

Documented Code For glossaries v4.48

Nicola L.C. Talbot

Dickimaw Books

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

2021-10-19

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 v4.48: L^AT_EX2_ε Package to Assist Generating Glossaries”.

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

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

INSTALL Installation instructions.

CHANGES Change log.

README.md Package summary.

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

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

Contents

1	Main Package Code	4
1.1	Package Definition	4
1.2	Package Options	5
1.3	Predefined Text	40
1.4	Xindy	50
1.5	Loops and conditionals	59
1.6	Defining new glossaries	65
1.7	Defining new entries	70
1.8	Resetting and unsetting entry flags	97
1.9	Keeping Track of How Many Times an Entry Has Been Unset	100
1.10	Loading files containing glossary entries	105
1.11	Using glossary entries in the text	105
1.12	Adding an entry to the glossary without generating text	166
1.13	Creating associated files	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
2	Prefix Support (glossaries-prefix Code)	273
3	Glossary Styles	280
3.1	Glossary hyper-navigation definitions (glossary-hypernav package)	280
3.2	In-line Style (glossary-inline.sty)	282
3.3	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
4 Backwards Compatibility	335
4.1 glossaries-compatible-207	335
4.2 glossaries-compatible-307	341
5 Accessibility Support (glossaries-accsupp Code)	355
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
6 Multi-Lingual Support	404
6.1 Polyglossia Captions	404
6.2 Rollback v4.46 (glossaries-2020-03-19.sty)	405
6.3 Rollback v4.46 (glossary-list-2020-03-19.sty)	572
Glossary	574
Change History	575
Index	603

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{v4.48}{2021-10-19}
```

Declare package:

```
4 \ProvidesPackage{glossaries}[2021/10/19 v4.48 (NLCT)]
```

Required packages:

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

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

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

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

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

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

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

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

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

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

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

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

Check if doc has been loaded.

```
f@gls@docloaded
```

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

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

```
16 {%
```

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

```
18 }%
```

```
19 {%
```

```

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

```

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

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

```
org@theglossary
```

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

```
@endtheglossary
```

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

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

```

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

```

End of doc stuff.

```
33 \fi
```

1.2 Package Options

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

`showtargetouter`

```

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

```

`showtargetsymbol`

```

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

```

sshowtargetfont

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

\@glsshowtarget

debug=showtargets will redefine this.

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

@@glsshowtarget

Need to detokenize the label in the event that it contains awkward characters like 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 redefine \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   \ifglssacronym
381     \renewcommand{\@gls@do@acronymsdef}{%
382       \DeclareAcronymList{acronym}%
383       \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
384       \renewcommand*{\acronymtype}{acronym}%
385       \newcommand*{\@gls@tr@set@acronym@toctitle}{%
386         \translatelet{\glossarytoctitle}{Acronyms}%
387       }%
388     }%
389   \else
390     \let\@gls@do@acronymsdef\relax
391   \fi
392 }

```

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

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

```

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

```

@do@acronymsdef Set default value

```

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

```

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

```

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

```

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

```

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

```

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

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

`addtoacronymlists`

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

`declareAcronymList` Identifies the named glossary as a list of acronyms and adds to the list. (Doesn't check if the glossary exists, but checks if label already in list. Use `\SetAcronymStyle` after identifying all the acronym lists.)

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

`glsIfListOfAcronyms`

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

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

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

Internal command requires label and list to be expanded:

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

Not found

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

Found

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

`glsisacronymlist` Convenient boolean.

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

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

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

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

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

`acronymlists`

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

The default counter associated with the numbers in the glossary is stored in `\glscounter`. This is initialised to the page counter. This is used as the default counter when a new glossary is defined, unless a different counter is specified in the optional argument to `\newglossary` (see [section 1.6](#)).

`\glscounter`

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

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

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

`gls@nohyperlist`

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

`lareNoHyperList`

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

`nohypertypes`

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

`glossariesWarning` Prints a warning message.

```

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

```

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

```

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

```

`glossariesWarningNoLine` 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 }

```

`nowarn` Define package option to suppress warnings

```

479 \@gls@declareoption{nowarn}{%
480   \if@gls@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 }

```

`missinglang@warn` 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 }

```

`nolangwarn` Suppress warning if language support not found.

```

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

```

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

```

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

```

theglossdefined Issue a warning if overriding theglossary

```

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

noredefwarn Suppress warning on redefinition of \printglossary

```

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

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

ls@sanitizedesc

```

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

glsssetexpandfield `\glsssetexpandfield{<field>}`

Sets field to always expand.

```

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

glsssetnoexpandfield `\glsssetnoexpandfield{<field>}`

Sets field to never expand.

```

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

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

```

521 \glsssetexpandfield{type}
```

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

```

522 \glsssetnoexpandfield{desc}
```

descplural@field

```

523 \glsssetnoexpandfield{descplural}
```

ls@sanitizename

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

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

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

@sanitizesymbol

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

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

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

bolplural@field

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

Sanitizing stuff:

ls@sanitizesort

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

ls@sanitizesort

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

@nosanitizesort

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

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

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

@nosanitizesort

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

idxstripaccents

This strips accents by redefining the standard accent commands to just do their argument. (This will be localised since `\glsnoidxstripaccents` is used within a group.) Anything outside this standard set really shouldn't be using `\makenoidxglossaries`. It's much better to use `xindy` or `bib2gls` with the correct language setting.

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

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



```

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

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

```

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

```

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

```

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

```

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

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

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

```

```

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

```

sanitizesort

```

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

```

Default setting:

```

655 \glsssanitizesorttrue
656 \glsssetnoexpandfield{sortvalue}%

```

setsanitizesort Default behaviour for \makenoidxglossaries is sanitizesort=false.

```

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

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

```

sanitize

```

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

```

```

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

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

otranslatorhook \@gls@notranslatorhook has been removed.

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

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

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

```

```

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

```

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

```

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

```

Set the default value:

```

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

```

```

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

```

indexonlyfirst Set whether to only index on first use.

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

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

shotcuts   Define acronym shortcuts.
795 \define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{%

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

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

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

\ifglxindy   Provide boolean to determine whether 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@premakeg` 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*\glnumbersgroupname{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     \ifglxsindy
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.

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

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

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

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

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

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

\@xdyattributelist Comma-separated list of attributes.

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

\@xdylocref Define list of markup location references.

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

\@gls@ifinlist

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

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

```

Suggested by egreg:

```

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

```

`xdylettergroups` Store letter group definitions.

```

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

```

`sAddLetterGroup` Add a new letter group. The first argument is the name of the letter group. The second argument is the xindy code specifying prefixes and ordering.

```

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

```

1.5 Loops and conditionals

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

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

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

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

`\forall acronyms`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\ifglossaryexists` Starred form includes ignored glossaries.

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

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

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

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

`\glsdetoklabel`

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

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

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

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

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

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

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

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

```

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

```

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

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

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

```

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

```

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

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

```

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

```

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

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

```

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

```

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

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

```

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

```

```

1734 }%
1735 }

```

arynoexistsordo

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

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

```

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

```

fglshaschildren

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

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

```

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

```

\ifglshasparent

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

```

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

```

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

```

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

```

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

```

1775 \newcommand*{\ifglsdessuppressed}[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 \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*\ignoredglossary}[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*\ignoredglossary}[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@defdocnewglossaryentry}{\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@glstetoklabel{#1}@#2\endcsname
2519   \expandafter\expandonce\csname glo@glstetoklabel{#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{\glstetoklabel{#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 }
```

`\ifglsfielddefeq{<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{\ifglsfielddefeq}[5]{%
2536   \glsdoifexists{#1}%
2537   {%
2538     \protected@edef\@glo@label{\glstetoklabel{#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 }
```

glswritedefhook

```
2562 \newcommand*{\glswritedefhook}{}%
```

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{@glotype@\@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\@glsnoname
2730 \@glsnoname
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\@glsnodeldesc
2746 \@glsnodeldesc
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 \ifdefined@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>]

```

`\glsresetall`

```

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

```

As above, but with only a local effect:

```

\glslocalresetall

```

```

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 \TeX counter or even an explicit \TeX count register but is just a macro. Any of the commands that use `\glsunset` or `\glslocalunset`, such as `\gls`, will automatically increment this value. Commands that don’t modify the first use flag (such as `\glstext` or `\glsentrytext`) don’t modify this value.

`\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@spl@##1##2[##3]{%
3005   \ifnum\glsentryprevcount{##2}=1\relax
3006     \cgl@splformat{##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.¹

`\loadglsentries[<type>]{<filename>}`

This command will input the file using `\input`. The optional argument specifies to which glossary the entries should be assigned if they haven't used the type key. If the optional argument is not specified, the default glossary is used. Only those entries used in the document (via `\glslink`, `\gls`, `\glspl` and uppercase variants or `\glsadd` and `\glsaddall` will appear in the glossary). The mandatory argument is the filename (with or without `.tex` extension).

`\loadglsentries`

```
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

¹and 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}
```

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

```
3109 \newcommand*{\glsentryfmt}{%
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{\glsentryplural{\glslabel}}}%
3122         {\glsentrydescplural{\glslabel}}}%
3123         {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
3124     }%
3125     {%
```

First use

```
3126         #1{\glsentryfirstplural{\glslabel}}}%
3127         {\glsentrydescplural{\glslabel}}}%
3128         {\glsentrysymbolplural{\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 `\defglsentryfmt`, 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{glscompatible-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{glscompatible-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 }

```

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

```

3326 \newcommand*{\glsngenacfmt}{%
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 `\glstype`.

```
4080   \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.

```
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 \gl 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}{\glsentryfirst{#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\gl@customtext{%
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\gl@label{#2}%
4458 \let\gl@sifplural\@secondoftwo
4459 \let\gl@scapscase\@secondofthree
4460 \let\gl@insert\@empty
4461 \def\gl@customtext{%
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@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
4476     \def\glslabel{#2}%
4477     \let\glsifplural\@secondoftwo
4478     \let\glsapscase\@thirdofthree
4479     \let\glsinsert\@empty
4480     \def\glscustomtext{%
4481       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
4482     }%
```

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

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

Short plural:

\acrshortpl

```
4487 \newrobustcmd*{\acrshortpl}{\@gls@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@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
4495     \def\glslabel{#2}%
4496     \let\glsifplural\@firstoftwo
4497     \let\glsapscase\@firstofthree
4498     \let\glsinsert\@empty
4499     \def\glscustomtext{%
4500       \acronymfont{\glsentryshortpl{#2}}#3%
4501     }%
```

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

```
4502   \@gls@link[#1]{#2}{\csname gls@\glsstyle @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   \glsdoifexists{#2}%  
4512   {%  
  
4513     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper  
  
4514     \def\glslabel{#2}%  
4515     \let\glsifplural\@firstoftwo  
4516     \let\gls caps case\@secondofthree  
4517     \let\glsinsert\@empty  
4518     \def\gls custom text{%  
4519       \acronymfont{\Glsentryshortpl{#2}}#3%  
4520     }%
```

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

```
4521   \@gls@link[#1]{#2}{\csname gls@\glsstyle @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   \glsdoifexists{#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 \glsdoifexists{#2}%
4607 {%

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

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

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

```
4613 \def\glscustomtext{%
4614 \glentrylongpl{#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 }
```

\glentryserv 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*{\glentryserv}[1]{%
4819   \@gls@entry@field{#1}{serv}%
4820 }
```

\Glsentryserv

```
4821 \newrobustcmd*{\Glsentryserv}[1]{%
4822   \@Gls@entry@field{#1}{serv}%
4823 }
```

\glentryservi 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*{\glentryservi}[1]{%
4825   \@gls@entry@field{#1}{servi}%
4826 }
```

\Glsentryservi

```
4827 \newrobustcmd*{\Glsentryservi}[1]{%
4828   \@Gls@entry@field{#1}{servi}%
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*\glstentrynumberlist}[1]{%
4859   \glstoifexists{#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\glstentrynumberlist\space instead%
4871     }%
4872     \glstentrynumberlist{#1}%
4873   }%
4874 }%
4875 {%
4876   \newcommand*\glsdisplaynumberlist}[1]{%
4877     \glstoifexists{#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"glslnumbers\string"}
5194     \write\glswrite{page_compositor \string"glscpositor\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 \ifglxindy
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:=\@gls@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 \ifglsautomake
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   \glscloclocationsfalse
```

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 \@glo@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 `\theglsentrycounter` 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 `\glspercentchar` 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@gls@debug
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 ($\langle csname \rangle \c@page$ 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@loc}}}%
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 \ifglxsindy
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@set@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@dotoctitle
```

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{\glstoklabel{##1}}%
6079   \gls@org@glossaryentryfield{##1}%
6080 }%
6081 \renewcommand{\subglossentry}[2]{%
6082   \protected@xdef\glscurrententrylabel{\glstoklabel{##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@glstetoklabel{\#1}@sort}%
6203   \letcs\@gls@sort@B{glo@glstetoklabel{\#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@glstetoklabel{\#1}@sort}%
6213   \letcs\@gls@sort@B{glo@glstetoklabel{\#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@glstetoklabel{\#1}@sort}%
6223   \letcs\@gls@sort@B{glo@glstetoklabel{\#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@glstetoklabel{\#1}@sort}%
6233   \letcs\@gls@sort@B{glo@glstetoklabel{\#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@glstetoklabel{#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@glstetoklabel{#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@glstdetoklabel{#2}@loclist}%
6463 {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%
6464 {}%
6465 \listcsgadd{glo@glstdetoklabel{#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 nonnumberlist key determines if this glossary should have a number list.

```

6523 \define@boolkey{printgloss}[gls]{nonnumberlist}[true]{%
6524   \ifglsnonnumberlist
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 }

```

@glsnonnextpages 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 \glsnonnextpages is place in the entry's description and 3 column tabular style glossary is used.) \org@glossaryentrynumbers needs to be set at the start of each glossary, in the event that \glossaryentrynumber is re-defined.

