\glsгенacfmt</code> . . . . .	235, 236, 242, 386, 387, 392, 471, 546, 547, 552
		<code>\glsгенentryfmt</code> . . . . .	235, 236, 241, 242, 246, 249, 251, 253, 255, 256, 258, 261, 263, 386, 387, 390, 392, 469, 546, 547, 550, 552, 554, 556, 557, 559, 561, 563, 565, 567
		<code>\glsgetgrouplabel</code> . . . . .	539
		<code>\glsgetgrouptitle</code> . . . . .	281, 284–288, 306–311, 326–329, 333, 538
		<code>\glsгlossarymark</code> . . . . .	46, 430
		<code>\glsгroupheading</code> . . . . .	172, 211, 282, 284, 285, 287, 288, 290, 291, 293, 301, 302, 304, 306–312, 314, 315, 319, 320, 322, 325–330, 332, 333, 339, 507, 533, 538
		<code>\glsгroupskip</code> . . . . .	171, 172, 211, 283, 284, 287, 290, 292, 293, 296, 297, 301, 303, 304, 312, 314, 316, 319, 321, 323, 326, 327, 329, 332, 338, 506, 507, 532, 538
		<code>\glshyperfirstfalse</code> . . . . .	242, 392, 552
		<code>\glshyperfirsttrue</code> . . . . .	28, 420
		<code>\glshyperlink</code> . . . . .	198, 502, 524
		<code>\glshypernavsep</code> . . . . .	281
		<code>\glshypernumber</code> . . . . .	45, 226, 227, 429, 540–542

<code>\glsifhyper</code> .....	474	<code>\glslistexpandedname</code> .....	286
<code>\glsifhyperon</code> .....	116, 474, 475	<code>\glslistgroupheaderfmt</code> .....	284–288
<code>\glsIfListOfAcronyms</code> .....	18, 19, 413	<code>\glslistinit</code> .....	286
<code>\glsifplural</code> .....		<code>\glslistnavigationitem</code> .	284, 285, 287, 288
..... 105, 109, 111, 112, 130–135, 149–		<code>\glslocalreset</code> .....	98, 462, 463
155, 240, 249, 253, 256, 258, 367, 371,		<code>\glslocalresetall</code> .....	463
374, 375, 377–379, 389, 466, 469, 471,		<code>\glslocalunset</code> 99, 131–136, 462, 463, 484–487	
484–486, 493–497, 550, 556, 559, 561, 563		<code>\glslocalunsetall</code> .....	463
<code>\glsifusedtranslatordict</code> .....	418	<code>\glslongaccessdisplay</code> 376, 378, 379, 386–396	
<code>\glsifusetranslator</code> .....		<code>\glslongkey</code> .....	400, 542
..... 26, 27, 41, 403, 418, 419, 427, 428		<code>\glslongpluralaccessdisplay</code> .....	
<code>\glsignore</code> .....	503	..... 377, 386, 387, 390–394, 396	
<code>\glsindexonlyfirstfalse</code> .....	28, 420	<code>\glslongpluralkey</code> .....	400, 542
<code>\glsinlinedescformat</code> .....	282, 341	<code>\glslongtok</code> .....	232,
<code>\glsinlinedopostchild</code> .....	282, 341	233, 235, 236, 241, 242, 246–252, 254–	
<code>\glsinlineemptydescformat</code> .....	282, 341	257, 259–264, 386, 387, 391, 392, 395–	
<code>\glsinlinenameformat</code> .....	282, 341	400, 544–547, 551, 552, 554–562, 564–568	
<code>\glsinlineparentchildseparator</code> 282, 341		<code>\glsLTpenaltycheck</code> .....	300
<code>\glsinlinepostchild</code> .....	282, 341	<code>\glsmcols</code> .....	306–311
<code>\glsinlineseparator</code> .....	282, 341	<code>\glsmoveentry</code> .....	460
<code>\glsinlinesubdescformat</code> .....	283, 341	<code>\GLSname</code> .....	489
<code>\glsinlinesubnameformat</code> .....	283, 341	<code>\Glsname</code> .....	488
<code>\glsinlinesubseparator</code> .....	283, 341	<code>\glsname</code> .....	488
<code>\glsinsert</code> .....	105–113, 130–135, 149–	<code>\glsnameaccessdisplay</code> .....	381, 385, 386
155, 240, 241, 249, 253, 255, 256, 258,		<code>\glsnamefont</code> .....	
259, 368–379, 390, 391, 466–472, 484–		..... 218–220, 354, 355, 385, 537, 538, 540	
486, 493–497, 550, 556, 559, 561, 563, 564		<code>\glsnavhyperlink</code> .....	281
<code>\glskeylisttok</code> .....	232,	<code>\glsnavhyperlinkname</code> .....	279, 280
247–250, 252, 254–257, 259–262, 264,		<code>\glsnavhypertarget</code> .....	
395–400, 544, 555–560, 562, 564–566, 568		..... 284–288, 307–311, 327–329, 333	
<code>\glslabel</code> .....	88, 105–113, 117–119,	<code>\glsnavigation</code> .....	
149–155, 235, 236, 240, 242, 249, 253,		..... 284–288, 307–311, 326, 328, 329, 333	
255, 256, 258, 368–376, 386, 387, 389,		<code>\glsnextpages</code> ..	9, 72, 201, 408, 445, 526, 535
390, 392, 456, 466–472, 474, 475, 493–		<code>\glsnoexpandfields</code> .....	448
497, 546, 547, 550, 552, 556, 559, 561, 563		<code>\glsnogroupskipfalse</code> .....	11, 409
<code>\glslabeltok</code> ....	232, 246–253, 255–257,	<code>\glsnoidxdisplayloc</code> 186, 188, 517, 518, 533	
259–264, 396–399, 544, 554–562, 564–568		<code>\glsnoidxdisplayloclisthandler</code> ....	
<code>\glsletentryfield</code> .....	497	..... 186, 517, 533	
<code>\glslink</code> .....	233, 241–243,	<code>\glsnoidxloclist</code> ...	185, 211, 517, 532, 533
248, 391, 393, 474, 544, 545, 551, 552, 555		<code>\glsnoidxloclisthandler</code> .....	212, 533
<code>\glslink options</code>		<code>\glsnoidxnumberlistloophandler</code> ....	
counter .....	115, 130, 271	..... 186, 517, 518	
format .....	115, 130, 224	<code>\glsnoidxstripaccents</code> .....	23, 415, 416
hyper .....	115, 117, 118, 130	<code>\glsnomakeindexwarning</code> .....	175, 431, 508
local .....	115	<code>\glsnonextpages</code> .....	71, 201, 445, 526, 535
<code>\glslinkcheckfirsthyperhook</code> ...	118, 475	<code>\glsnopostdotfalse</code> .....	11, 408
<code>\glslinkpostsetkeys</code> .....	119, 475	<code>\glsnoxindywarning</code> .....	44, 50–52, 55–
<code>\glslinkvar</code> .....	116, 474	57, 168, 429, 431–434, 436, 437, 503, 504	
<code>\glslistdottedwidth</code> .....	285, 289, 342	<code>\glsnumberformat</code> .....	164, 429, 502