```

6538 \newcommand*{\@glsnonnextpages}{%
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 place in the entry's description and 3 column tabular style glossary is used.) \org@glossaryentrynumbers needs to be set at the start of each glossary, in the event that \glossaryentrynumber is re-defined.

```

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\csuse{@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\glspacspace{##1}%
7087     (\protect\firstacronymfont{\glentryshort{##1}})%
7088   }%
7089   \renewcommand*{\Genacrfullformat}[2]{%
7090     \Glentrylong{##1}##2\glspacspace{##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 \Glstrylongpl{\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 \Glstrylong{\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\aclong`
 Plural long form

`\aclp`
 7456 `\let\aclp\aclongpl`
 First letter upper case long form

`\Acl`
 7457 `\let\Acl\Aclong`
 First letter upper case plural long form

`\Aclp`
 7458 `\let\Aclp\Aclongpl`
 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@assign@firstpl\gl@assign@firstpl
7603 \let\@org@gl@assign@plural\gl@assign@plural
7604 \let\@org@gl@assign@symbolplural\gl@assign@symbolplural
7605 \def\gl@assign@firstpl##1##2{%
7606   \@@gl@expand@field{##1}{firstpl}{##2}%
7607 }%
7608 \def\gl@assign@plural##1##2{%
7609   \@@gl@expand@field{##1}{plural}{##2}%
7610 }%
7611 \def\gl@assign@symbolplural##1##2{%
7612   \@@gl@expand@field{##1}{symbolplural}{##2}%
7613 }%
7614 \do@newglossaryentry
7615 \let\gl@assign@plural\@org@gl@assign@plural
7616 \let\gl@assign@firstpl\@org@gl@assign@firstpl
7617 \let\gl@assign@symbolplural\@org@gl@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@smallcaps
7621     \@gl@deprecated@acr@warning{footnote,smallcaps,description}%
7622     {footnote-sc-desc}{short-sc-footnote-desc}%
7623   \else
7624     \ifgl@smaller
7625       \@gl@deprecated@acr@warning{footnote,smaller,description}%
7626       {footnote-sm-desc}{short-sm-footnote-desc}%
7627     \else
7628       \@gl@deprecated@acr@warning{footnote,description}%
7629       {footnote-desc}{short-footnote-desc}%
7630     \fi
7631   \fi
7632 \renewcommand{\newacronym}[4][{}]{%
7633   \ifx\@gl@acronymlists\@empty
7634     \def\@glo@type{\acronymtype}%
7635     \setkeys{glossentry}{##1}%
7636     \DeclareAcronymList{\@glo@type}%
7637     \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
7638   \fi
7639   \gl@keylisttok{##1}%
7640   \gl@labeltok{##2}%
7641   \gl@shorttok{##3}%
7642   \gl@longtok{##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{\label{}}

```

8243 \newcommand*{\showgloplural}[1]{%
8244   \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
8245 }

```

\showglofirst \showglofirst{\label{}}

```

8246 \newcommand*{\showglofirst}[1]{%
8247   \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
8248 }

```

\showglofirstpl \showglofirstpl{\label{}}

```

8249 \newcommand*{\showglofirstpl}[1]{%
8250   \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
8251 }

```

\showglotype \showglotype{\label{}}

```

8252 \newcommand*{\showglotype}[1]{%
8253   \expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
8254 }

```

\showglocounter \showglocounter{\label{}}

```

8255 \newcommand*{\showglocounter}[1]{%
8256   \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
8257 }

```

\showglouser \showglouser{\label{}}

```

8258 \newcommand*{\showglouser}[1]{%

```

```
8259 \expandafter\show\csname glo@glstetoklabel{#1}@useri\endcsname
8260 }
```

```
\showglouserii \showglouserii{<label>}
```

```
8261 \newcommand*{\showglouserii}[1]{%
8262 \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
8263 }
```

```
\showglouseriii \showglouseriii{<label>}
```

```
8264 \newcommand*{\showglouseriii}[1]{%
8265 \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
8266 }
```

```
\showglouseriv \showglouseriv{<label>}
```

```
8267 \newcommand*{\showglouseriv}[1]{%
8268 \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
8269 }
```

```
\showglouserv \showglouserv{<label>}
```

```
8270 \newcommand*{\showglouserv}[1]{%
8271 \expandafter\show\csname glo@glstetoklabel{#1}@userv\endcsname
8272 }
```

```
\showglouservi \showglouservi{<label>}
```

```
8273 \newcommand*{\showglouservi}[1]{%
8274 \expandafter\show\csname glo@glstetoklabel{#1}@uservi\endcsname
8275 }
```

```
\showgloname \showgloname{<label>}
```

```
8276 \newcommand*{\showgloname}[1]{%
8277 \expandafter\show\csname glo@glstetoklabel{#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@glstetoklabel{#1}@long\endcsname
8299 }
```

`\showgloindex` `\showgloindex{<label>}`

```
8300 \newcommand*{\showgloindex}[1]{%
8301   \expandafter\show\csname glo@glstetoklabel{#1}@index\endcsname
8302 }
```

`\showgloflag` `\showgloflag{<label>}`

```
8303 \newcommand*{\showgloflag}[1]{%
8304   \expandafter\show\csname ifglo@glstetoklabel{#1}@flag\endcsname
8305 }
```

`\showgloloclist` `\showgloloclist{<label>}`

```
8306 \newcommand*{\showgloloclist}[1]{%
8307   \expandafter\show\csname glo@glstetoklabel{#1}@loclist\endcsname
8308 }
```

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

```
8309 \newcommand*{\showglofield}[2]{%
8310   \csshow{glo@glstetoklabel{#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 ifglcompatible-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 “`\gls{<label>}`” on first use but use “`\an\gls{<label>}`” on subsequent use.

```
8336 \NeedsTeXFormat{LaTeX2e}
```

```
8337 \ProvidesPackage{glossaries-prefix}[2021/10/19 v4.48 (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's

```
8455 \newrobustcmd{\Pgl's}{\@gls@hyp@opt\Pgl's}
```

\@Pgl's Unstarred version.

```
8456 \newcommand*{\@Pgl's}[2][]{%  
8457   \new@ifnextchar[%  
8458   {\@Pgl's@{#1}{#2}}%  
8459   {\@Pgl's@{#1}{#2}[]}%  
8460 }
```

\@Pgl's@ Read in the final optional argument:

```
8461 \def\Pgl's@#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:

\Pgl'spl

```
8486 \newrobustcmd{\Pgl'spl}{\@gls@hyp@opt\Pgl'spl}
```

\@Pgl'spl Unstarred version.

```
8487 \newcommand*{\@Pgl'spl}[2][]{%  
8488   \new@ifnextchar[%  
8489   {\@Pgl'spl@{#1}{#2}}%  
8490   {\@Pgl'spl@{#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     }
```

```

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     \glsoifexists{#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}[2021/10/19 v4.48 (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}}
```

`avhyperlinkname` 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}
```

`snavhypertarget`

```
\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@hypergrouplist@\@glo@type}%
8600   {%
8601     \def\@gls@list{}%
8602   }%
8603   {%
8604     \expandafter\let\expandafter\@gls@list
8605     \csname \@gls@hypergrouplist@\@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}{\@glsgetgrouptitle{glssymbols}}%
8617   \glshypernavsep
8618   \glsnavhyperlink{glsnumbers}{\@glsgetgrouptitle{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}[2021/10/19 v4.48 (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{v4.48}{2021-10-19}
```

Declare package:

```
8766 \ProvidesPackage{glossary-list}[2021/10/19 v4.48 (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 }

```

`tgrouphaderfmt` Provide a way of adjusting the format of the group headings.

```
8770 \newcommand*{\glslistgrouphaderfmt}[1]{#1}
```

`tnavigationitem` Provide a way of adjusting the format of the navigation header. This puts the navigation line inside the optional argument of `item` to prevent unwanted space occurring at the start, but this can cause a problem if the navigation line is too long. With this command, it makes it easier for the user to customise the style without having to remember to modify `\glossaryheader` after the style has been set.

```
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\glslistgrouphaderfmt\@firstofone
8781       \let\glsgetgrouptitle\@firstofone
8782       \let\glsnavhypertarget\@secondoftwo
8783     }%
8784   }%
8785   {}%
8786 }
```

`listexpandedname`

```
8787 \newcommand{\glslistexpandedname}[1]{%
8788   \glsunexpandedfieldvalue{#1}{name}%
8789 }
```

`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.

```
8790 \newglossarystyle{list}{%
  Use description environment:
8791   \renewenvironment{theglossary}%
8792     {\glslistinit\begin{description}}{\end{description}}%
  No header at the start of the environment:
8793   \renewcommand*{\glossaryheader}{}%
  No group headings:
8794   \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries start a new item in the list:

```
8795 \renewcommand*{\glossentry}[2]{%
8796   \item[\glssentryitem{##1}%
8797     \glstarget{##1}{\glossentryname{##1}}]
8798     \glossentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries continue on the same line:

```
8799 \renewcommand*{\subglossentry}[3]{%
8800   \glssubentryitem{##2}%
8801   \glstarget{##2}{\strut}\space
8802   \glossentrydesc{##2}\glspostdescription\space ##3.}%
```

Add vertical space between groups:

```
8803 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
8804 }
```

listgroup The listgroup style is like the list style, but the glossary groups have headings.

```
8805 \newglossarystyle{listgroup}{%
```

Base it on the list style:

```
8806 \setglossarystyle{list}%
```

Each group has a heading:

```
8807 \renewcommand*{\glsgroupheading}[1]{%
8808   \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.

```
8809 \newglossarystyle{listhypergroup}{%
```

Base it on the list style:

```
8810 \setglossarystyle{list}%
```

Add navigation links at the start of the environment.

```
8811 \renewcommand*{\glossaryheader}{%
8812   \glslistnavigationitem{\glslnavigation}}%
```

Each group has a heading with a hypertarget:

```
8813 \renewcommand*{\glsgroupheading}[1]{%
8814   \item[\glslistgroupheaderfmt
8815     {\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.

```
8816 \newglossarystyle{altlist}{%
```

Base it on the list style:

```
8817 \setglossarystyle{list}%
```


Main (level 0) entries start a new item in the list with a line break after the entry name:

```
8818 \renewcommand*{\glossentry}[2]{%
8819 \item[\glssentryitem{##1}%
8820 \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.

```
8821 \mbox{}\par\nobreak\@afterheading
8822 \glossentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries start a new paragraph:

```
8823 \renewcommand{\subglossentry}[3]{%
8824 \par
8825 \glssubentryitem{##2}%
8826 \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
8827 }
```

`altlistgroup` The `altlistgroup` glossary style is like the `altlist` style, but the glossary groups have headings.

```
8828 \newglossarystyle{altlistgroup}{%
```

Base it on the `altlist` style:

```
8829 \setglossarystyle{altlist}%
```

Each group has a heading:

```
8830 \renewcommand*{\glsgroupheading}[1]{%
8831 \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.

```
8832 \newglossarystyle{altlisthypergroup}{%
```

Base it on the `altlist` style:

```
8833 \setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
8834 \renewcommand*{\glossaryheader}{%
8835 \glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a `hypertarget`:

```
8836 \renewcommand*{\glsgroupheading}[1]{%
8837 \item[\glslistgroupheaderfmt
8838 {\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.

```
8839 \newglossarystyle{listdotted}{%
```

Base it on the `list` style:

```
8840 \setglossarystyle{list}%
```

Each main (level 0) entry starts a new item:

```

8841 \renewcommand*{\glossentry}[2]{%
8842   \item[]\makebox[\glslistdottedwidth][l]{%
8843     \glsentryitem{##1}%
8844     \glstarget{##1}{\glossentryname{##1}}%
8845     \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%

```

Sub entries have the same format as main entries:

```

8846 \renewcommand*{\subglossentry}[3]{%
8847   \item[]\makebox[\glslistdottedwidth][l]{%
8848     \glssubentryitem{##2}%
8849     \glstarget{##2}{\glossentryname{##2}}%
8850     \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
8851 }

```

listdottedwidth

```

8852 \newlength\glslistdottedwidth
8853 \setlength{\glslistdottedwidth}{.5\hsize}

```

sublistdotted This style is similar to the `glostylelistdotted` style, except that the main entries just have the name displayed.

```
8854 \newglossarystyle{sublistdotted}{%
```

Base it on the `listdotted` style:

```
8855 \setglossarystyle{listdotted}%
```

Main (level 0) entries just display the name:

```

8856 \renewcommand*{\glossentry}[2]{%
8857   \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}}%
8858 }

```

3.5 Glossary Styles using `longtable` (the `glossary-long` package)

The glossary styles defined in the package used the `longtable` environment in the glossary.

```
8859 \ProvidesPackage{glossary-long}[2021/10/19 v4.48 (NLCT)]
```

Requires the package:

```
8860 \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`.)

```

8861 \@ifundefined{glsdescwidth}{%
8862   \newlength\glsdescwidth
8863   \setlength{\glsdescwidth}{0.6\hsize}
8864 }{}

```

`lspagelistwidth` This is a length that governs the width of the page list column.

```
8865 \@ifundefined{glspagelistwidth}{%
8866   \newlength{glspagelistwidth
8867   \setlength{glspagelistwidth}{0.1\hsize}
8868 }{}
```

`long` The long glossary style command which uses the `longtable` environment:

```
8869 \newglossarystyle{long}{%
```

Use `longtable` with two columns:

```
8870 \renewenvironment{theglossary}{%
8871   {\begin{longtable}{lp{glsglwdescwidth}}}%
8872   {\end{longtable}}%
```

Do nothing at the start of the environment:

```
8873 \renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
8874 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
8875 \renewcommand{\glossentry}[2]{%
8876   \glsglentryitem{##1}\glsgltarget{##1}{\glossentryname{##1}} &
8877   \glossentrydesc{##1}\glsglpostdescription\space ##2\tabularnewline
8878 }%
```

Sub entries displayed on the following row without the name:

```
8879 \renewcommand{\subglossentry}[3]{%
8880   &
8881   \glsglsubentryitem{##2}%
8882   \glsgltarget{##2}{\strut}\glossentrydesc{##2}\glsglpostdescription\space
8883   ##3\tabularnewline
8884 }%
```

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>)

```
8885 \ifglsgnogroupskip
8886   \renewcommand*{\glsgroupskip}{}%
8887 \else
8888   \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
8889 \fi
8890 }
```

`longborder` The `longborder` style is like the above, but with horizontal and vertical lines:

```
8891 \newglossarystyle{longborder}{%
```

Base it on the `glostylelong` style:

```
8892 \setglossarystyle{long}%
```

Use `longtable` with two columns with vertical lines between each column:

```
8893 \renewenvironment{theglossary}{%
8894   \begin{longtable}{|lp{glsglwdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
8895 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8896 }
```

`longheader` The `longheader` style is like the `long` style but with a header:

```
8897 \newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
8898 \setglossarystyle{long}%
```

Set the table's header:

```
8899 \renewcommand*{\glossaryheader}{%
8900 \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
8901 }
```

`longheaderborder` The `longheaderborder` style is like the `long` style but with a header and border:

```
8902 \newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
8903 \setglossarystyle{longborder}%
```

Set the table's header and add horizontal line to table's foot:

```
8904 \renewcommand*{\glossaryheader}{%
8905 \hline\bfseries \entryname & \bfseries
8906 \descriptionname\tabularnewline\hline
8907 \endhead
8908 \hline\endfoot}%
8909 }
```

`long3col` The `long3col` style is like `long` but with 3 columns

```
8910 \newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
8911 \renewenvironment{theglossary}%
8912 {\begin{longtable}{lp{\glsgdescwidth}p{\glspagelistwidth}}}%
8913 {\end{longtable}}%
```

No table header:

```
8914 \renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
8915 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
8916 \renewcommand{\glossentry}[2]{%
8917 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
8918 \glossentrydesc{##1} & ##2\tabularnewline
8919 }%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
8920 \renewcommand{\subglossentry}[3]{%
8921     &
8922     \glssubentryitem{##2}%
8923     \glstarget{##2}{\strut}\glossentrydesc{##2} &
8924     ##3\tabularnewline
8925 }%
```

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>)

```
8926 \ifglsgroupskip
8927 \renewcommand*{\glsgroupskip}{}%
8928 \else
8929 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
8930 \fi
8931 }
```

`long3colborder` The `long3colborder` style is like the `long3col` style but with a border:

```
8932 \newglossarystyle{long3colborder}{%
    Base it on the glostylelong3col style:
8933 \setglossarystyle{long3col}%
    Use a longtable with 3 columns with vertical lines around them:
8934 \renewenvironment{theglossary}%
8935 {\begin{longtable}{|l|p{\glstdescwidth}|p{\glspagelistwidth}|}%
8936 {\end{longtable}}%
    Place horizontal lines at the head and foot of the table:
8937 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8938 }
```

`long3colheader` The `long3colheader` style is like `long3col` but with a header row:

```
8939 \newglossarystyle{long3colheader}{%
    Base it on the glostylelong3col style:
8940 \setglossarystyle{long3col}%
    Set the table's header:
8941 \renewcommand*{\glossaryheader}{%
8942     \bfseries\entryname&\bfseries\descriptionname&
8943     \bfseries\pagelistname\tabularnewline\endhead}%
8944 }
```

`colheaderborder` The `long3colheaderborder` style is like the above but with a border

```
8945 \newglossarystyle{long3colheaderborder}{%
    Base it on the glostylelong3colborder style:
8946 \setglossarystyle{long3colborder}%
}
```

Set the table's header and add horizontal line at table's foot:

```

8947 \renewcommand*{\glossaryheader}{%
8948     \hline
8949     \bfseries\entryname&\bfseries\descriptionname&
8950     \bfseries\pagelistname\tabularnewline\hline\endhead
8951     \hline\endfoot}%
8952 }
```

`long4col` The `long4col` style has four columns where the third column contains the value of the associated symbol key.

```

8953 \newglossarystyle{long4col}{%
```

Use a longtable with 4 columns:

```

8954 \renewenvironment{theglossary}%
8955     {\begin{longtable}{l1l1l}}%
8956     {\end{longtable}}%
```

No table header:

```

8957 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```

8958 \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):

```

8959 \renewcommand{\glossentry}[2]{%
8960     \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
8961     \glossentrydesc{##1} &
8962     \glossentrysymbol{##1} &
8963     ##2\tabularnewline
8964 }%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```

8965 \renewcommand{\subglossentry}[3]{%
8966     &
8967     \glssubentryitem{##2}%
8968     \glstarget{##2}{\strut}\glossentrydesc{##2} &
8969     \glossentrysymbol{##2} & ##3\tabularnewline
8970 }%
```

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>)

```

8971 \ifglsnogroupskip
8972     \renewcommand*{\glsgroupskip}{}%
8973 \else
8974     \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
8975 \fi
8976 }
```

long4colheader The long4colheader style is like long4col but with a header row.

```

8977 \newglossarystyle{long4colheader}{%
      Base it on the glostylelong4col style:
8978   \setglossarystyle{long4col}%
      Table has a header:
8979   \renewcommand*{\glossaryheader}{%
8980     \bfseries\entryname&\bfseries\descriptionname&
8981     \bfseries \symbolname&
8982     \bfseries\pagelistname\tabularnewline\endhead}%
8983 }
```

long4colborder The long4colborder style is like long4col but with a border.

```

8984 \newglossarystyle{long4colborder}{%
      Base it on the glostylelong4col style:
8985   \setglossarystyle{long4col}%
      Use a longtable with 4 columns surrounded by vertical lines:
8986   \renewenvironment{theglossary}%
8987     {\begin{longtable}{|l|l|l|l|}}%
8988     {\end{longtable}}%
      Add horizontal lines to the head and foot of the table:
8989   \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
8990 }
```

colheaderborder The long4colheaderborder style is like the above but with a border.

```

8991 \newglossarystyle{long4colheaderborder}{%
      Base it on the glostylelong4col style:
8992   \setglossarystyle{long4col}%
      Use a longtable with 4 columns surrounded by vertical lines:
8993   \renewenvironment{theglossary}%
8994     {\begin{longtable}{|l|l|l|l|}}%
8995     {\end{longtable}}%
      Add table header and horizontal line at the table's foot:
8996   \renewcommand*{\glossaryheader}{%
8997     \hline\bfseries\entryname&\bfseries\descriptionname&
8998     \bfseries \symbolname&
8999     \bfseries\pagelistname\tabularnewline\hline\endhead
9000     \hline\endfoot}%
9001 }
```

altlong4col The altlong4col style is like the long4col style but can have multiline descriptions and page lists.

```

9002 \newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
9003 \setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9004 \renewenvironment{theglossary}%
9005   {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
9006   {\end{longtable}}%
9007 }
```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```
9008 \newglossarystyle{altlong4colheader}{%
```

Base it on the `glostylelong4colheader` style:

```
9009 \setglossarystyle{long4colheader}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9010 \renewenvironment{theglossary}%
9011   {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
9012   {\end{longtable}}%
9013 }
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```
9014 \newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
9015 \setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9016 \renewenvironment{theglossary}%
9017   {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}}%
9018   {\end{longtable}}%
9019 }
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
9020 \newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `glostylelong4colheaderborder` style:

```
9021 \setglossarystyle{long4colheaderborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
9022 \renewenvironment{theglossary}%
9023   {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}}%
9024   {\end{longtable}}%
9025 }
```


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>

```
9026 \ProvidesPackage{glossary-longbooktabs}[2021/10/19 v4.48 (NLCT)]
```

Requires booktabs package:

```
9027 \RequirePackage{booktabs}
```

and the base packages for long styles:

```
9028 \RequirePackage{glossary-long}
```

```
9029 \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.

```
9030 \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.

```
9031 \glspatchLToutput
```

As with the longheader style, use the long style as a base.

```
9032 \setglossarystyle{long}{%
```

Add a header with rules.

```
9033 \renewcommand*{\glossaryheader}{%
```

```
9034 \toprule \bfseries \entryname & \bfseries
```

```
9035 \descriptionname\tabularnewline\midrule\endhead
```

```
9036 \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.

```
9037 \ifglsgnolgroupskip
```

```
9038 \renewcommand*{\glsgroupskip}{}%
```

```
9039 \else
```

```
9040 \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
```

```
9041 \fi
```

```
9042 }
```

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.

```
9043 \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.

```
9044 \glspatchLToutput
```

Use the long3col style as a base.

```
9045 \setglossarystyle{long3col}%
```

Add a header with rules.

```
9046 \renewcommand*{\glossaryheader}{%  
9047   \toprule \bfseries \entryname &  
9048   \bfseries \descriptionname &  
9049   \bfseries \pagelistname  
9050   \tabularnewline\midrule\endhead  
9051   \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.

```
9052 \ifglsgnogroupskip  
9053   \renewcommand*{\glsgroupskip}{}%  
9054 \else  
9055   \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%  
9056 \fi  
9057 }
```

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.

```
9058 \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.

```
9059 \glspatchLToutput
```

Use the long4col style as a base.

```
9060 \setglossarystyle{long4col}%
```

Add a header with rules.

```
9061 \renewcommand*{\glossaryheader}{%  
9062   \toprule \bfseries \entryname &  
9063   \bfseries \descriptionname &  
9064   \bfseries \symbolname &  
9065   \bfseries \pagelistname  
9066   \tabularnewline\midrule\endhead  
9067   \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.

```
9068 \ifglsgnogroupskip  
9069   \renewcommand*{\glsgroupskip}{}%  
9070 \else  
9071   \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%  
9072 \fi  
9073 }
```

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.

```
9074 \newglossarystyle{altlong4col-booktabs}{%
```

The patch \glspatchLToutput is already applied in long4col-booktabs and so doesn't need to be here.

```
9075 \glspatchLToutput
```

Use the long4col-booktabs style as a base.

```
9076 \setglossarystyle{long4col-booktabs}{%
```

Change the column specifications:

```
9077 \renewenvironment{theglossary}{%
```

```
9078 {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
```

```
9079 {\end{longtable}}%
```

```
9080 }
```

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.

```
9081 \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.

```
9082 \glspatchLToutput
```

Use the long-booktabs style as a base.

```
9083 \setglossarystyle{long-booktabs}{%
```

Adjust the column specification.

```
9084 \renewenvironment{theglossary}{%
```

```
9085 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}}}%
```

```
9086 {\end{longtable}}%
```

```
9087 }
```

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.

```
9088 \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.

```
9089 \glspatchLToutput
```

Use the long3col-booktabs style as a base.

```
9090 \setglossarystyle{long3col-booktabs}{%
```

Adjust the column specification.

```
9091 \renewenvironment{theglossary}{%
```

```
9092 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}}%
```

```
9093 >{\raggedright}p{\glspagelistwidth}}}%
```

```
9094 {\end{longtable}}%
```

```
9095 }
```

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.

```
9096 \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.

```
9097 \glspatchLToutput
```

Use the altlong4col-booktabs style as a base.

```
9098 \setglossarystyle{altlong4col-booktabs}{%
```

Adjust the column specification.

```
9099 \renewenvironment{theglossary}{%
```

```
9100 {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}l%
```

```
9101 >{\raggedright}p{\glspagelistwidth}}}%
```

```
9102 {\end{longtable}}}%
```

```
9103 }
```

sLTpenaltycheck

```
9104 \newcommand*{\glslTpenaltycheck}{%
```

```
9105 \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
```

```
9106 }
```

enaltygroupskip

```
9107 \newcommand{\glspenaltygroupskip}{%
```

```
9108 \noalign{\penalty-50\vskip\normalbaselineskip}}
```

restoreLToutput Provide a way of restoring \LT@output for the user.

```
9109 \let\@gls@org@LT@output\LT@output
```

```
9110 \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

```
9111 \newcommand*{\glspatchLToutput}{%
```

```
9112 \renewcommand*{\LT@output}{%
```

```
9113 \ifnum\outputpenalty <-\@Mi
```

```
9114 \ifnum\outputpenalty > -\LT@end@pen
```

```
9115 \LT@err{floats and marginpars not allowed in a longtable}\@ehc
```

```
9116 \else
```

```
9117 \setbox\z@\vbox{\unvbox\@cclv}%
```

```
9118 \ifdim \ht\LT@lastfoot>\ht\LT@foot
```

```
9119 \dimen@\pagegoal
```

```
9120 \advance\dimen@-\ht\LT@lastfoot
```

```
9121 \ifdim\dimen@<\ht\z@
```

```
9122 \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
```

```
9123 \makecol
```

```
9124 \@outputpage
```

```

9125         \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
9126     \fi
9127 \fi
9128 \global\@colroom\@colht
9129 \global\vsizel\@colht
9130 {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
9131 \fi
9132 \else
9133     \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
9134     \@makecol
9135     \@outputpage
9136     \global\vsizel\@colroom
9137     \copy\LT@head
9138     \glsLTpenaltycheck
9139     \nobreak
9140 \fi
9141 }%
9142 }

```

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.

9143 \ProvidesPackage{glossary-longragged}[2021/10/19 v4.48 (NLCT)]

Requires the package:

9144 \RequirePackage{array}

Requires the package:

9145 \RequirePackage{longtable}

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```

9146 \@ifundefined{glsdescwidth}{%
9147     \newlength\glsdescwidth
9148     \setlength{\glsdescwidth}{0.6\hsize}
9149 }{}

```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```

9150 \@ifundefined{glspagelistwidth}{%
9151     \newlength\glspagelistwidth
9152     \setlength{\glspagelistwidth}{0.1\hsize}
9153 }{}

```

`longragged` The longragged glossary style is like the long but uses ragged right formatting for the description column.