<code>\glsnumberlistloop</code> .....	186, 517, 518	<code>\gls.savewritesfalse</code> .....	33, 422
<code>\glsnumbersgroupname</code> .....		<code>\glssee</code> .....	524
.....	36, 42, 222, 424, 426, 428, 539	<code>\glsseeformat</code> .....	
<code>\glsnumlistlastsep</code> .....	164, 186, 502, 517	.....	170, 184, 186, 338, 505, 516–518, 524
<code>\glsnumlistparser</code> .....	165, 183, 502, 515	<code>\glsseeitem</code> .....	198, 524
<code>\glsnumlistsep</code> .....	164, 186, 502, 517	<code>\glsseeitemformat</code> .....	198, 524
<code>\glsopenbrace</code> .....		<code>\glsseelastsep</code> .....	198, 524
.....	54, 172, 338, 339, 435, 503, 506, 507	<code>\glsseelist</code> .....	197, 524
<code>\glsorder</code> .....	29, 179–182, 207, 420, 512–514, 530	<code>\glsseesep</code> .....	198, 524
<code>\glsorg@endtheglossary</code> .....	4, 404, 405	<code>\glsSetAlphaCompositor</code> .....	429
<code>\glsorg@PrintChanges</code> .....	4, 404, 405	<code>\glsSetCompositor</code> .....	429
<code>\glsorg@theglossary</code> .....	4, 404	<code>\glssetexpandfield</code> .....	
<code>\glspagelistwidth</code> ...	291, 292, 295, 298,	.....	21, 24–26, 414, 415, 417, 418
	299, 302–305, 313–315, 317, 318, 320–324	<code>\glssetnoexpandfield</code> .....	21, 22, 24, 25, 414–417
<code>\glspar</code> .....	428	<code>\GlsSetQuote</code> .....	88, 173, 456, 507, 508, 511
<code>\glspatchLTOoutput</code> .....	296–299	<code>\glsSetSuffixF</code> .....	429
<code>\glspatchtabularx</code> .....	461	<code>\glsSetSuffixFF</code> .....	429
<code>\glspenaltygroupskip</code> .....	296, 297	<code>\glssettoctitle</code> .....	41, 201, 426, 428, 525
<code>\glspersentchar</code> .....		<code>\GlsSetWriteIstHook</code> .....	508
.....	79, 171, 173, 449, 450, 503, 506, 507	<code>\GlsSetXdyCodePage</code> .....	437
<code>\GLSpl</code> .....	486	<code>\GlsSetXdyFirstLetterAfterDigits</code> ..	503
<code>\Glspl</code> .....	104, 246, 466, 485, 554	<code>\GlsSetXdyLanguage</code> .....	437
<code>\glspl</code> .....	103, 246, 466, 485, 554	<code>\GlsSetXdyLocationClassOrder</code> .....	436
<code>\GLSplural</code> .....	488	<code>\GlsSetXdyMinRangeLength</code> .....	504
<code>\Glsplural</code> .....	488	<code>\GlsSetXdyNumberGroupOrder</code> ....	503, 504
<code>\glsplural</code> .....	488	<code>\GlsSetXdyStyles</code> .....	437
<code>\glspluralaccessdisplay</code> .....	368, 369, 372, 380	<code>\glsshortaccessdisplay</code> .....	
<code>\glspluralsuffix</code> .....	40, 91,	.....	158, 375–378, 386–389, 391–396, 498
	235–237, 387, 392–394, 426, 458, 546, 547	<code>\glsshortaccsupp</code> .....	364
<code>\glspostdescription</code> ..	42, 283, 284, 287,	<code>\glsshortkey</code> .....	400, 542
	288, 290, 301, 312, 319, 326, 327, 329,	<code>\glsshortpluralaccessdisplay</code> .....	
	331, 332, 341–344, 346–350, 352, 408, 428	.....	374, 375, 377, 386, 387, 391–394, 396
<code>\glspostinline</code> .....	282	<code>\glsshortpluralkey</code> .....	400, 542
<code>\glspostlinkhook</code> .....	117, 131–136,	<code>\glsshorttok</code> .....	232, 246–252, 254–257, 259–
	149–156, 377–379, 474, 484–487, 493–497		264, 395–400, 544, 554–560, 562, 564–568
<code>\glsprefixsep</code> .....	275–278	<code>\glsshowaccsupp</code> .....	5, 405, 406
<code>\glsprestandardsort</code> .....	13, 410	<code>\glsshowtarget</code> .....	6, 406
<code>\glsquote</code> .....	503	<code>\glsshowtargetfont</code> .....	5, 406
<code>\glsrefentry</code> .....	536	<code>\glsshowtargetouter</code> .....	5, 406
<code>\glsreset</code> .....	98, 462, 463	<code>\glsshowtargetsymbols</code> .....	5, 406
<code>\glsresetall</code> .....	463	<code>\glssortnumberfmt</code> .....	14, 15, 410, 411
<code>\glsresetentrycounter</code> .....	216, 536	<code>\glsspace</code> .....	229, 247, 248, 542
<code>\glsresetentrylist</code> .....		<code>\glsstepentry</code> .....	217, 536
.....	171, 209, 215, 338, 506, 531, 535	<code>\glsstepsubentry</code> .....	217, 536, 537
<code>\glsresetsubentrycounter</code> .....		<code>\glssubentrycounterfalse</code> .....	12, 409
.....	217, 282, 341, 535, 536	<code>\glssubentrycounterlabel</code> ...	217, 536, 537
<code>\glssanitizesortfalse</code> .....	25, 417	<code>\glssubentryitem</code> .....	
<code>\glssanitizesorttrue</code> .....	25, 417	.....	283–285, 287–290, 292, 293, 301,
<code>\glssavenumberlistfalse</code> .....	9, 408		



303, 304, 312, 314, 315, 319, 321, 323, 326, 327, 329, 331, 341–348, 350–353, 536	\GlsUseAcrEntryDispStyle .....
\GLSsymbol .....	..... 234, 237–239, 242–244, 388, 389, 392, 394, 395, 545–551, 553, 554
\Glsymbol .....	\GlsUseAcrStyleDefs .....
\glssymbol .....	.... 234, 237–239, 242–244, 388, 389, 392, 394, 395, 545, 546, 548–551, 553, 554
\glssymbolaccessdisplay .....	\GLSuseri .....
..... 369–371, 382, 385, 386	\Glsuseri .....
\GLSsymbolplural .....	\glssuseri .....
\Glsymbolplural .....	\glssuseriaccessdisplay .....
\glssymbolplural .....	382, 383
\glssymbolpluralaccessdisplay .....	\GLSuserii .....
..... 368, 369, 382	\Glsuserii .....
\glssymbolsgroupname .....	\glssuserii .....
..... 35, 42, 222, 424, 426, 428, 539	\glssuseriiaccessdisplay .....
\glstarget .....	383
220, 283–293, 301–304, 312, 314, 315, 319, 321–323, 326, 327, 329, 331, 332, 341–353, 537, 538	\GLSuseriii .....
\GLStext .....	491
487	\Glsuseriii .....
\Glstext .....	491
487	\glssuseriii .....
\glstext .....	491
487	\glssuseriiiaccessdisplay .....
\glstextaccessdisplay 369, 370, 373, 374, 379	383
\glstextformat .....	\GLSuseriv .....
117, 119, 466, 474, 475	491
\glstextup .....	\glssuseriv .....
40, 394, 426, 542	491
\glstildechar 50, 171, 172, 433, 503, 506, 507	\glssuserivaccessdisplay .....
\glstranslatefalse .....	384
27, 418, 419	\GLSuserv .....
\glstranslatetrue .....	492
27, 28, 419	\Glsuserv .....
\glstreechildpredesc .....	492
326, 327	\glssuserv .....
\glstreegroupheaderfmt .....	492
..... 306–311, 326–329, 333	\glssuservaccessdisplay .....
\glstreeindent .. 327, 329, 331, 332, 347–349	384
\glstreeitem .....	\glswrallowprimitivemodstrue .. 193, 521
306, 307, 325	\glswrite .....
\glstreenamebox .....	169–174, 182, 188, 189, 336–340, 504–508, 514, 518, 519
331, 332	\glswritedefhook .....
\glstreenamefmt .... 324, 326, 327, 329–332	79, 450, 455
\glstreenavigationfmt .....	\glswriteentry .....
..... 307–311, 326, 328, 329, 333	190, 520
\glstreepredesc .....	\glswritefiles .....
326, 327, 329	33, 188, 422, 518
\glstreesubitem .....	\glsxindyfalse .....
306, 325	29, 420
\glstreesubsubitem .....	\glsxindytrue .....
306, 325	30, 421
\glstype .....	
117, 118, 131–135, 149– 156, 377–379, 474, 475, 484–487, 493–497	<b>H</b>
\glsucmarkfalse .....	\H .....
11, 409	23, 416
\glsucmarktrue .....	\hangindent .....
11, 409	310, 311, 325, 327, 329, 331–333, 346–349
\glsunexpandedfieldvalue .....	\hbox .....
286	96, 285, 289, 342, 461
\glsunset .....	\hfill .....
.. 96, 99–101, 131–136, 461–464, 484–487	285, 289, 342
\glsunsetall .....	\hline .....
463	291–294, 301–305, 313–324
\glsupacrpluralsuffix 237, 243, 251, 255, 257, 260, 426, 548, 553, 557, 561, 563, 565	\hsize .....
	285, 289, 290, 300, 311, 312, 318
	\hspace .....
	325
	\hss .....
	285, 289, 342

<code>\ht</code> .....	299	186, 228, 274, 286, 325, 363, 440, 441,
<code>\hyperbf</code> .....	541	445, 449, 461, 474, 497, 499, 516, 517, 542
<code>\hyperdef</code> .....	38, 426	<code>\ifdefempty</code> .....
<code>\hyperemph</code> .....	542	19, 31, 47, 58, 62–
<code>\hyperit</code> .....	542	64, 68, 105, 109, 111, 181, 183, 210, 211,
<code>\hyperlink</code> .....	115, 129, 226, 474, 483, 541	232, 233, 240, 249, 253, 255, 258, 367,
<code>\hypermd</code> .....	541	371, 374, 389, 413, 422, 430, 431, 438,
<code>hyperref package</code> .....	196, 199, 224, 225, 271	440, 441, 443, 466, 469, 471, 514, 515,
<code>\hyperrm</code> .....	541	531, 532, 544, 545, 550, 556, 559, 561, 563
<code>\hypersc</code> .....	542	<code>\ifdefequal</code> .....
<code>\hypersf</code> .....	541	30,
<code>\hypersl</code> .....	542	31, 61–64, 74–76, 80, 90, 94, 211, 364,
<code>\hypertarget</code> .....	129, 483	421, 439–441, 447, 448, 450, 457, 460, 532
<code>\hypertt</code> .....	541	<code>\ifdefstrequal</code> .....
<code>\hyperup</code> .....	542	86, 455
		<code>\ifdefstring</code> .....
		10, 41, 65, 179–
		181, 183, 207, 208, 210, 212, 355, 408,
		427, 442, 512, 513, 515, 529, 530, 532, 533
		<code>\ifdefvoid</code> .
		22, 23, 93, 211, 415, 460, 532, 533
		<code>\ifdim</code> .....
		236, 299, 330, 547
		<code>\iffalse</code> .....
		93, 98, 459, 462
		<code>\IfFileExists</code> .....
		10,
		26, 27, 180, 181, 202, 408, 418, 419, 513, 526
		<code>\ifglossaryexists</code> .....
		45, 56, 61, 178–181, 429, 437–439, 511–513
<code>\IeC</code> .....	23, 415	<code>\ifgls@sanitize@description</code> ....
<code>\if</code> .....	121, 195, 335, 476, 477, 522	24, 416
<code>\if@endfor</code> .....	280	<code>\ifgls@sanitize@name</code> .....
<code>\if@gl@debug</code> .....	4, 20, 189, 405, 414, 519	24, 417
<code>\if@gl@docloaded</code> ...	4, 16, 34, 404, 411, 423	<code>\ifgls@sanitize@symbol</code> .....
<code>\if@gl@isacronymlist</code> .....	117, 413, 474	24, 417
<code>\if@openright</code> .....	48, 431	<code>\ifgls@xindy@gl@numbers</code> .....
<code>\ifbool</code> .....	17, 28,	57, 437
	33, 60, 106, 107, 412, 420, 423, 438, 467, 468	<code>\ifglsacrdescription</code> .....
<code>\ifboolexpr</code> .....	41, 65, 222, 427, 442, 539	263, 567
<code>\ifcase</code> .....	4, 7, 8, 27, 35, 71,	<code>\ifglsacrdua</code> .....
	214, 326, 346, 405–407, 419, 423, 445, 534	....
<code>\ifcsdef</code> .....	26, 42, 48, 74, 75,	251, 258, 261, 263, 557, 563, 565, 567
	82–87, 114, 190, 204, 205, 207, 209, 223,	<code>\ifglsacrfootnote</code> .....
	234, 363, 418, 428, 431, 447, 452–455,	117, 263, 474, 567
	472, 473, 519, 527, 528, 530, 531, 540, 545	<code>\ifglsacronym</code> .....
<code>\ifcsemtyp</code> .....	62, 274, 440	17, 412
<code>\ifcsequal</code> .....	62, 440	<code>\ifglsacrshortcuts</code> .....
<code>\ifcsname</code> .....	86	37, 246, 425, 554
<code>\ifcsstrequal</code> .....	87, 455	<code>\ifglsacrsmalldcaps</code> .....
<code>\ifcsstring</code> .....	86, 454	250–252, 254,
<code>\ifcsundef</code> .....	7, 14, 30, 38, 41, 44, 46,	255, 257, 260, 261, 557, 558, 561, 563, 565
	48, 59, 66, 68, 70, 89, 91, 92, 100, 115,	<code>\ifglsacrsmaller</code> .....
	119, 120, 129, 179, 180, 188, 199, 203,	250–252, 254, 255, 257, 557, 558, 561, 563
	205, 213, 217, 222–225, 227, 233, 281,	<code>\ifglsautomake</code> .....
	340, 364–367, 407, 410, 420, 426, 427,	32, 183, 422, 515
	429–431, 438, 442–444, 456, 458, 459,	<code>\ifglsdescsuppressed</code> .....
	463, 473, 475, 476, 483, 512, 513, 519,	282, 440
	524, 526, 528, 534, 537, 539, 540, 542, 545	<code>\ifglsentrycounter</code> .....
<code>\ifdef</code> .....	63,	....
	64, 72, 78, 79, 96, 115, 156, 160, 185,	11, 12, 37, 216, 217, 409, 425, 536
		<code>\ifglsentryexists</code> .....
		. 60, 77, 79, 88, 90, 438, 439, 449, 455, 457
		<code>\ifglsesclocations</code> .....
		192, 521
		<code>\ifglsfieldcseq</code> .....
		455
		<code>\ifglsfielddefeq</code> .....
		454
		<code>\ifglsfieldeq</code> .....
		454
		<code>\ifglschilden</code> .....
		282, 341, 439
		<code>\ifglsdesc</code> .....
		282, 440
		<code>\ifglsfield</code> .....
		440