```
9154 \newglossarystyle{longragged}{%
```

Use longtable with two columns:

```
9155 \renewenvironment{theglossary}{%
9156     {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%
9157     {\end{longtable}}}%
```

Do nothing at the start of the environment:

```
9158 \renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
9159 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
9160 \renewcommand{\glossentry}[2]{%
9161     \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9162     \glossentrydesc{##1}\glspostdescription\space ##2%
9163     \tabularnewline
9164 }%
```

Sub entries displayed on the following row without the name:

```
9165 \renewcommand{\subglossentry}[3]{%
9166     &
9167     \glssubentryitem{##2}%
9168     \glstarget{##2}{\strut}\glossentrydesc{##2}%
9169     \glspostdescription\space ##3%
9170     \tabularnewline
9171 }%
```

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>)

```
9172 \ifglsgroupskip
9173     \renewcommand*{\glsgroupskip}{}%
9174 \else
9175     \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
9176 \fi
9177 }
```

ongraggedborder The longraggedborder style is like the above, but with horizontal and vertical lines:

```
9178 \newglossarystyle{longraggedborder}{%
```

Base it on the glostylelongragged style:

```
9179 \setglossarystyle{longragged}%
```

Use longtable with two columns with vertical lines between each column:

```
9180 \renewenvironment{theglossary}{%
9181     \begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}}}%
9182     {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
9183 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9184 }
```

longraggedheader The longraggedheader style is like the longragged style but with a header:

```
9185 \newglossarystyle{longraggedheader}{%  
    Base it on the glostylelongragged style:  
9186 \setglossarystyle{longragged}%  
    Set the table's header:  
9187 \renewcommand*{\glossaryheader}{%  
9188 \bfseries \entryname & \bfseries \descriptionname  
9189 \tabularnewline\endhead}%  
9190 }
```

longraggedheaderborder The longraggedheaderborder style is like the longragged style but with a header and border:

```
9191 \newglossarystyle{longraggedheaderborder}{%  
    Base it on the glostylelongraggedborder style:  
9192 \setglossarystyle{longraggedborder}%  
    Set the table's header and add horizontal line to table's foot:  
9193 \renewcommand*{\glossaryheader}{%  
9194 \hline\bfseries \entryname & \bfseries \descriptionname  
9195 \tabularnewline\hline  
9196 \endhead  
9197 \hline\endfoot}%  
9198 }
```

longragged3col The longragged3col style is like longragged but with 3 columns

```
9199 \newglossarystyle{longragged3col}{%  
    Use a longtable with 3 columns:  
9200 \renewenvironment{theglossary}%  
9201 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}%  
9202 >{\raggedright}p{\glspagelistwidth}}}%  
9203 {\end{longtable}}%  
    No table header:  
9204 \renewcommand*{\glossaryheader}{}%  
    No headings between groups:  
9205 \renewcommand*{\glsgroupheading}[1]{}%  
    Main (level 0) entries on a row (name in first column, description in second column, page list  
    in last column):  
9206 \renewcommand{\glossentry}[2]{%  
9207 \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
9208 \glossentrydesc{##1} & ##2\tabularnewline  
9209 }%  
    Sub-entries on a separate row (no name, description in second column, page list in third  
    column):  
9210 \renewcommand{\subglossentry}[3]{%  
9211 &
```

```

9212 \glssubentryitem{##2}%
9213 \glstarget{##2}{\strut}\glossentrydesc{##2} &
9214 ##3\tabularnewline
9215 }%

```

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>)

```

9216 \ifglsgroupskip
9217 \renewcommand*{\glsgroupskip}{}%
9218 \else
9219 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
9220 \fi
9221 }

```

ragged3colborder The longragged3colborder style is like the longragged3col style but with a border:

```

9222 \newglossarystyle{longragged3colborder}{%
    Base it on the glostylelongragged3col style:
9223 \setglossarystyle{longragged3col}%
    Use a longtable with 3 columns with vertical lines around them:
9224 \renewenvironment{theglossary}%
9225 {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|%
9226 >{\raggedright}p{\glspagelistwidth}|}%
9227 {\end{longtable}}%

```

Place horizontal lines at the head and foot of the table:

```

9228 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9229 }

```

ragged3colheader The longragged3colheader style is like longragged3col but with a header row:

```

9230 \newglossarystyle{longragged3colheader}{%
    Base it on the glostylelongragged3col style:
9231 \setglossarystyle{longragged3col}%
    Set the table's header:
9232 \renewcommand*{\glossaryheader}{%
9233 \bfseries\entryname&\bfseries\descriptionname&
9234 \bfseries\pagelistname\tabularnewline\endhead}%
9235 }

```

colheaderborder The longragged3colheaderborder style is like the above but with a border

```

9236 \newglossarystyle{longragged3colheaderborder}{%
    Base it on the glostylelongragged3colborder style:
9237 \setglossarystyle{longragged3colborder}%
    Set the table's header and add horizontal line at table's foot:
9238 \renewcommand*{\glossaryheader}{%
9239 \hline
9240 \bfseries\entryname&\bfseries\descriptionname&

```



```

9241 \bfseries\pagelistname\tabularnewline\hline\endhead
9242 \hline\endfoot}%
9243 }

```

`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.

```

9244 \newglossarystyle{altlongragged4col}{%
    Use a longtable with 4 columns where the second and last columns may have multiple lines
    in each row:

```

```

9245 \renewenvironment{theglossary}%
9246 {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}l%
9247 >{\raggedright}p{\glspagelistwidth}}}%
9248 {\end{longtable}}%

```

No table header:

```

9249 \renewcommand*{\glossaryheader}{}%

```

No group headings:

```

9250 \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):

```

9251 \renewcommand{\glossentry}[2]{%
9252 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9253 \glossentrydesc{##1} & \glossentrysymbol{##1} &
9254 ##2\tabularnewline
9255 }%

```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```

9256 \renewcommand{\subglossentry}[3]{%
9257 &
9258 \glssubentryitem{##2}%
9259 \glstarget{##2}{\strut}\glossentrydesc{##2} &
9260 \glossentrysymbol{##2} & ##3\tabularnewline
9261 }%

```

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>)

```

9262 \ifglsgroupskip
9263 \renewcommand*{\glsgroupskip}{}%
9264 \else
9265 \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
9266 \fi
9267 }

```

`altlongragged4colheader` The `altlongragged4colheader` style is like `altlongragged4col` but with a header row.

```

9268 \newglossarystyle{altlongragged4colheader}{%

```

Base it on the `glostylealtlongragged4col` style:

```

9269 \setglossarystyle{altlongragged4col}%

```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9270 \renewenvironment{theglossary}%
9271   {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}l%
9272     >{\raggedright}p{\glspagelistwidth}}}%
9273   {\end{longtable}}%
```

Table has a header:

```
9274 \renewcommand*{\glossaryheader}{%
9275   \bfseries\entryname&\bfseries\descriptionname&
9276   \bfseries \symbolname&
9277   \bfseries\pagelistname\tabularnewline\endhead}%
9278 }
```

ragged4colborder The altlongragged4colborder style is like altlongragged4col but with a border.

```
9279 \newglossarystyle{altlongragged4colborder}{%
```

Base it on the glostylealtlongragged4col style:

```
9280 \setglossarystyle{altlongragged4col}%
```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9281 \renewenvironment{theglossary}%
9282   {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|l|}%
9283     >{\raggedright}p{\glspagelistwidth}|}%
9284   {\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```
9285 \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
9286 }
```

colheaderborder The altlongragged4colheaderborder style is like the above but with a header as well as a border.

```
9287 \newglossarystyle{altlongragged4colheaderborder}{%
```

Base it on the glostylealtlongragged4col style:

```
9288 \setglossarystyle{altlongragged4col}%
```

Use a longtable with 4 columns where the second and last columns may have multiple lines in each row:

```
9289 \renewenvironment{theglossary}%
9290   {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|l|}%
9291     >{\raggedright}p{\glspagelistwidth}|}%
9292   {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
9293 \renewcommand*{\glossaryheader}{%
9294   \hline\bfseries\entryname&\bfseries\descriptionname&
9295   \bfseries \symbolname&
9296   \bfseries\pagelistname\tabularnewline\hline\endhead
9297   \hline\endfoot}%
9298 }
```

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.

```
9299 \ProvidesPackage{glossary-mcols}[2021/10/19 v4.48 (NLCT)]
```

Required packages:

```
9300 \RequirePackage{multicol}
```

```
9301 \RequirePackage{glossary-tree}
```

`\indexspace` The are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
9302 \providecommand{\indexspace}{%
```

```
9303   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
```

```
9304 }
```

`\glsmcols` Define macro in which to store the number of columns. (Defaults to 2.)

```
9305 \newcommand*{\glsmcols}{2}
```

`mcolindex` Multi-column index style. Same as the index, but puts the glossary in multiple columns. (Ideally the glossary title should go in the optional argument of `multicols`, but the title isn't part of the glossary style.)

```
9306 \newglossarystyle{mcolindex}{%
```

```
9307   \setglossarystyle{index}%
```

```
9308   \renewenvironment{theglossary}%
```

```
9309     {%
```

```
9310       \begin{multicols}{\glsmcols}
```

```
9311       \setlength{\parindent}{0pt}%
```

```
9312       \setlength{\parskip}{0pt plus 0.3pt}%
```

```
9313       \let\item\glstreeitem
```

```
9314       \let\subitem\glstreesubitem
```

```
9315       \let\subsubitem\glstreesubsubitem
```

```
9316     }%
```

```
9317     {\end{multicols}}%
```

```
9318 }
```

`mcolindexgroup` As `mcolindex` but has headings:

```
9319 \newglossarystyle{mcolindexgroup}{%
```

```
9320   \setglossarystyle{mcolindex}%
```

```
9321   \renewcommand*{\glsgroupheading}[1]{%
```

```
9322     \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
```

```
9323 }
```

`indexhypergroup` The `mcolindexhypergroup` style is like the `mcolindexgroup` style but has hyper navigation.

```
9324 \newglossarystyle{mcolindexhypergroup}{%
```

Base it on the `glostylemcolindex` style:

```
9325 \setglossarystyle{mcolindex}%
```

Put navigation links to the groups at the start of the glossary:

```
9326 \renewcommand*{\glossaryheader}{%
9327   \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.

```
9328 \renewcommand*{\glsgroupheading}[1]{%
9329   \item\glstreegroupheaderfmt
9330     {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
9331   \indexspace}%
9332 }
```

`colindexspannav` Similar to `mcolindexhypergroup`, but puts the navigation line in the optional argument of `multicols`.

```
9333 \newglossarystyle{mcolindexspannav}{%
9334   \setglossarystyle{index}%
9335   \renewenvironment{theglossary}%
9336     {%
9337       \begin{multicols}{\glsmcols}\noindent\glstreenavigationfmt{\glsnavigation}]
9338       \setlength{\parindent}{0pt}%
9339       \setlength{\parskip}{0pt plus 0.3pt}%
9340       \let\item\glstreeitem}%
9341     {\end{multicols}}}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
9342 \renewcommand*{\glsgroupheading}[1]{%
9343   \item\glstreegroupheaderfmt
9344     {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
9345   \indexspace}%
9346 }
```

`mcoltree` Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
9347 \newglossarystyle{mcoltree}{%
9348   \setglossarystyle{tree}%
9349   \renewenvironment{theglossary}%
9350     {%
9351       \begin{multicols}{\glsmcols}
9352       \setlength{\parindent}{0pt}%
9353       \setlength{\parskip}{0pt plus 0.3pt}%
9354     }%
9355     {\end{multicols}}}%
9356 }
```

mcoltreegroup Like the **mcoltree** style but the glossary groups have headings.

```

9357 \newglossarystyle{mcoltreegroup}{%
    Base it on the glostylemcoltree style:
9358   \setglossarystyle{mcoltree}%

    Each group has a heading (in bold) followed by a vertical gap):
9359   \renewcommand{\glsgroupheading}[1]{\par
9360     \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
9361 }
```

ltreehypergroup The **mcoltreehypergroup** style is like the **treegroup** style, but has a set of links to the groups at the start of the glossary.

```

9362 \newglossarystyle{mcoltreehypergroup}{%
    Base it on the glostylemcoltree style:
9363   \setglossarystyle{mcoltree}%

    Put navigation links to the groups at the start of the theglossary environment:
9364   \renewcommand*{\glossaryheader}{%
9365     \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

    Each group has a heading (in bold with a target) followed by a vertical gap):
9366   \renewcommand*{\glsgroupheading}[1]{%
9367     \par\noindent
9368     \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9369     \indexspace}%
9370 }
```

mcoltreespannav Similar to the **mcoltreehypergroup** style but the navigation line is put in the optional argument of the **multicols** environment.

```

9371 \newglossarystyle{mcoltreespannav}{%
9372   \setglossarystyle{tree}%
9373   \renewenvironment{theglossary}%
9374   {%

9375     \begin{multicols}{\glsmcols}\noindent\glstreenavigationfmt{\glsnavigation}]
9376     \setlength{\parindent}{0pt}%
9377     \setlength{\parskip}{0pt plus 0.3pt}%
9378   }%
9379   {\end{multicols}}%

    Each group has a heading (in bold with a target) followed by a vertical gap):
9380   \renewcommand*{\glsgroupheading}[1]{%
9381     \par\noindent
9382     \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9383     \indexspace}%
9384 }
```

`mcoltreenoname` Multi-column index style. Same as the `treenoname`, but puts the glossary in multiple columns.

```

9385 \newglossarystyle{mcoltreenoname}{%
9386   \setglossarystyle{treenoname}%
9387   \renewenvironment{theglossary}%
9388   {%
9389     \begin{multicols}{\glsmcols}
9390     \setlength{\parindent}{0pt}%
9391     \setlength{\parskip}{0pt plus 0.3pt}%
9392   }%
9393   {\end{multicols}}%
9394 }
```

`treenonamegroup` Like the `mcoltreenoname` style but the glossary groups have headings.

```

9395 \newglossarystyle{mcoltreenonamegroup}{%
    Base it on the glostylemcoltreenoname style:
9396   \setglossarystyle{mcoltreenoname}%
    Give each group a heading:
9397   \renewcommand{\glsgroupheading}[1]{\par
9398     \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
9399 }
```

`onamehypergroup` The `mcoltreenonamehypergroup` style is like the `mcoltreenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```

9400 \newglossarystyle{mcoltreenonamehypergroup}{%
    Base it on the glostylemcoltreenoname style:
9401   \setglossarystyle{mcoltreenoname}%
    Put navigation links to the groups at the start of the theglossary environment:
9402   \renewcommand*{\glossaryheader}{%
9403     \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    Each group has a heading (in bold with a target) followed by a vertical gap):
9404   \renewcommand*{\glsgroupheading}[1]{%
9405     \par\noindent
9406     \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
9407     \indexspace}%
9408 }
```

`eenonamespannav` Similar to the `mcoltreenonamehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

9409 \newglossarystyle{mcoltreenonamespannav}{%
9410   \setglossarystyle{treenoname}%
9411   \renewenvironment{theglossary}%
9412   {%
9413     \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
9414     \setlength{\parindent}{0pt}%
9415     \setlength{\parskip}{0pt plus 0.3pt}%

```

```

9416 }%
9417 {\end{multicols}}%
    Each group has a heading (in bold with a target) followed by a vertical gap):
9418 \renewcommand*{\glsgroupheading}[1]{%
9419     \par\noindent
9420     \glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
9421     \indexspace}%
9422 }

```

mcolalttree Multi-column index style. Same as the alttree, but puts the glossary in multiple columns.

```

9423 \newglossarystyle{mcolalttree}{%
9424     \setglossarystyle{alttree}%
9425     \renewenvironment{theglossary}%
9426     {%
9427         \begin{multicols}{\glsmcols}
9428         \def\@gls@prevlevel{-1}%
9429         \mbox{}\par
9430     }%
9431     {\par\end{multicols}}%
9432 }

```

colalttreegroup Like the mcolalttree style but the glossary groups have headings.

```

9433 \newglossarystyle{colalttreegroup}{%
    Base it on the glostylemcolalttree style:
9434     \setglossarystyle{mcolalttree}%
    Give each group a heading.
9435     \renewcommand{\glsgroupheading}[1]{\par
9436         \def\@gls@prevlevel{-1}%
9437         \hangindent0pt\relax
9438         \parindent0pt\relax
9439         \glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
9440 }

```

treehypergroup The mcolalttreehypergroup style is like the mcolalttreegroup style, but has a set of links to the groups at the start of the glossary.

```

9441 \newglossarystyle{mcolalttreehypergroup}{%
    Base it on the glostylemcolalttree style:
9442     \setglossarystyle{mcolalttree}%
    Put the navigation links in the header
9443     \renewcommand*{\glossaryheader}{%
9444         \par
9445         \def\@gls@prevlevel{-1}%
9446         \hangindent0pt\relax
9447         \parindent0pt\relax
9448         \glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Put a hypertarget at the start of each group

```

9449 \renewcommand*{\glsgroupheading}[1]{%
9450   \par
9451   \def\@gls@prevlevel{-1}%
9452   \hangindent0pt\relax
9453   \parindent0pt\relax
9454   \glstreegroupheaderfmt{\glsgrouphypertarget{##1}{\glsgroupgetgrouptitle{##1}}}\par
9455   \indexspace}%
9456 }
```

`\alttreespannav` Similar to the `\mcolalttreehypergroup` style but the navigation line is put in the optional argument of the multicols environment.

```

9457 \newglossarystyle{mcolalttreespannav}{%
9458   \setglossarystyle{alttree}%
9459   \renewenvironment{theglossary}%
9460   {%
9461     \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsgroupnavigation}]
9462     \def\@gls@prevlevel{-1}%
9463     \mbox{}\par
9464   }%
9465   {\par\end{multicols}}}%

```

Put a hypertarget at the start of each group

```

9466 \renewcommand*{\glsgroupheading}[1]{%
9467   \par
9468   \def\@gls@prevlevel{-1}%
9469   \hangindent0pt\relax
9470   \parindent0pt\relax
9471   \glstreegroupheaderfmt{\glsgrouphypertarget{##1}{\glsgroupgetgrouptitle{##1}}}\par
9472   \indexspace}%
9473 }
```

3.9 Glossary Styles using supertabular environment (glossary-super package)

The glossary styles defined in the package use the supertabular environment.

```

9474 \ProvidesPackage{glossary-super}[2021/10/19 v4.48 (NLCT)]

```

Requires the package:

```

9475 \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.

```

9476 \@ifundefined{glsdescwidth}{%
9477   \newlength\glsdescwidth
9478   \setlength{\glsdescwidth}{0.6\hsize}
9479 }{}

```


`glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if has been loaded.

```
9480 \@ifundefined{glspagelistwidth}{%
9481   \newlength{glspagelistwidth
9482   \setlength{glspagelistwidth}{0.1\hsize}
9483 }{}
```

`super` The super glossary style uses the supertabular environment (it uses lengths defined in the package.)

```
9484 \newglossarystyle{super}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
9485   \renewenvironment{theglossary}%
9486     {\tablehead{}\tabletail{}}%
9487     \begin{supertabular}{lp{glsglscwidth}}%
9488     {\end{supertabular}}%
```

Do nothing at the start of the table:

```
9489   \renewcommand*{glossaryheader}{}%
```

No group headings:

```
9490   \renewcommand*{glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
9491   \renewcommand{glossentry}[2]{%
9492     \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9493     \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
9494   }%
```

Sub entries put in a row (no name, description and page list in second column):

```
9495   \renewcommand{subglossentry}[3]{%
9496     &
9497     \glssubentryitem{##2}%
9498     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
9499     ##3\tabularnewline
9500   }%
```

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>)

```
9501   \ifglsgroupskip
9502     \renewcommand*{glsgroupskip}{}%
9503   \else
9504     \renewcommand*{glsgroupskip}{& \tabularnewline}%
9505   \fi
9506 }
```

`superborder` The superborder style is like the above, but with horizontal and vertical lines:

```
9507 \newglossarystyle{superborder}{%
```

Base it on the `glostylessuper` style:

```
9508   \setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:

```
9509 \renewenvironment{theglossary}%
9510   {\tablehead{\hline}\tabletail{\hline}%
9511    \begin{supertabular}{|l|p{\glsdescwidth}|}%
9512    {\end{supertabular}}%
9513 }
```

superheader The superheader style is like the super style, but with a header:

```
9514 \newglossarystyle{superheader}{%
```

Base it on the glostylessuper style:

```
9515 \setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and no tail:

```
9516 \renewenvironment{theglossary}%
9517   {\tablehead{\bfseries \entryname &
9518    \bfseries\descriptionname\tabularnewline}%
9519    \tabletail{}}%
9520    \begin{supertabular}{lp{\glsdescwidth}}%
9521    {\end{supertabular}}%
9522 }
```

superheaderborder The superheaderborder style is like the super style but with a header and border:

```
9523 \newglossarystyle{superheaderborder}{%
```

Base it on the glostylessuper style:

```
9524 \setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and horizontal lines above and below the table:

```
9525 \renewenvironment{theglossary}%
9526   {\tablehead{\hline\bfseries \entryname &
9527    \bfseries \descriptionname\tabularnewline\hline}%
9528    \tabletail{\hline}
9529    \begin{supertabular}{|l|p{\glsdescwidth}|}%
9530    {\end{supertabular}}%
9531 }
```

super3col The super3col style is like the super style, but with 3 columns:

```
9532 \newglossarystyle{super3col}{%
```

Put the glossary in a supertabular environment with three columns and no head or tail:

```
9533 \renewenvironment{theglossary}%
9534   {\tablehead{}\tabletail{}}%
9535   \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
9536   {\end{supertabular}}%
```

Do nothing at the start of the table:

```
9537 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9538 \renewcommand*{\glsgrouphheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
9539 \renewcommand{\glossentry}[2]{%
9540   \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9541   \glossentrydesc{##1} & ##2\tabularnewline
9542 }%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
9543 \renewcommand{\subglossentry}[3]{%
9544   &
9545   \glssubentryitem{##2}%
9546   \glstarget{##2}{\strut}\glossentrydesc{##2} &
9547   ##3\tabularnewline
9548 }%
```

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>)

```
9549 \ifglsnogroupskip
9550 \renewcommand*{\glsgroupskip}{}%
9551 \else
9552 \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9553 \fi
9554 }
```

super3colborder The super3colborder style is like the super3col style, but with a border:

```
9555 \newglossarystyle{super3colborder}{%
```

Base it on the glostylesuper3col style:

```
9556 \setglossarystyle{super3col}%
```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```
9557 \renewenvironment{theglossary}%
9558   {\tablehead{\hline}\tabletail{\hline}%
9559   \begin{supertabular}{|l|p{\glsgdescwidth}|p{\glspagelistwidth}|}%
9560   {\end{supertabular}}%
9561 }
```

super3colheader The super3colheader style is like the super3col style but with a header row:

```
9562 \newglossarystyle{super3colheader}{%
```

Base it on the glostylesuper3col style:

```
9563 \setglossarystyle{super3col}%
```

Put the glossary in a supertabular environment with three columns, a header and no tail:

```
9564 \renewenvironment{theglossary}%
9565   {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9566     \bfseries\pagelistname\tabularnewline}\tabletail{}}%
```

```

9567     \begin{supertabular}{lp{\glsgdescwidth}p{\glspagelistwidth}}}%
9568     {\end{supertabular}}}%
9569 }

```

colheaderborder The super3colheaderborder style is like the super3col style but with a header and border:

```

9570 \newglossarystyle{super3colheaderborder}{%
    Base it on the glostylesuper3colborder style:
9571     \setglossarystyle{super3colborder}%
    Put the glossary in a supertabular environment with three columns, a header with horizontal
    lines and a horizontal line in the tail:
9572     \renewenvironment{theglossary}%
9573     {\tablehead{\hline
9574         \bfseries\entryname&\bfseries\descriptionname&
9575         \bfseries\pagelistname\tabularnewline\hline}%
9576     \tabletail{\hline}%
9577     \begin{supertabular}{|lp{\glsgdescwidth}|p{\glspagelistwidth}|}%
9578     {\end{supertabular}}}%
9579 }

```

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.

```

9580 \newglossarystyle{super4col}{%
    Put the glossary in a supertabular environment with four columns and no head or tail:
9581     \renewenvironment{theglossary}%
9582     {\tablehead{}\tabletail{}}%
9583     \begin{supertabular}{|llll|}%
9584     \end{supertabular}}}%
    Do nothing at the start of the table:
9585     \renewcommand*{\glossaryheader}{}%
    No group headings:
9586     \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:
9587     \renewcommand{\glossentry}[2]{%
9588         \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9589         \glossentrydesc{##1} &
9590         \glossentrysymbol{##1} & ##2\tabularnewline
9591     }%
    Sub entries on a row with no name, the description in the second column, symbol in third
    column and page list in last column:
9592     \renewcommand{\subglossentry}[3]{%
9593         &
9594         \glssubentryitem{##2}%
9595         \glstarget{##2}{\strut}\glossentrydesc{##2} &

```

```

9596     \glossentrysymbol{##2} & ##3\tabularnewline
9597 }%

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)

9598 \ifglsgroupskip
9599   \renewcommand*{\glsgroupskip}{}%
9600 \else
9601   \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9602 \fi
9603 }

```

super4colheader The super4colheader style is like the super4col but with a header row.

```

9604 \newglossarystyle{super4colheader}{%

Base it on the glostylesuper4col style:

9605   \setglossarystyle{super4col}%

Put the glossary in a supertabular environment with four columns, a header and no tail:

9606   \renewenvironment{theglossary}%
9607     {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9608               \bfseries\symbolname &
9609               \bfseries\pagelistname\tabularnewline}%
9610     \tabletail{}}%
9611     \begin{supertabular}{1111}}%
9612     {\end{supertabular}}}%
9613 }

```

super4colborder The super4colborder style is like the super4col but with a border.

```

9614 \newglossarystyle{super4colborder}{%

Base it on the glostylesuper4col style:

9615   \setglossarystyle{super4col}%

Put the glossary in a supertabular environment with four columns and a horizontal line in the
head and tail:

9616   \renewenvironment{theglossary}%
9617     {\tablehead{\hline}\tabletail{\hline}%
9618     \begin{supertabular}{1111111}}%
9619     {\end{supertabular}}}%
9620 }

```

colheaderborder The super4colheaderborder style is like the super4col but with a header and border.

```

9621 \newglossarystyle{super4colheaderborder}{%

Base it on the glostylesuper4col style:

9622   \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:

9623   \renewenvironment{theglossary}%

```

```

9624     {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
9625         \bfseries\symbolname &
9626         \bfseries\pagelistname\tabularnewline\hline}%
9627     \tabletail{\hline}%
9628     \begin{supertabular}{|l|l|l|l|}%
9629     {\end{supertabular}}%
9630 }

```

altsuper4col The altsuper4col glossary style is like super4col but has provision for multiline descriptions.

```

9631 \newglossarystyle{altsuper4col}{%

```

Base it on the glostylessuper4col style:

```

9632 \setglossarystyle{super4col}%

```

Put the glossary in a supertabular environment with four columns and no head or tail:

```

9633 \renewenvironment{theglossary}%
9634     {\tablehead{}\tabletail{}}%
9635     \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}%
9636     {\end{supertabular}}%
9637 }

```

super4colheader The altsuper4colheader style is like the altsuper4col but with a header row.

```

9638 \newglossarystyle{altsuper4colheader}{%

```

Base it on the glostylessuper4colheader style:

```

9639 \setglossarystyle{super4colheader}%

```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```

9640 \renewenvironment{theglossary}%
9641     {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9642         \bfseries\symbolname &
9643         \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9644     \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}%
9645     {\end{supertabular}}%
9646 }

```

super4colborder The altsuper4colborder style is like the altsuper4col but with a border.