## I



<code>\ifglshaslong</code> .....	102–104, 157, 235, 236, 240, 242, 256, 386, 387, 389, 392, 440, 465, 466, 498, 546, 547, 550, 552, 561	<code>\ifstrequal</code> .....	222, 539
<code>\ifglshasparent</code> .	205, 211, 330, 439, 528, 532	<code>\ifthenelse</code> ....	25, 38, 48, 119, 182, 222, 263, 280, 418, 426, 431, 475, 514, 539, 567
<code>\ifglshasprefix</code> .....	275–277	<code>\IfTrackedLanguage</code> .....	175, 508
<code>\ifglshasprefixfirst</code> .....	275, 276, 278	<code>\IfTrackedLanguageFileExists</code> .....	41, 403, 404, 427
<code>\ifglshasprefixfirstplural</code> .	275, 277, 278	<code>\iftrue</code> .....	93, 98, 459, 462
<code>\ifglshasprefixplural</code> .....	275, 277, 278	<code>\ifundef</code> .....	11, 12, 66, 77, 89, 169, 173, 182, 409, 442, 449, 456, 504, 507, 514
<code>\ifglshasshort</code> .....	440	<code>\ifvmode</code> .....	166, 503
<code>\ifglshassymbol</code> .....	249, 253, 258, 326, 327, 329, 331, 332, 440, 556, 559, 563	<code>\ifvoid</code> .....	300
<code>\ifglshyperfirst</code> .....	117, 474	<code>\ifx</code> .....	11, 13, 15, 18, 31, 36, 38, 49, 51, 53, 54, 57, 58, 90– 92, 95, 119, 120, 123–128, 157, 158, 169, 172, 174, 175, 177, 188, 192, 194–198, 201, 222, 223, 225, 226, 247, 248, 250, 252, 255, 257, 258, 260, 262, 264, 265, 336–338, 340, 341, 346–349, 354, 364, 409–413, 424–426, 432–435, 437, 438, 457–461, 475, 476, 478–483, 498, 504, 506, 508–511, 519, 521–525, 539–541, 555, 557, 559, 560, 562, 563, 565, 566, 568
<code>\ifglsindeonlyfirst</code> .....	190, 520	<code>\immediate</code> .....	31, 32, 77, 79, 102, 179–181, 188, 189, 203, 422, 449, 450, 465, 512, 513, 519, 526, 527
<code>\ifglsnogroupskip</code> .....	284, 287, 290, 292, 293, 296, 297, 301, 303, 304, 312, 314, 316, 319, 321, 323, 326, 327, 329, 332	<code>\in@</code> .....	36, 425
<code>\ifglsnonumberlist</code> .....	215, 535	<code>\indexname</code> .....	36, 424
<code>\ifglsnopostdot</code> .....	11, 408	<code>\indexspace</code> .....	284, 287, 306–311, 326–329, 332, 333
<code>\ifglsnnumberline</code> .....	48, 431	<code>\input</code> .....	40, 104, 427, 466
<code>\ifglssanitizesort</code> .....	22, 25, 415, 417	<code>inputenc</code> package .....	210
<code>\ifglssavenumberlist</code> .....	74, 183, 198, 447, 515, 524	<code>\inputencodingname</code> .....	30, 420
<code>\ifglssavewrites</code> .	33, 178, 190, 422, 511, 519	<code>\InputIfFileExists</code> .....	78, 449
<code>\ifglssubentrycounter</code> .....	12, 37, 216, 217, 409, 425, 535, 536	<code>\istfile</code> .....	518
<code>\ifglstoc</code> .....	48, 431	<code>\istfilename</code> .....	42, 169, 173, 180–182, 336, 339, 428, 504, 507, 512, 514
<code>\ifglstranslate</code> .....	40, 418, 427	<code>\item</code> .....	284–289, 306, 307, 325–327, 341, 342, 346
<code>\ifglsucmark</code> .....	46, 47, 430		
<code>\ifglssused</code> .....	102, 105–111, 117, 166, 190, 249, 253, 255, 258, 275–278, 368–374, 438, 465–471, 474, 503, 520, 556, 559, 561, 563		
<code>\ifglswrallowprimitivemods</code> ....	194, 521		
<code>\ifglsxindy</code> .....	42– 44, 49–52, 54–57, 67, 94, 95, 122, 167– 169, 175, 179, 180, 195, 197, 203, 334– 336, 420, 428, 429, 432–437, 443, 460, 477, 503, 504, 508, 512, 513, 522, 523, 526		
<code>\ifignoredglossary</code> .....	89, 93, 190, 199, 443, 457, 459, 520, 524		
<code>\ifin@</code> .....	36, 425		
<code>\ifinlistcs</code> .....	208, 213, 531, 534		
<code>\ifinner</code> .....	5, 406		
<code>\ifKV@glslink@hyper</code> .....	118, 119, 475		
<code>\ifKV@glslink@local</code> ....	131–136, 484–487		
<code>\ifmeasuring@</code> .....	96, 461		
<code>\ifmmode</code> .....	5, 406		
<code>\ifnum</code> .	14, 32, 100, 101, 181, 210, 299, 327, 329, 331, 347, 348, 410, 422, 464, 514, 532		
<code>\ifstrempty</code> .....	6, 341, 406		

  

**J**

<code>\jobname</code> .....	31, 43, 77–79, 169, 173, 179–181, 202, 336, 339, 422, 428, 449, 504, 507, 512–514, 526
-----------------------------	----------------------------------------------------------------------------------------------

  

**K**

<code>\key@ifundefined</code> .....	81, 82, 451
<code>\KV@glslink@hyperfalse</code> .....	115, 117, 118, 129, 474, 475, 483
<code>\KV@glslink@hypertrue</code> ..	115, 130, 474, 484