```

9647 \newglossarystyle{altsuper4colborder}{%

```

Base it on the glostylessuper4colborder style:

```

9648 \setglossarystyle{super4colborder}%

```

Put the glossary in a supertabular environment with four columns and a horizontal line in the head and tail:

```

9649 \renewenvironment{theglossary}%
9650     {\tablehead{\hline}\tabletail{\hline}%
9651     \begin{supertabular}%
9652         {|l|lp{\glsgdescwidth}|l|lp{\glspagelistwidth}|}%
9653     {\end{supertabular}}%
9654 }

```

`colheaderborder` The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```

9655 \newglossarystyle{altsuper4colheaderborder}{%
      Base it on the glostylessuper4colheaderborder style:
9656   \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:
9657   \renewenvironment{theglossary}%
9658     {\tablehead{\hline
9659       \bfseries\entryname &
9660       \bfseries\descriptionname &
9661       \bfseries\symbolname &
9662       \bfseries\pagelistname\tabularnewline\hline}%
9663     \tabletail{\hline}%
9664     \begin{supertabular}%
9665       {\l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}%
9666     {\end{supertabular}}%
9667 }
```

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.

```

9668 \ProvidesPackage{glossary-superragged}[2021/10/19 v4.48 (NLCT)]
```

Requires the package:

```

9669 \RequirePackage{array}
```

Requires the package:

```

9670 \RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```

9671 \@ifundefined{glsdescwidth}{%
9672   \newlength\glsdescwidth
9673   \setlength{\glsdescwidth}{0.6\hsize}
9674 }{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```

9675 \@ifundefined{glspagelistwidth}{%
9676   \newlength\glspagelistwidth
9677   \setlength{\glspagelistwidth}{0.1\hsize}
9678 }{}
```

superragged The superragged glossary style uses the supertabular environment.

```
9679 \newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
9680 \renewenvironment{theglossary}%
```

```
9681 {\tablehead{}\tabletail{}}%
```

```
9682 \begin{supertabular}{1>{\raggedright}p{\glstdescwidth}}%
```

```
9683 {\end{supertabular}}%
```

Do nothing at the start of the table:

```
9684 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9685 \renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
9686 \renewcommand{\glossentry}[2]{%
```

```
9687 \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
```

```
9688 \glossentrydesc{##1}\glspostdescription\space ##2%
```

```
9689 \tabularnewline
```

```
9690 }%
```

Sub entries put in a row (no name, description and page list in second column):

```
9691 \renewcommand{\subglossentry}[3]{%
```

```
9692 &
```

```
9693 \glssubentryitem{##2}%
```

```
9694 \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
```

```
9695 ##3%
```

```
9696 \tabularnewline
```

```
9697 }%
```

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>)

```
9698 \ifglsnogroupskip
```

```
9699 \renewcommand*{\glsgroupskip}{}%
```

```
9700 \else
```

```
9701 \renewcommand*{\glsgroupskip}{& \tabularnewline}%
```

```
9702 \fi
```

```
9703 }
```

perraggedborder The superraggedborder style is like the above, but with horizontal and vertical lines:

```
9704 \newglossarystyle{superraggedborder}{%
```

Base it on the `glostylessuperragged` style:

```
9705 \setglossarystyle{superragged}%
```

Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:

```
9706 \renewenvironment{theglossary}%
```

```
9707 {\tablehead{\hline}\tabletail{\hline}%
```

```
9708 \begin{supertabular}{1|1>{\raggedright}p{\glstdescwidth}|}}%
```



```

9709     {\end{supertabular}}}%
9710 }

```

superraggedheader The superraggedheader style is like the super style, but with a header:

```

9711 \newglossarystyle{superraggedheader}{%
    Base it on the glostylesuperragged style:
9712   \setglossarystyle{superragged}%
    Put the glossary in a supertabular environment with two columns, a header and no tail:
9713   \renewenvironment{theglossary}%
9714     {\tablehead{\bfseries \entryname & \bfseries \descriptionname
9715       \tabularnewline}%
9716     \tabletail{}}%
9717   \begin{supertabular}{l>{\raggedright}p{\glstdescwidth}}}%
9718   {\end{supertabular}}}%
9719 }

```

superraggedheaderborder The superraggedheaderborder style is like the superragged style but with a header and border:

```

9720 \newglossarystyle{superraggedheaderborder}{%
    Base it on the glostylesuper style:
9721   \setglossarystyle{superragged}%
    Put the glossary in a supertabular environment with two columns, a header and horizontal
    lines above and below the table:
9722   \renewenvironment{theglossary}%
9723     {\tablehead{\hline\bfseries \entryname &
9724       \bfseries \descriptionname\tabularnewline\hline}%
9725     \tabletail{\hline}
9726     \begin{supertabular}{ll>{\raggedright}p{\glstdescwidth}l}}}%
9727     {\end{supertabular}}}%
9728 }

```

superragged3col The superragged3col style is like the superragged style, but with 3 columns:

```

9729 \newglossarystyle{superragged3col}{%
    Put the glossary in a supertabular environment with three columns and no head or tail:
9730   \renewenvironment{theglossary}%
9731     {\tablehead{}\tabletail{}}%
9732     \begin{supertabular}{l>{\raggedright}p{\glstdescwidth}%
9733       >{\raggedright}p{\glspagelistwidth}}}%
9734     {\end{supertabular}}}%
    Do nothing at the start of the table:
9735   \renewcommand*{\glossaryheader}{}%
    No group headings:
9736   \renewcommand*{\glsgroupheading}[1]{}%

```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```

9737 \renewcommand{\glossentry}[2]{%
9738   \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9739   \glossentrydesc{##1} &
9740   ##2\tabularnewline
9741 }%
```

Sub entries on a row (no name, description in second column, page list in last column):

```

9742 \renewcommand{\subglossentry}[3]{%
9743   &
9744   \glssubentryitem{##2}%
9745   \glstarget{##2}{\strut}\glossentrydesc{##2} &
9746   ##3\tabularnewline
9747 }%
```

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>)

```

9748 \ifglsgroupskip
9749   \renewcommand*{\glsgroupskip}{}%
9750 \else
9751   \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9752 \fi
9753 }
```

ragged3colborder The superragged3colborder style is like the superragged3col style, but with a border:

```

9754 \newglossarystyle{superragged3colborder}{%
```

Base it on the glostylesuperragged3col style:

```

9755 \setglossarystyle{superragged3col}%
```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```

9756 \renewenvironment{theglossary}%
9757   {\tablehead{\hline}\tabletail{\hline}%
9758   \begin{supertabular}{|l|>\raggedright}p{\glsgdescwidth}|%
9759   >\raggedright}p{\glspagelistwidth}|}%
9760   {\end{supertabular}}%
9761 }
```

ragged3colheader The superragged3colheader style is like the superragged3col style but with a header row:

```

9762 \newglossarystyle{superragged3colheader}{%
```

Base it on the glostylesuperragged3col style:

```

9763 \setglossarystyle{superragged3col}%
```

Put the glossary in a supertabular environment with three columns, a header and no tail:

```

9764 \renewenvironment{theglossary}%
9765   {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9766   \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9767   \begin{supertabular}{l>\raggedright}p{\glsgdescwidth}%
```

```

9768         >\raggedright}p{\glspagelistwidth}}}%
9769     {\end{supertabular}}}%
9770 }

```

colheaderborder The superragged3colheaderborder style is like the superragged3col style but with a header and border:

```

9771 \newglossarystyle{superragged3colheaderborder}{%
    Base it on the glostylesuperragged3colborder style:
9772 \setglossarystyle{superragged3colborder}%
    Put the glossary in a supertabular environment with three columns, a header with horizontal
    lines and a horizontal line in the tail:
9773 \renewenvironment{theglossary}%
9774     {\tablehead{\hline
9775         \bfseries\entryname&\bfseries\descriptionname&
9776         \bfseries\pagelistname\tabularnewline\hline}%
9777     \tabletail{\hline}%
9778     \begin{supertabular}{|l|>\raggedright}p{\glsdescwidth}|%
9779         >\raggedright}p{\glspagelistwidth}|}%
9780     {\end{supertabular}}}%
9781 }

```

superragged4col The altsuperragged4col glossary style is like altsuper4col style in the package but uses ragged right formatting in the description and page list columns.

```

9782 \newglossarystyle{altsuperragged4col}{%
    Put the glossary in a supertabular environment with four columns and no head or tail:
9783 \renewenvironment{theglossary}%
9784     {\tablehead{}\tabletail{}%
9785         \begin{supertabular}{|l>\raggedright}p{\glsdescwidth}l%
9786         >\raggedright}p{\glspagelistwidth}}}%
9787     {\end{supertabular}}}%
    Do nothing at the start of the table:
9788 \renewcommand*{\glossaryheader}{}%

```

No group headings:

```

9789 \renewcommand*{\glsgroupheading}[1]{}%

```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```

9790 \renewcommand{\glossentry}[2]{%
9791     \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
9792     \glossentrydesc{##1} &
9793     \glossentrysymbol{##1} & ##2\tabularnewline
9794 }%

```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```

9795 \renewcommand{\subglossentry}[3]{%

```

```

9796      &
9797      \glssubentryitem{##2}%
9798      \glstarget{##2}{\strut}\glossentrydesc{##2} &
9799      \glossentrysymbol{##2} & ##3\tabularnewline
9800  }%

```

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>)

```

9801  \ifglsgroupskip
9802    \renewcommand*{\glsgroupskip}{}%
9803  \else
9804    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
9805  \fi
9806 }

```

agged4colheader The altsuperragged4colheader style is like the altsuperragged4col style but with a header row.

```

9807 \newglossarystyle{altsuperragged4colheader}{%

```

Base it on the glostylealtsuperragged4col style:

```

9808   \setglossarystyle{altsuperragged4col}%

```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```

9809   \renewenvironment{theglossary}%
9810     {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
9811               \bfseries\symbolname &
9812               \bfseries\pagelistname\tabularnewline}\tabletail{}}%
9813     \begin{supertabular}{l>{\raggedright}p{\glsgdescwidth}l%
9814       >{\raggedright}p{\glspagelistwidth}}}%
9815     {\end{supertabular}}}%
9816 }

```

agged4colborder The altsuperragged4colborder style is like the altsuperragged4col style but with a border.

```

9817 \newglossarystyle{altsuperragged4colborder}{%

```

Base it on the glostylealtsuperragged4col style:

```

9818   \setglossarystyle{altsuper4col}%

```

Put the glossary in a supertabular environment with four columns and a horizontal line in the head and tail:

```

9819   \renewenvironment{theglossary}%
9820     {\tablehead{\hline}\tabletail{\hline}%
9821     \begin{supertabular}%
9822       {\l|>{\raggedright}p{\glsgdescwidth}|l|}%
9823       >{\raggedright}p{\glspagelistwidth}|}}%
9824     {\end{supertabular}}}%
9825 }

```

colheaderborder The altsuperragged4colheaderborder style is like the altsuperragged4col style but with a header and border.

```

9826 \newglossarystyle{altsuperragged4colheaderborder}{%

```

Base it on the `glostylealtsuperragged4col` style:

```
9827 \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:

```
9828 \renewenvironment{theglossary}%
9829   {\tablehead{\hline
9830     \bfseries\entryname &
9831     \bfseries\descriptionname &
9832     \bfseries\symbolname &
9833     \bfseries\pagelistname\tabularnewline\hline}%
9834   \tabletail{\hline}%
9835   \begin{supertabular}%
9836     {||>\raggedright}p{\glstdescwidth}|||%
9837     >\raggedright}p{\glspagelistwidth}||}%
9838   {\end{supertabular}}%
9839 }
```

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.

```
9840 \ProvidesPackage{glossary-tree}[2021/10/19 v4.48 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
9841 \providecommand{\indexspace}{%
9842   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
9843 }
```

`\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.

```
9844 \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.

```
9845 \newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```
9846 \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.

```

9847 \ifdef\@idxitem
9848 {\newcommand{\glstreeitem}{\@idxitem}}
9849 {\newcommand{\glstreeitem}{\par\hangindent40\p@}}

```

`\glstreesubitem` Level 1 item used in index style.

```

9850 \ifdef\subitem
9851 {\let\glstreesubitem\subitem}
9852 {\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}

```

`\glstreesubsubitem` Level 1 item used in index style.

```

9853 \ifdef\subsubitem
9854 {\let\glstreesubsubitem\subsubitem}
9855 {\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}

```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `almtree` style).

```

9856 \newcommand{\glstreepredesc}{\space}

```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treename` and `almtree` style).

```

9857 \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.

```

9858 \newglossarystyle{index}{%

```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```

9859 \renewenvironment{theglossary}%
9860 {\setlength{\parindent}{0pt}%
9861 \setlength{\parskip}{0pt plus 0.3pt}%
9862 \let\item\glstreeitem
9863 \let\subitem\glstreesubitem
9864 \let\subsubitem\glstreesubsubitem
9865 }%

```

```

9866 {\par}%

```

Do nothing at the start of the environment:

```

9867 \renewcommand*{\glossaryheader}{}%

```

No group headers:

```

9868 \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.

```

9869 \renewcommand*{\glossentry}[2]{%

```

```

9870 \item\glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9871 \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9872 \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
9873 }%

```

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.

```

9874 \renewcommand{\subglossentry}[3]{%
9875 \ifcase##1\relax
9876 % level 0
9877 \item
9878 \or
9879 % level 1
9880 \subitem
9881 \glssubentryitem{##2}%
9882 \else
9883 % all other levels
9884 \subsubitem
9885 \fi
9886 \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
9887 \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
9888 \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
9889 }%

```

Vertical gap between groups is the same as that used by indices:

```

9890 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}

```

indexgroup The `indexgroup` style is like the `index` style but has headings.

```

9891 \newglossarystyle{indexgroup}{%

```

Base it on the `glostyleindex` style:

```

9892 \setglossarystyle{index}%

```

Add a heading for each group. This puts the group's title in bold followed by a vertical gap.

```

9893 \renewcommand*{\glsgroupheading}[1]{%
9894 \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
9895 \indexspace
9896 }%
9897 }

```

indexhypergroup The `indexhypergroup` style is like the `indexgroup` style but has hyper navigation.