L	
<code>\L</code> .....	24, 416
<code>\l</code> .....	24, 416
<code>\label</code> .....	8, 214, 216, 407, 534, 536
<code>\language</code> .....	29, 420
<code>\leaders</code> .....	285, 289, 342
<code>\leavevmode</code> .....	87, 118, 455, 475
<code>\let</code> ..	4, 5, 10, 13–17, 23, 24, 26, 27, 31–33, 35–38, 41, 42, 51, 52, 63–65, 74, 76, 77, 87, 88, 90, 92, 93, 96, 98, 99, 101, 104, 116–119, 121, 129–136, 149–155, 157, 158, 164, 165, 173, 174, 178–187, 190, 191, 194, 197–202, 211, 213, 216, 220, 232, 245–247, 250, 252–262, 272, 280, 281, 286, 299, 306, 307, 325, 340, 358, 359, 377–379, 396–400, 403–405, 408, 410–412, 415, 416, 418, 419, 421–428, 433, 434, 441, 447, 448, 455–459, 461– 463, 465, 466, 474, 475, 477, 483–487, 493–498, 502, 507, 508, 511–513, 515– 521, 524–526, 532–535, 538, 544, 554–566
<code>\letcs</code> .....	31, 61–64, 79, 81, 82, 85, 91, 92, 156, 157, 179, 180, 185, 186, 204, 206, 210, 211, 218, 222, 330, 439–441, 449, 451, 454, 458, 497, 498, 512, 513, 516, 517, 527–529, 532, 537, 539
link text .....	<u>105</u>
<code>\listcsadd</code> .....	208, 531
<code>\listcsgadd</code> .....	213, 534
<code>\listcsxadd</code> .....	204, 527, 528
<code>\listead</code> .....	208, 530
<code>\listxadd</code> .....	77
<code>\loadglsentries</code> .....	104, 466
<code>\long</code> .....	87, 226, 455, 541
<code>\longnewglossaryentry</code> .....	88, 455, 456
<code>\longprovideglossaryentry</code> .....	455, 456
longtable package .....	<u>289</u> , <u>296</u> , <u>300</u>
<code>\LT@end@pen</code> .....	299
<code>\LT@err</code> .....	299
<code>\LT@foot</code> .....	299, 300
<code>\LT@head</code> .....	300
<code>\LT@lastfoot</code> .....	299, 300
<code>\LT@output</code> .....	299
M	
<code>\makeatletter</code> .....	78, 202, 449, 526
<code>\makebox</code> ....	285, 289, 330–332, 342, 348, 349
makeglossaries	29, 43, 56, 57, 66, 175, 182, 203
<code>\makeglossaries</code> .....	6, 7, 30–35, 39, 71, 178, 183, 185, 187, 203, 406, 407, 421–424, 426, 445, 511, 514–516, 518, 527
<code>\makeglossary</code> .....	33, 35, 423, 424
makeindex .....	9, 13, 29, 32, 39, 43, 45, 49, 65–67, 69, 95, 120, 124, 167, 170, 173– 175, 178, 189, 193–196, 199, 221, 335, 336
delim_n .....	45
delim_r .....	45
page_compositor .....	43
special characters .....	122, 123, 167
<code>\makenoidxglossaries</code> ...	6, 7, 30, 31, 71, 183, 187, 406, 407, 421, 445, 515, 516, 518
<code>\MakeTextUppercase</code> .....	3, 404
<code>\MakeUppercase</code> .....	369–371, 378, 379
<code>\marginpar</code> .....	5, 406
<code>\markboth</code> .....	46, 430
<code>\mbox</code> ....	166, 284, 288, 310, 311, 330, 342, 503
memoir class .....	<u>189</u>
<code>\memUHead</code> .....	46, 430
<code>\MessageBreak</code> .....	20, 34, 65, 201, 354, 403, 404, 414, 423, 442, 525
mfirstuc package .....	<u>1</u> , <u>157</u>
<code>\mfirstucMakeUppercase</code> .....	3, 46, 47, 83, 106, 108–113, 137–148, 150, 152, 154, 156, 233, 240, 241, 243, 253, 259, 277, 278, 372–376, 379–384, 390, 391, 393, 404, 430, 453, 467, 469–472, 487– 493, 495–497, 544, 545, 550–552, 559, 564
<code>\midrule</code> .....	296, 297
<code>\month</code> .....	169, 173, 336, 339, 504, 507
multicol package .....	<u>306</u>
N	
<code>\n</code> .....	173, 174, 339, 507, 508
<code>\NeedsTeXFormat</code> .....	..... 3, 272, 285, 334, 340, 354, 403, 404
<code>\new@glossaryentry</code> ..	77, 185, 448, 449, 516
<code>\new@ifnextchar</code> ....	65, 82, 83, 102, 103, 130–134, 136–155, 228–231, 274–278, 442, 452, 453, 465, 466, 484–497, 542, 543
<code>\newacronym</code> .....	227, 232, 248, 250, 252, 255, 257, 260, 262, 264, 542, 544, 555, 557, 558, 560, 562, 565, 566, 568
<code>\newacronymhook</code> .....	232, 248, 250, 252, 255, 257, 260, 262, 264, 395, 544, 555, 557, 559, 561, 562, 565, 566, 568
<code>\newacronymstyle</code>	235–240, 242–244, 545–554
<code>\newcommand</code> .....	... 5–23, 25, 26, 28, 29, 31–40, 42–52,

54–62, 64–68, 70–88, 93–105, 109, 111,  
 113, 114, 116–119, 121, 122, 129–168,  
 174, 175, 177–181, 183, 184, 187–192,  
 194–200, 202, 204–210, 212–224, 226–  
 234, 236, 244, 246–270, 273–281, 283,  
 284, 286, 299, 306, 324, 325, 330, 340,  
 356, 360–364, 386, 395, 400–402, 406–  
 415, 417, 418, 420, 422–456, 459–463,  
 465, 466, 469, 471–475, 477, 483–504,  
 508, 511–515, 518–533, 535–547, 554–570  
 \newcount ..... 14, 15, 74, 410, 411, 447  
 \newcounter ..... 11, 12, 409  
 \newenvironment ..... 217, 537  
 \newglossary ..... 16, 17, 35,  
 36, 67, 182, 411, 412, 424, 442, 443, 514, 515  
 \newglossaryentry .. 6, 36, 73, 74, 77, 99,  
 232, 246, 249, 251, 253, 256, 259, 261,  
 264, 396–399, 406, 424, 446, 448, 463,  
 544, 554, 556, 558, 560, 561, 564, 565, 568  
 \newglossaryentry options  
   access ..... 358, 360  
   counter ..... 70  
   description .....  
     . 28, 69, 74, 76, 88, 140, 158, 228, 257, 357  
   descriptionaccess ..... 361, 365  
   descriptionplural ..... 141, 357  
   descriptionpluralaccess ..... 361, 365  
   entrycounter ..... 214  
   first .... 70, 91, 130, 137, 159, 254, 260, 356  
   firstaccess ..... 361, 365  
   firstplural ..... 70, 138, 160, 356  
   firstpluralaccess ..... 361, 365  
   format ..... 169  
   long ..... 111, 163, 357  
   longaccess ..... 362, 366  
   longplural ..... 163, 357  
   longpluralaccess ..... 362, 366  
   name .. 69, 73, 76, 88, 139, 157, 198, 356, 358  
   nonumberlist ..... 71, 72  
   parent ..... 71, 76  
   plural ..... 69, 91, 138, 356  
   pluralaccess ..... 361, 365  
   prefix ..... 272  
   prefixfirst ..... 272  
   prefixfirstplural ..... 273  
   prefixplural ..... 273  
   see ..... 6, 9, 71, 76, 182, 184  
   short ..... 111, 162, 357, 363  
   shortaccess ..... 361, 366, 395  
   shortpl ..... 363  
   shortplural ..... 162, 357  
   shortpluralaccess ..... 361, 366  
   sort ..... 69, 161, 189, 221  
   symbol ..... 69, 70, 142,  
     250–252, 254, 260, 293, 315, 356, 358, 363  
   symbolaccess ..... 361, 365  
   symbolplural ..... 143, 356  
   symbolpluralaccess ..... 361, 365  
   text .... 69, 70, 130, 136, 158, 250, 254, 356  
   textaccess ..... 360, 364  
   type ..... 16, 70, 104, 160  
   user1 ..... 143, 161, 357  
   user1access ..... 362, 366  
   user2 ..... 144, 161, 357  
   user2access ..... 362, 366  
   user3 ..... 145, 161, 357  
   user3access ..... 362, 366  
   user4 ..... 146, 162, 358  
   user4access ..... 362, 367  
   user5 ..... 147, 162, 358  
   user5access ..... 362, 367  
   user6 ..... 147, 162, 358  
   user6access ..... 362, 367  
 \newglossarystyle ..... 281,  
 284–299, 301–323, 325–330, 332, 333, 540  
 \newif ..... 3, 4, 18, 26,  
 29, 32, 193, 404, 405, 413, 418, 420, 422, 521  
 \newignoredglossary ..... 443  
 \newlength ..... 129,  
 285, 289, 290, 300, 311, 312, 318, 328, 483  
 \newrobustcmd ..... 6, 61–63, 76, 77, 83,  
 102, 103, 117, 130–155, 157–163, 165,  
 166, 197, 198, 228–231, 273–278, 330,  
 364–367, 406, 439, 440, 448, 449, 452,  
 453, 465, 466, 474, 484–503, 524, 542, 543  
 \newterm ..... 36, 424  
 \newtoks .... 122, 178, 231, 232, 478, 511, 544  
 \newwrite ..... 31, 77, 169,  
 173, 178, 182, 422, 449, 504, 507, 511, 514  
 \nfss@text ..... 5, 406  
 ngerman package ..... 174  
 \noalign ..... 299  
 \nobreak ..... 284, 288, 300, 342  
 \noexpand ..... 18,  
 38, 50, 51, 78, 92, 93, 114, 119–122, 128,  
 164, 175–177, 179–181, 183, 191, 192,  
 195, 199, 203, 206, 218, 220, 227, 232,  
 246, 247, 249, 251, 253, 254, 256, 259,

261, 263, 264, 334, 354, 355, 396–400, 413, 426, 432, 433, 449, 459, 460, 472, 473, 475–477, 482, 483, 502, 509–513, 515, 520–522, 524, 526–529, 537, 538, 542, 544, 554–556, 558, 560–562, 564–568	\org@glossarytitle ..... 201, 525
\nohyperpage ..... 225, 540	\org@glspostdescription ..... 42, 428
\noindent ..... 220, 307–311, 328, 329, 538	\org@ifKV@glslink@hyper .... 118, 119, 475
\noist ..... 339, 340, 511	\outputpenalty ..... 299
\nopostdesc ..... ... 36, 42, 87, 201, 341, 424, 428, 455, 526	<b>P</b>
\normalbaselineskip ..... 299	\p@ ..... 284, 285, 306, 324, 325
\ns@ACRfull ..... 230, 543	\p@glshyp@opt ..... 116, 474
\ns@Acrfull ..... 229, 542	package options:
\ns@acrfull ..... 228, 542	acronym ..... 16, 17, 39, 200, 228
\ns@ACRfullpl ..... 231, 543	true ..... 17
\ns@Acrfullpl ..... 230, 543	counter ..... 19
\ns@acrfullpl ..... 230, 543	debug
\ns@ACRlong ..... 153, 495, 496	showaccsupp ..... 6
\ns@Acrlong ..... 152, 153, 495	showtargets ..... 5, 6
\ns@acrlong ..... 152, 495	description ..... 254, 255
\ns@ACRlongpl ..... 155, 497	disablemakegloss ..... 31
\ns@Acrlongpl ..... 154, 155, 496	dua ..... 253–255
\ns@acrlongpl ..... 154, 496	entrycounter ..... 11, 214, 216, 286
\ns@ACRshort ..... 149, 150, 493	true ..... 12
\ns@Acrshort ..... 149, 493	esclocations
\ns@acrshort ..... 148, 492	false ..... 9
\ns@ACRshortpl ..... 151, 494	footnote ..... 131–135, 251, 253, 254, 257
\ns@Acrshortpl ..... 151, 494	hyperfirst
\ns@acrshortpl ..... 150, 494	false ..... 131–135
\ns@newglossary ..... 66, 442	index ..... 36
\null ..... 121– 128, 175–177, 202, 477–483, 509–511, 526	kernelglossredefs
\number ..... 14, 78, 91, 101, 191, 194, 220, 355, 410, 449, 458, 465, 520, 521, 538	nowarn ..... 34
\numberline ..... 48, 431	makeindex ..... 171, 271
\numexpr ..... 101, 465	nogroupskip .... 290, 292, 293, 296, 297, 301, 303, 304, 312, 314, 316, 319, 321, 323
<b>O</b>	nolist ..... 265
\O ..... 24, 416	nolong ..... 265, 289
\o ..... 24, 416	nomain ..... 16
\OE ..... 23, 416	nonumberlist ..... 9
\oe ..... 24, 416	nosuper ..... 265
\oldacronym ..... 542	notree ..... 265
\openout ..... 31, 77, 169, 173, 179, 336, 339, 422, 449, 504, 507, 512	nowarn ..... 4
\OR ..... 263, 567	numberline ..... 7
\or ..... 4, 5, 7, 8, 27, 35, 214, 326, 346, 405, 407, 419, 424, 534	record ..... 274
\org@glossaryentrynumbers ..... 201, 216, 526, 535	sanitize ..... 24, 69, 157, 158
	sanitizesort ..... 21
	savewrites ..... 33
	false ..... 178
	true ..... 182, 188
	section ..... 7, 47
	sort
	def ..... 13, 14
	none ..... 13