```

9898 \newglossarystyle{indexhypergroup}{%

```

Base it on the `glostyleindex` style:

```

9899 \setglossarystyle{index}%

```

Put navigation links to the groups at the start of the glossary:

```

9900 \renewcommand*{\glossaryheader}{%
9901 \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.

```

9902 \renewcommand*\glsgroupheading}[1]{%
9903   \item\glstreegroupheaderfmt
9904     {\glshnavhypertarget{##1}{\glsgrouptitle{##1}}}%
9905   \indexspace}%
9906 }
```

tree The tree glossary style is similar in style to the index style, but can have arbitrary levels.

```

9907 \newglossarystyle{tree}{%
```

Set the paragraph indentation and skip:

```

9908 \renewenvironment{theglossary}%
9909   {\setlength{\parindent}{0pt}%
9910    \setlength{\parskip}{0pt plus 0.3pt}}%
9911   {}%
```

Do nothing at the start of the theglossary environment:

```

9912 \renewcommand*\glossaryheader{}%
```

No group headings:

```

9913 \renewcommand*\glsgroupheading}[1]{}%
```

Main (level 0) entries: name in bold, followed by symbol in brackets (if it exists), the description and the page list:

```

9914 \renewcommand{\glossentry}[2]{%
9915   \hangindent0pt\relax
9916   \parindent0pt\relax
9917   \glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9918   \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9919   \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
9920   }%
```

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.

```

9921 \renewcommand{\subglossentry}[3]{%
9922   \hangindent##1\glstreeindent\relax
9923   \parindent##1\glstreeindent\relax
9924   \ifnum##1=1\relax
9925     \glssubentryitem{##2}%
9926     \fi
9927   \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
9928   \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
9929   \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
9930   }%
```

Vertical gap between groups is the same as that used by indices:

```

9931 \renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}
```

treegroup Like the tree style but the glossary groups have headings.

```

9932 \newglossarystyle{treegroup}{%
```


Base it on the glostyletree style:

```
9933 \setglossarystyle{tree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
9934 \renewcommand{\glsgroupheading}[1]{\par
9935 \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
9936 \indexspace}%
9937 }
```

treehypergroup The treehypergroup style is like the treegroup style, but has a set of links to the groups at the start of the glossary.

```
9938 \newglossarystyle{treehypergroup}{%
```

Base it on the glostyletree style:

```
9939 \setglossarystyle{tree}%
```

Put navigation links to the groups at the start of the theglossary environment:

```
9940 \renewcommand*{\glossaryheader}{%
9941 \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
9942 \renewcommand*{\glsgroupheading}[1]{%
9943 \par\noindent
9944 \glstreegroupheaderfmt
9945 {\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
9946 \indexspace}%
9947 }
```

\glstreeindent Length governing left indent for each level of the tree style.

```
9948 \newlength\glstreeindent
9949 \setlength{\glstreeindent}{10pt}
```

treenoname The treenoname glossary style is like the tree style, but doesn't print the name or symbol for sub-levels.

```
9950 \newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
9951 \renewenvironment{theglossary}%
9952 {\setlength{\parindent}{0pt}%
9953 \setlength{\parskip}{0pt plus 0.3pt}}%
9954 {}%
```

No header:

```
9955 \renewcommand*{\glossaryheader}{}%
```

No group headings:

```
9956 \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.

```
9957 \renewcommand{\glossentry}[2]{%
```

```

9958 \hangindentOpt\relax
9959 \parindentOpt\relax
9960 \glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
9961 \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
9962 \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
9963 }%

```

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.

```

9964 \renewcommand{\subglossentry}[3]{%
9965 \hangindent##1\glstreeindent\relax
9966 \parindent##1\glstreeindent\relax
9967 \ifnum##1=1\relax
9968 \glssubentryitem{##2}%
9969 \fi
9970 \glstarget{##2}{\strut}%
9971 \glossentrydesc{##2}\glspostdescription\space##3\par
9972 }%

```

Vertical gap between groups is the same as that used by indices:

```

9973 \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
9974 }

```

treenonamegroup Like the `treenoname` style but the glossary groups have headings.

```

9975 \newglossarystyle{treenonamegroup}{%

```

Base it on the `glostytreetreenoname` style:

```

9976 \setglossarystyle{treenoname}%

```

Give each group a heading:

```

9977 \renewcommand{\glsgroupheading}[1]{\par
9978 \noindent\glstreegroupheaderfmt
9979 {\glsggetgrouptitle{##1}}\par\indexspace}%
9980 }

```

onamehypergroup The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```

9981 \newglossarystyle{treenonamehypergroup}{%

```

Base it on the `glostytreetreenoname` style:

```

9982 \setglossarystyle{treenoname}%

```

Put navigation links to the groups at the start of the `theglossary` environment:

```

9983 \renewcommand*{\glossaryheader}{%
9984 \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```

9985 \renewcommand*{\glsgroupheading}[1]{%
9986 \par\noindent
9987 \glstreegroupheaderfmt
9988 {\glsnavhypertarget{##1}{\glsggetgrouptitle{##1}}}\par
9989 \indexspace}%
9990 }

```

`esttoplevelname` Find the widest name over all parentless entries in the given glossary or glossaries.

```

9991 \newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
9992   \dimen@=0pt\relax
9993   \gls@tmplen=0pt\relax
9994   \forallglossaries[#1]{\@gls@type}%
9995   {%
9996     \forallglsentries[\@gls@type]{\@glo@label}%
9997     {%
9998       \ifglsahasparent{\@glo@label}%
9999       }%
10000     {%
10001       \settowidth{\dimen@}%
10002         {\glstreenamfmt{\glsentryname{\@glo@label}}}%
10003       \ifdim\dimen@>\gls@tmplen
10004         \gls@tmplen=\dimen@
10005       \letcs{\@glswidestname}{glo\glsdetoklabel{\@glo@label}@name}%
10006       \fi
10007     }%
10008   }%
10009 }%
10010 }

```

`\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.

```

10011 \newcommand*{\glssetwidest}[2][0]{%
10012   \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
10013     #2}%
10014 }

```

`\@glswidestname` Initialise `\@glswidestname`.

```

10015 \newcommand*{\@glswidestname}{}

```

`\glstreenamebox` Used by the alttree style to create the box for the name and associated information.

```

10016 \newcommand*{\glstreenamebox}[2]{%
10017   \makebox[#1][l]{#2}%
10018 }

```

`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`.

```

10019 \newglossarystyle{alttree}{%

```

Redefine theglossary environment.

```

10020 \renewenvironment{theglossary}%
10021   {\def\@gls@prevlevel{-1}%
10022    \mbox{}\par}%
10023   {\par}%

```

Set the header and group headers to nothing.

```

10024 \renewcommand*{\glossaryheader}{}%
10025 \renewcommand*{\glsgroupheading}[1]{}%

```

Redefine the way that the level 0 entries are displayed.

```
10026 \renewcommand{\glossentry}[2]{%
10027   \ifnum\@gls@prevlevel=0\relax
10028   \else
```

Find out how big the indentation should be by measuring the widest entry.

```
10029     \settowidth{\glstreeindent}{\glstreenamfmt{\@glswidestname\space}}%
10030   \fi
```

Set the hangindent and paragraph indent.

```
10031   \hangindent\glstreeindent
10032   \parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
10033   \makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%
10034     \glssentryitem{##1}\glstreenamfmt{\glstarget{##1}{\glossentryname{##1}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
10035   \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
10036   \glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
10037   \def\@gls@prevlevel{0}%
10038 }%
```

Redefine the way sub-entries are displayed.

```
10039 \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.

```
10040   \ifnum##1=1\relax
10041     \glssubentryitem{##2}%
10042   \fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
10043   \ifnum\@gls@prevlevel=##1\relax
10044   \else
```

Compute the widest entry for this level, or for level 0 if not defined for this level. Store in `\gls@tmplen`

```
10045     \@ifundefined{@glswidestname\romannumeral##1}{%
10046       \settowidth{\gls@tmplen}{\glstreenamfmt{\@glswidestname\space}}}%
10047     \settowidth{\gls@tmplen}{\glstreenamfmt{%
10048       \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
10049   \ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
10050      \setlength\glstreeindent\gls@tmplen
10051      \addtolength\glstreeindent\parindent
10052      \parindent\glstreeindent
10053      \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`.

```
10054      \ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
10055      \settowidth{\glstreeindent}{\glstreenamfmt{%
10056      \@glswidestname\space}}}{%
10057      \settowidth{\glstreeindent}{\glstreenamfmt{%
10058      \csname @glswidestname\romannumeral\@gls@prevlevel
10059      \endcsname\space}}}{%
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```
10060      \addtolength\parindent{-\glstreeindent}%
10061      \setlength\glstreeindent\parindent
10062      \fi
10063      \fi
```

Set the hanging indentation.

```
10064      \hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```
10065      \makebox[0pt][r]{\glstreenambox{\gls@tmplen}{%
10066      \glstreenamfmt{\glstarget{##2}{\glossentryname{##2}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
10067      \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
```

Do the description followed by the description terminator and location list.

```
10068      \glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```
10069      \def\@gls@prevlevel{##1}%
10070      }%
```

Vertical gap between groups is the same as that used by indices:

```
10071      \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
10072 }
```

almtreegroup Like the `almtree` style but the glossary groups have headings.

```
10073 \newglossarystyle{almtreegroup}{%
```

Base it on the `glostylealmtree` style:

```
10074      \setglossarystyle{almtree}%
```

Give each group a heading.

```
10075      \renewcommand{\glsgroupheading}[1]{\par
10076      \def\@gls@prevlevel{-1}%
10077      \hangindent0pt\relax
```

```

10078     \parindent0pt\relax
10079     \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}}%
10080     \par\indexspace}%
10081 }

```

alttreehypergroup The alttreehypergroup style is like the alttreegroup style, but has a set of links to the groups at the start of the glossary.

```

10082 \newglossarystyle{alttreehypergroup}{%
    Base it on the glostylealttree style:
10083     \setglossarystyle{alttree}%
    Put the navigation links in the header
10084     \renewcommand*{\glossaryheader}{%
10085         \par
10086         \def\@gls@prevlevel{-1}%
10087         \hangindent0pt\relax
10088         \parindent0pt\relax
10089         \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    Put a hypertarget at the start of each group
10090     \renewcommand*{\glsgroupheading}[1]{%
10091         \par
10092         \def\@gls@prevlevel{-1}%
10093         \hangindent0pt\relax
10094         \parindent0pt\relax
10095         \glstreegroupheaderfmt
10096         {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
10097         \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.

```
10098 \NeedsTeXFormat{LaTeX2e}
10099 \ProvidesPackage{glossaries-compatible-207}[2021/10/19 v4.48 (NLCT)]
```

AddXdyAttribute Adds an attribute in old format.

```
10100 \ifglsxindy
10101   \renewcommand*\GlsAddXdyAttribute[1]{%
10102     \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
10103     \expandafter\toks@\expandafter{\@xdylocref}%
10104     \edef\@xdylocref{\the\toks@ ^^J}%
10105     (markup-locref
10106     :open \string"\string~n\string\setentrycounter
10107       {\noexpand\glscounter}%
10108       \expandafter\string\csname#1\endcsname
10109       \expandafter\@gobble\string\{\string" ^^J
10110       :close \string"\expandafter\@gobble\string\}\string" ^^J
10111       :attr \string"#1\string"))}}
```

Only has an effect before `\writeist`:

```
10112 \fi
```

sAddXdyCounters

```
10113 \renewcommand*\GlsAddXdyCounters[1]{%
10114   \GlossariesWarning{\string\GlsAddXdyCounters\space not available
10115     in compatibility mode.}%
10116 }
```

Add predefined attributes

```
10117 \GlsAddXdyAttribute{glsnumberformat}
10118 \GlsAddXdyAttribute{textrm}
10119 \GlsAddXdyAttribute{textsf}
10120 \GlsAddXdyAttribute{texttt}
10121 \GlsAddXdyAttribute{textbf}
10122 \GlsAddXdyAttribute{textmd}
10123 \GlsAddXdyAttribute{textit}
10124 \GlsAddXdyAttribute{textup}
10125 \GlsAddXdyAttribute{textsl}
```

```

10126 \GlsAddXdyAttribute{textsc}
10127 \GlsAddXdyAttribute{emph}
10128 \GlsAddXdyAttribute{glshypernumber}
10129 \GlsAddXdyAttribute{hyperrm}
10130 \GlsAddXdyAttribute{hypersf}
10131 \GlsAddXdyAttribute{hypertt}
10132 \GlsAddXdyAttribute{hyperbf}
10133 \GlsAddXdyAttribute{hypermd}
10134 \GlsAddXdyAttribute{hyperit}
10135 \GlsAddXdyAttribute{hyperup}
10136 \GlsAddXdyAttribute{hypersl}
10137 \GlsAddXdyAttribute{hypersc}
10138 \GlsAddXdyAttribute{hyperemph}

```

sAddXdyLocation Restore v2.07 definition:

```

10139 \ifglxindy
10140 \renewcommand*{\GlsAddXdyLocation}[2]{%
10141 \edef\@xdyuserlocationdefs{%
10142 \@xdyuserlocationdefs ^^J%
10143 (define-location-class \string"#1\string"^^J\space\space
10144 \space(#2))
10145 }%
10146 \edef\@xdyuserlocationnames{%
10147 \@xdyuserlocationnames^^J\space\space\space
10148 \string"#1\string"}%
10149 }
10150 \fi

```

\@do@wrglossary

```

10151 \renewcommand{\@do@wrglossary}[1]{%
  Determine whether to use xindy or makeindex syntax
10152 \ifglxindy
  Need to determine if the formatting information starts with a ( or ) indicating a range.
10153 \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
10154 \def\@glo@range{}%
10155 \expandafter\if\@glo@prefix(\relax
10156 \def\@glo@range{:open-range}%
10157 \else
10158 \expandafter\if\@glo@prefix)\relax
10159 \def\@glo@range{:close-range}%
10160 \fi
10161 \fi

  Get the location and escape any special characters
10162 \protected@edef\@glslocref{\theglentrycounter}%
10163 \@gls@checkmkidxchars\@glslocref

  Write to the glossary file using xindy syntax.
10164 \glossary[\csname glo@#1@type\endcsname]{%

```



```

10165 (indexentry :tkey (\csname glo@#1@index\endcsname)
10166 :locoref \string"\@glslocoref\string" %
10167 :attr \string"\@glo@suffix\string" \@glo@range
10168 )
10169 }%
10170 \else
    Convert the format information into the format required for makeindex
10171 \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
    Write to the glossary file using makeindex syntax.
10172 \glossary[\csname glo@#1@type\endcsname]{%
10173 \string\glossaryentry{\csname glo@#1@index\endcsname
10174 \@gls@encapchar\@glo@numfmt}{\theglsentrycounter}}%
10175 \fi
10176 }

```

t@glo@numformat Only had 3 arguments in v2.07

```

10177 \def\@set@glo@numformat#1#2#3{%
10178 \expandafter\@glo@check@mkidxrangechar#3\@nil
10179 \protected@edef#1{%
10180 \@glo@prefix setentrycounter[] {#2}%
10181 \expandafter\string\csname\@glo@suffix\endcsname
10182 }%
10183 \@gls@checkmkidxchars#1%
10184 }

```

\writeist Redefine \writeist back to the way it was in v2.07, but change \istfile to \glswrite.