standard	13	toctitle	213
use	13, 14	type	16, 198, 213
style	8, 265	\printindex	36, 424
subentrycounter	12, 214, 216	\printnoidxglossaries	185, 516, 525
toc	7	\printnoidxglossary	184, 185, 187, 200, 207, 208, 215, 516, 518, 525, 529, 530, 535
true	7	\printnoidxglossary options	
translate	27	sort	215
false	26	\printnumbers	36, 424
translator	26	\printsymbols	35, 424
xindy	29, 30, 171, 271	\ProcessOptions	272, 354
\PackageError	6, 7, 16, 32, 39, 50, 56, 60, 61, 65, 70, 73, 74, 81– 87, 89–91, 99, 115, 156, 177, 178, 182, 185, 187, 207–209, 213, 215, 223, 224, 233, 234, 251, 252, 258, 261, 340, 367, 406, 407, 411, 422, 426, 432, 437–439, 442, 444, 446, 447, 451–457, 463, 473, 497, 511, 515, 516, 518, 529–531, 534, 535, 539, 540, 545, 546, 557, 558, 563, 565	\ProcessOptionsX	37, 425
\PackageInfo	4, 5, 31, 179, 189, 405, 421, 512, 519	\protect	48, 113, 235, 236, 242, 249, 253, 256, 376, 377, 386, 387, 392, 393, 431, 472, 546, 547, 552, 556, 559, 561
\PackageWarning	4, 5, 20, 405, 414	\protected@csedef	84, 453
\PackageWarningNoLine	4, 5, 20, 403–405, 414	\protected@csxdef	84, 453
\pagegoal	299	\protected@eappto	31, 68, 94, 191
\pagelistname	42, 292–294, 297, 303–305, 314–318, 321–324, 426, 428	\protected@edef	6, 8, 15, 18, 31, 51, 53, 55, 57, 58, 61, 66, 68, 84–90, 93–95, 106, 107, 113, 118, 120, 164, 166, 184, 190, 192, 195, 199, 204, 205, 214, 216, 218, 220, 223, 232, 258, 264, 273, 279, 335, 336, 354, 355, 363, 364, 395, 406, 407, 433, 434, 436, 438, 457, 460, 467, 468, 472, 476, 502, 519, 521, 522, 534, 537–540, 544, 563, 568
\par	42, 220, 284, 285, 288, 306, 308–311, 324, 325, 327–333, 342, 347–349, 428, 538	\protected@write	65, 66, 169– 171, 182, 184, 187, 190, 199, 280, 336, 441, 442, 504, 505, 514, 516, 518, 520, 524
\parindent	306–311, 325, 327–329, 331–333, 347–349	\protected@xdef	13– 15, 18, 23, 75, 90, 93, 95, 96, 195, 202, 359, 360, 410–412, 415, 447, 460, 461, 522
\parskip	306–309, 325, 327, 328	\providecommand	17, 30, 39, 40, 47, 65, 102, 130, 171, 182, 185, 188, 203, 218, 219, 274, 284, 285, 306, 324, 355, 412, 421, 426, 430, 441, 465, 484, 505, 514, 516, 518, 526, 527, 537, 538
\PassOptionsToPackage	272, 354	\provideglossaryentry	448, 449
\penalty	299	\ProvidesFile	40, 427
\phantomsection	47, 431	\ProvidesGlossariesLang	427
polyglossia package	26, 40	\ProvidesPackage	3, 272, 279, 281, 284, 285, 289, 296, 300, 306, 311, 318, 324, 334, 340, 354, 403, 404
\printacronyms	412		
\PrintChanges	404		
\printglossaries	183, 515, 525		
\printglossary	17, 20, 35, 36, 183, 200, 215, 412, 414, 424, 515, 524, 525, 535		
\printglossary options			
entrycounter	214		
nogroupskip	214		
nonumberlist	215		
nopostdot	214		
numberedsection	214		
style	213		
subentrycounter	214		
title	213		

## R

\r	23, 416
\raggedright	298, 299, 301–305, 319–324
\raisebox	129, 483
\ref	217, 536

\refstepcounter .....	216, 536	\setabbreviationstyle .....	262
\relax .....	4, 8, 10, 14–17, 27, 31, 32, 35–37, 52, 65, 70, 71, 75, 78, 90, 92, 96, 100, 101, 116, 117, 119, 121, 123–128, 157, 158, 171–175, 177, 178, 181, 182, 184–186, 191, 195, 197, 198, 201, 202, 210, 211, 213, 214, 222, 223, 265, 280, 284–286, 299, 306, 310, 311, 324, 326, 327, 329–333, 335, 338–340, 346–349, 358, 359, 377–379, 396–399, 405, 407, 408, 410–412, 418, 419, 422–425, 434, 441, 444, 445, 448, 449, 457, 459, 461, 464, 465, 474, 476–482, 498, 506–511, 514–517, 520, 522–526, 532, 534, 539, 540, 568	\SetAcronymLists .....	413
\renewacronymstyle .....	386–389, 392, 394, 395, 545	\SetAcronymStyle .....	28, 29, 420, 567
\renewcommand .....	3–5, 7–10, 12, 13, 16, 17, 19–21, 25, 27–30, 32, 33, 35, 37, 41–44, 57, 68, 71, 72, 87, 99–101, 164, 166, 168, 174, 176, 181–186, 202, 203, 214, 232, 233, 235–239, 241–244, 248, 250–252, 255, 257, 260, 262, 264, 282–294, 296, 297, 299, 301–316, 319–323, 325–335, 340–348, 350–353, 355, 358, 359, 367, 371, 374, 376, 377, 385–389, 391–395, 397–399, 404–414, 417, 419–422, 424, 425, 427–429, 437, 443, 445, 455, 463–465, 502–504, 508, 509, 514–517, 526, 527, 534, 544–555, 557, 558, 560–563, 565, 566, 568	\setacronymstyle .....	262, 545
\renewenvironment .....	218, 282, 284, 286, 290–295, 298, 299, 301–325, 327, 328, 330, 537	\setbool .....	25, 417
\renewglossarystyle .....	540	\setbox .....	299, 300
\RequireGlossariesLang .....	41, 403, 404, 426, 427	\setcounter .....	216, 535, 536
\RequirePackage .....	3, 9, 10, 26, 27, 37, 40, 265, 271, 272, 289, 296, 300, 306, 311, 318, 355, 403, 404, 408, 418, 419, 425, 427, 568, 570	\SetCustomDisplayStyle .....	264, 567, 568
\restorecounters@ .....	120, 476	\SetCustomStyle .....	568
\romannumeral .....	191, 194, 330–332, 348, 520, 521	\SetDefaultAcronymDisplayStyle .....	248, 554, 555
<b>S</b>		\SetDefaultAcronymStyle .....	263, 555, 567
\s@glshyp@opt .....	116, 474	\SetDescriptionAcronymDisplayStyle .....	255, 559–561
\s@GlsSetXdyFirstLetterAfterDigits .....	167, 168, 503	\SetDescriptionAcronymStyle .....	263, 560, 567
\s@GlsSetXdyNumberGroupOrder .....	168, 503, 504	\SetDescriptionDUAAcronymDisplayStyle .....	252, 253, 557, 559
\s@ifglossaryexists .....	59, 199, 438, 525	\SetDescriptionDUAAcronymStyle .....	263, 558, 567
\s@newglossary .....	66, 442	\SetDescriptionFootnoteAcronymDisplayStyle .....	250, 251, 556, 557
\savecounters@ .....	119, 476	\SetDescriptionFootnoteAcronymStyle .....	263, 557, 567
\seenname .....	197, 426, 524	\SetDUADisplayStyle .....	262, 565–567
		\SetDUASyle .....	263, 566, 567
		\setentrycounter .....	51, 171, 212, 334, 433, 506, 533, 539
		\SetFootnoteAcronymDisplayStyle .....	257, 561, 562
		\SetFootnoteAcronymStyle .....	263, 562, 567
		\SetGenericNewAcronym .....	234, 544, 545
		\setglossarypreamble .....	429
		\setglossarysection .....	430
		\setglossarystyle .....	201, 223, 265, 284, 285, 287–292, 294–299, 301–324, 326, 328, 329, 332, 333, 525, 539, 540, 568
		\setglossentrycompatibility .....	213, 223, 534, 538, 540
		\setkeys .....	26, 30, 37, 47, 89, 118, 166, 201, 232, 248, 250, 252, 255, 257, 260, 262, 264, 418, 421, 425, 430, 456, 475, 502, 503, 525, 544, 555, 557, 559, 560, 562, 565, 566, 568
		\setlength .....	285, 289, 290, 300, 306–309, 311, 312, 318, 325, 327, 328, 332, 348, 349
		\SetSmallAcronymDisplayStyle .....	260, 563, 565
		\SetSmallAcronymStyle .....	263, 565, 567
		\setStyleFile .....	428
		\settoheight .....	129, 483

\settextwidth .....	236, 330–332, 348, 547	435–438, 445, 446, 463, 465, 466, 472–	
\setupglossaries .....	425	474, 501, 504–507, 511–516, 518, 524,	
\sfcode .....	11, 408	525, 527, 536, 540, 542, 546–554, 559, 563	
\show .....	265–270, 401, 402, 568–570	\spacefactor .....	11, 408
\showacronymlists .....	570	\SS .....	24, 416
\showgloccounter .....	569	\ss .....	24, 416
\showglodesc .....	569	\string .....	6, 7, 16, 20, 30–
\showglodescplural .....	569		35, 39, 49, 50, 53–55, 57, 65, 66, 71, 73,
\showglofield .....	570		74, 79, 82, 83, 94, 95, 99, 100, 102–104,
\showglofirst .....	569		114, 116, 120, 121, 123–125, 127, 128,
\showglofirstpl .....	569		164, 167–175, 177, 178, 182–185, 187,
\showgloflag .....	570		188, 195–197, 201, 203, 207, 208, 215,
\showgloindex .....	570		220, 223, 262, 280, 334–340, 406, 407,
\showglolevel .....	569		411, 414, 421–423, 426, 431–438, 441,
\showgloloclist .....	570		442, 445, 446, 449, 450, 452, 460, 461,
\showglolong .....	570		463, 465, 466, 472–474, 476–480, 482,
\showglongname .....	569		501, 503–508, 510, 511, 514–516, 518,
\showglopent .....	568		522, 523, 525–527, 529, 530, 535, 538, 540
\showglopplural .....	569	\strut .....	220, 284, 285, 287–290, 292,
\showgloshort .....	570		293, 301, 303, 304, 312, 314, 315, 319,
\showglosort .....	569		321, 323, 329, 341–345, 347, 350–353, 538
\showglossaries .....	570	\subglossentry .....	94, 202,
\showglossarycounter .....	570		211, 220, 283–285, 287–290, 292, 293,
\showglossaryentries .....	570		301, 302, 304, 312, 314, 315, 319, 321,
\showglossaryin .....	570		322, 326, 327, 329, 331, 460, 526, 532, 538
\showglossaryout .....	570	\subitem .....	306, 325, 326, 346
\showglossarytitle .....	570	\subsubitem .....	306, 325, 326, 346
\showglosymbol .....	570	supertabular package .....	<u>10</u> , <u>265</u> , <u>311</u> , <u>318</u>
\showglosymbolplural .....	570	\symbolname .....	42,
\showglotext .....	569		294, 297, 305, 316–318, 323, 324, 426, 428
\showglotype .....	569		
\showglouserl .....	569		T
\showglouserll .....	569	\t .....	23, 416
\showglouserlll .....	569	\thead .....	312–324
\showglouserllll .....	569	\tbody .....	312–324
\showglouserlv .....	569	\tblarnewline .....	290–294, 296,
\showglouserv .....	569		297, 301–305, 312–324, 344, 345, 350, 351
\showglouservi .....	569	\texorpdfstring .....	160, 499
\SmallNewAcronymDef .....	260, 564, 565	\textbar .....	281
\space .....	6, 7,	\textbf .....	220, 226, 324, 346–349, 538, 541
	30–35, 39, 50, 54, 55, 57, 71, 73, 74, 99,	textcase package .....	<u>3</u>
	100, 102–104, 113, 114, 116, 164, 169–	\textit .....	226, 542
	173, 177, 179–183, 185, 187, 198, 201,	\textmd .....	226, 541
	203, 217, 223, 229, 235, 236, 238, 239,	\textrm .....	226, 541
	241–244, 253, 258, 281, 283, 284, 287,	\textsc .....	227, 237, 243, 251, 255, 257,
	288, 290, 301, 312, 319, 325–327, 329,		260, 394, 542, 548, 553, 557, 561, 563, 565
	331, 332, 334, 335, 337–339, 341–344,	\textsf .....	226, 541
	346–350, 352, 376, 377, 386–389, 391,	\textsl .....	227, 542
	393–396, 406, 407, 421–423, 426, 432,	\textsmaller .....	237, 243, 251, 255,
			257, 260, 394, 548, 553, 557, 561, 563, 565



\texttt	226, 541	\uccode	210, 532
\textulc	228, 542	\undef	72, 78, 199, 445, 449, 524
\textup	227, 228, 542	\unskip	87, 285, 289, 342, 455
\TH	24, 416	\unvbox	299, 300
\th	24, 416	\usedictionary	41, 427
\the	38, 41, 50, 55, 57, 66, 123–128, 169, 173, 175, 177, 188, 190, 194, 199, 210, 218, 220, 226, 232, 233, 235, 236, 241, 242, 246, 247, 249, 251–254, 256, 259, 261, 263, 264, 334, 336, 339, 355, 386, 387, 391, 392, 395–400, 426, 427, 433, 436, 438, 442, 478–483, 504, 507, 509–511, 519, 521, 524, 532, 537, 538, 541, 544–547, 551, 552, 554–556, 558, 560–562, 564–568	\usepackage	207, 208, 529, 530
\the@numberlist	164, 165, 502	\UTFviii@two@octets	24, 416
\theglossary	4, 404	\UTFviii@two@octets@combine	24, 416
\theglossaryentry	11, 217, 409, 536		
\theglossarysubentry	12, 217, 409, 536	V	
\theglentrycounter	119, 120, 192, 195, 335, 336, 475, 476, 521, 522	\v	23, 416
\theH	196, 523	\vbox	299, 300
\theHglossaryentry	11, 409	\vsize	300
\theHglossarysubentry	12, 409	\vskip	284, 285, 299, 306, 324
\theHglentrycounter	..... 120, 192, 195, 476, 521, 522	\vss	299, 300
\thesection	38, 426		
\this@dialect	41, 403, 404, 427	W	
\tiny	5, 406	\warn@nomakeglossaries	..... 182, 184, 185, 514–516
\toks@	38, 41, 50, 55, 57, 66, 123– 128, 175, 177, 198, 199, 218–220, 226, 334, 354, 355, 426, 427, 433, 436, 438, 442, 478–483, 509–511, 524, 537, 538, 541	\warn@noprintglossary	..... 183–185, 202, 515, 516, 524, 526
\toprule	296, 297	\write	32, 79, 102, 169–174, 179–181, 185, 188, 203, 336–340, 422, 449, 450, 465, 504–508, 512, 513, 516, 519, 526, 527
tracklang package	40, 403	\writeist	178, 340, 504, 507, 508, 511
\trans@languages	41, 427		
\translate	41, 42, 427, 428	X	
\translatelet	16, 17, 35, 36, 411, 412, 424	\x	541
translator package	16, 17, 26, 35, 36, 40, 41, 199	\xatlevel@	119, 475
\triangleright	5, 406	\xcapitalisewords	499
\ttfamily	6, 406	\xdef	91, 280, 457–459, 526
\TX@trial	96, 461	\xglssaccsupp	364
\typeout	20, 414	\xifinlist	77, 78
		\xifinlistcs	204, 205, 208, 527, 528, 530
		xindy	9, 13, 23, 29, 30, 32, 43, 44, 49, 52, 54, 56, 57, 95, 127, 128, 167– 170, 189, 193, 195, 199, 203, 221, 271, 335
		\xmakefirstuc	106–108, 113, 156, 158, 273, 467, 468, 472, 497, 498
		\xspace	227, 542
		xspace package	3, 227
		Y	
		\year	169, 173, 336, 339, 504, 507
		Z	
		\z@	299, 300