```

10185 \ifglxindy
10186 \def\writeist{%
10187 \openout\glswrite=\istfilename
10188 \write\glswrite{;; xindy style file created by the glossaries
10189 package in compatible-2.07 mode}%
10190 \write\glswrite{;; for document '\jobname' on
10191 \the\year-\the\month-\the\day}%
10192 \write\glswrite{^^J; required styles^^J}
10193 \@for\@xdystyle:=\@xdyrequiredstyles\do{%
10194 \ifx\@xdystyle\@empty
10195 \else
10196 \protected@write\glswrite{{(require
10197 \string"\@xdystyle.xdy\string")}}%
10198 \fi
10199 }%
10200 \write\glswrite{^^J%
10201 ; list of allowed attributes (number formats)^^J}%
10202 \write\glswrite{(define-attributes ((\@xdyattributes)))}%
10203 \write\glswrite{^^J; user defined alphabets^^J}%
10204 \write\glswrite{\@xdyuseralphabets}%
10205 \write\glswrite{^^J; location class definitions^^J}%
10206 \protected@edef\@gls@roman{\@roman{0}\string"

```

```

10207     \string"roman-numbers-lowercase\string" :sep \string"}}%
10208 \@onelevel@sanitize\@gls@roman
10209 \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
10210     :sep \string"}}%
10211 \@onelevel@sanitize\@tmp
10212 \ifx\@tmp\@gls@roman
10213     \write\glswrite{(define-location-class
10214         \string"roman-page-numbers\string"^^J\space\space\space
10215         (\string"roman-numbers-lowercase\string")
10216         :min-range-length \@glsminrange)}}%
10217 \else
10218     \write\glswrite{(define-location-class
10219         \string"roman-page-numbers\string"^^J\space\space\space
10220         (:sep "\@gls@roman")
10221         :min-range-length \@glsminrange)}}%
10222 \fi
10223 \write\glswrite{(define-location-class
10224     \string"Roman-page-numbers\string"^^J\space\space\space
10225     (\string"roman-numbers-uppercase\string")
10226     :min-range-length \@glsminrange)}}%
10227 \write\glswrite{(define-location-class
10228     \string"arabic-page-numbers\string"^^J\space\space\space
10229     (\string"arabic-numbers\string")
10230     :min-range-length \@glsminrange)}}%
10231 \write\glswrite{(define-location-class
10232     \string"alpha-page-numbers\string"^^J\space\space\space
10233     (\string"alpha\string")
10234     :min-range-length \@glsminrange)}}%
10235 \write\glswrite{(define-location-class
10236     \string"Alpha-page-numbers\string"^^J\space\space\space
10237     (\string"ALPHA\string")
10238     :min-range-length \@glsminrange)}}%
10239 \write\glswrite{(define-location-class
10240     \string"Appendix-page-numbers\string"^^J\space\space\space
10241     (\string"ALPHA\string"
10242     :sep \string"\@glsAlphacompositor\string"
10243     \string"arabic-numbers\string")
10244     :min-range-length \@glsminrange)}}%
10245 \write\glswrite{(define-location-class
10246     \string"arabic-section-numbers\string"^^J\space\space\space
10247     (\string"arabic-numbers\string"
10248     :sep \string"\glscompositor\string"
10249     \string"arabic-numbers\string")
10250     :min-range-length \@glsminrange)}}%
10251 \write\glswrite{^^J; user defined location classes}%
10252 \write\glswrite{\@xdyuserlocationdefs}%
10253 \write\glswrite{^^J; define cross-reference class^^J}%
10254 \write\glswrite{(define-crossref-class \string"see\string"
10255     :unverified )}%

```

```

10256 \write\glswrite{(markup-crossref-list
10257 :class \string"see\string"^^J\space\space\space
10258 :open \string"\string\glsseeformat\string"
10259 :close \string"{}\string")}%
10260 \write\glswrite{^^J; define the order of the location classes}%
10261 \write\glswrite{(define-location-class-order
10262 (\@xdylocationclassorder))}%
10263 \write\glswrite{^^J; define the glossary markup^^J}%
10264 \write\glswrite{(markup-index^^J\space\space\space
10265 :open \string"\string
10266 \glossarysection[\string\glossarytoctitle]{\string
10267 \glossarytitle}\string\glossarypreamble\string~n\string\begin
10268 {theglossary}\string\glossaryheader\string~n\string" ^^J\space
10269 \space\space:close \string"\expandafter\@gobble
10270 \string%\string~n\string
10271 \end{theglossary}\string\glossarypostamble
10272 \string~n\string" ^^J\space\space\space
10273 :tree)}}%
10274 \write\glswrite{(markup-letter-group-list
10275 :sep \string"\string\glsgroupskip\string~n\string")}%
10276 \write\glswrite{(markup-indexentry
10277 :open \string"\string\relax \string\glsresetentrylist
10278 \string~n\string")}%
10279 \write\glswrite{(markup-locclass-list :open
10280 \string"\glsopenbrace\string\glossaryentrynumbers
10281 \glsopenbrace\string\relax\space \string"^^J\space\space\space
10282 :sep \string", \string"
10283 :close \string"\glsclosebrace\glsclosebrace\string")}%
10284 \write\glswrite{(markup-locref-list
10285 :sep \string"\string\delimN\space\string")}%
10286 \write\glswrite{(markup-range
10287 :sep \string"\string\delimR\space\string")}%
10288 \@onelevel@sanitize\gls@suffixF
10289 \@onelevel@sanitize\gls@suffixFF
10290 \ifx\gls@suffixF\@empty
10291 \else
10292 \write\glswrite{(markup-range
10293 :close "\gls@suffixF" :length 1 :ignore-end)}%
10294 \fi
10295 \ifx\gls@suffixFF\@empty
10296 \else
10297 \write\glswrite{(markup-range
10298 :close "\gls@suffixFF" :length 2 :ignore-end)}%
10299 \fi
10300 \write\glswrite{^^J; define format to use for locations^^J}%
10301 \write\glswrite{\@xdylocref}%
10302 \write\glswrite{^^J; define letter group list format^^J}%
10303 \write\glswrite{(markup-letter-group-list
10304 :sep \string"\string\glsgroupskip\string~n\string")}%

```

```

10305 \write\glswrite{^^J; letter group headings^^J}%
10306 \write\glswrite{(markup-letter-group
10307   :open-head \string"\string\glsgroupheading
10308   \glsoopenbrace\string"^^J\space\space\space
10309   :close-head \string"\glsclosebrace\string")}%
10310 \write\glswrite{^^J; additional letter groups^^J}%
10311 \write\glswrite{\@xdylettergroups}%
10312 \write\glswrite{^^J; additional sort rules^^J}
10313 \write\glswrite{\@xdysortrules}%
10314 \noist}
10315 \else
10316 \edef\@gls@actualchar{\string?}
10317 \edef\@gls@encapchar{\string|}
10318 \edef\@gls@levelchar{\string!}
10319 \edef\@gls@quotechar{\string"}
10320 \def\writeist{\relax
10321   \openout\glswrite=\istfilename
10322   \write\glswrite{\expandafter\@gobble\string\% makeindex style file
10323     created by the glossaries package}
10324   \write\glswrite{\expandafter\@gobble\string\% for document
10325     '\jobname' on \the\year-\the\month-\the\day}
10326   \write\glswrite{actual '\@gls@actualchar'}
10327   \write\glswrite{encap '\@gls@encapchar'}
10328   \write\glswrite{level '\@gls@levelchar'}
10329   \write\glswrite{quote '\@gls@quotechar'}
10330   \write\glswrite{keyword \string"\string\glossaryentry\string"}
10331   \write\glswrite{preamble \string"\string\glossarysection[\string
10332     \glossarytoctitle]{\string\glossarytitle}\string
10333     \glossarypreamble\string\n\string\begin{theglossary}\string
10334     \glossaryheader\string\n\string"}
10335   \write\glswrite{postamble \string"\string%\string\n\string
10336     \end{theglossary}\string\glossarypostamble\string\n
10337     \string"}
10338   \write\glswrite{group_skip \string"\string\glsgroupskip\string\n
10339     \string"}
10340   \write\glswrite{item_0 \string"\string%\string\n\string"}
10341   \write\glswrite{item_1 \string"\string%\string\n\string"}
10342   \write\glswrite{item_2 \string"\string%\string\n\string"}
10343   \write\glswrite{item_01 \string"\string%\string\n\string"}
10344   \write\glswrite{item_x1
10345     \string"\string\relax \string\glsresetentrylist\string\n
10346     \string"}
10347   \write\glswrite{item_12 \string"\string%\string\n\string"}
10348   \write\glswrite{item_x2
10349     \string"\string\relax \string\glsresetentrylist\string\n
10350     \string"}
10351   \write\glswrite{delim_0 \string"\string\{\string
10352     \glossaryentrynumbers\string\{\string\relax \string"}
10353   \write\glswrite{delim_1 \string"\string\{\string

```

```

10354      \glossaryentrynumbers\string\{\string\relax \string"}
10355      \write\glswrite{delim_2 \string"\string\{\string
10356      \glossaryentrynumbers\string\{\string\relax \string"}
10357      \write\glswrite{delim_t \string"\string\}\string\}\string"}
10358      \write\glswrite{delim_n \string"\string\delimN \string"}
10359      \write\glswrite{delim_r \string"\string\delimR \string"}
10360      \write\glswrite{headings_flag 1}
10361      \write\glswrite{heading_prefix
10362      \string"\string\glsgroupheading\string\{\string"}
10363      \write\glswrite{heading_suffix
10364      \string"\string\}\string\relax
10365      \string\glsgroupresetentrylist \string"}
10366      \write\glswrite{symhead_positive \string"glssymbols\string"}
10367      \write\glswrite{numhead_positive \string"glsglossarynumbers\string"}
10368      \write\glswrite{page_compositor \string"glsglossarycompositor\string"}
10369      \@gls@escbsdq\gls@suffixF
10370      \@gls@escbsdq\gls@suffixFF
10371      \ifx\gls@suffixF\@empty
10372      \else
10373      \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
10374      \fi
10375      \ifx\gls@suffixFF\@empty
10376      \else
10377      \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
10378      \fi
10379      \noist
10380  }
10381 \fi

```

\noist

```
10382 \renewcommand*{\noist}{\let\writeist\relax}
```

4.2 glossaries-compatible-307

```

10383 \NeedsTeXFormat{LaTeX2e}
10384 \ProvidesPackage{glossaries-compatible-307}[2021/10/19 v4.48 (NLCT)]

```

Compatibility macros for predefined glossary styles:

`\atglossarystyle` Defines a compatibility glossary style.

```

10385 \newcommand{\compatglossarystyle}[2]{%
10386   \ifcsundef{@glscompstyle@#1}%
10387   {%
10388     \csdef{@glscompstyle@#1}{#2}%
10389   }%
10390   {%
10391     \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
10392   }%
10393 }

```

Backward compatible inline style.

```

10394 \compatglossarystyle{inline}{%
10395   \renewcommand{\glossaryentryfield}[5]{%
10396     \glsinlinedopostchild
10397     \gls@inlinesep
10398     \def\glo@desc{##3}%
10399     \def\@no@post@desc{\nopostdesc}%
10400     \glstentryitem{##1}\glsinlinenameformat{##1}{##2}%
10401     \ifx\glo@desc\@no@post@desc
10402       \glsinlineemptydescformat{##4}{##5}%
10403     \else
10404       \ifstrepty{##3}%
10405         {\glsinlineemptydescformat{##4}{##5}}%
10406         {\glsinlinedescformat{##3}{##4}{##5}}%
10407     \fi
10408     \ifglshaschildren{##1}%
10409     {%
10410       \glsresetsubentrycounter
10411       \glsinlineparentchildseparator
10412       \def\gls@inlinesubsep{}%
10413       \def\gls@inlinepostchild{\glsinlinepostchild}%
10414     }%
10415   }%
10416   \def\gls@inlinesep{\glsinlineseparator}%
10417 }%

```

Sub-entries display description:

```

10418 \renewcommand{\glossarysubentryfield}[6]{%
10419   \gls@inlinesubsep%
10420   \glsinlinesubnameformat{##2}{##3}%
10421   \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
10422   \def\gls@inlinesubsep{\glsinlinesubseparator}%
10423 }%
10424 }

```

Backward compatible list style.

```

10425 \compatglossarystyle{list}{%
10426   \renewcommand*{\glossaryentryfield}[5]{%
10427     \item[\glstentryitem{##1}\glstarget{##1}{##2}]
10428     ##3\glspostdescription\space ##5}%

```

Sub-entries continue on the same line:

```

10429   \renewcommand*{\glossarysubentryfield}[6]{%
10430     \glssubentryitem{##2}%
10431     \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
10432 }

```

Backward compatible listgroup style.

```

10433 \compatglossarystyle{listgroup}{%
10434   \csuse{@glscompstyle@list}%
10435 }%

```

Backward compatible listhypergroup style.

```
10436 \compatglossarystyle{listhypergroup}{%
10437   \csuse{@glscompstyle@list}%
10438 }%
```

Backward compatible altlist style.

```
10439 \compatglossarystyle{altlist}{%
10440   \renewcommand*{\glossaryentryfield}[5]{%
10441     \item[\glstentryitem{##1}\glstarget{##1}{##2}]%
10442     \mbox{}\par\nobreak\@afterheading
10443     ##3\glspostdescription\space ##5}%
10444   \renewcommand{\glossarysubentryfield}[6]{%
10445     \par
10446     \glssubentryitem{##2}%
10447     \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
10448 }%
```

Backward compatible altlistgroup style.

```
10449 \compatglossarystyle{altlistgroup}{%
10450   \csuse{@glscompstyle@altlist}%
10451 }%
```

Backward compatible altlisthypergroup style.

```
10452 \compatglossarystyle{altlisthypergroup}{%
10453   \csuse{@glscompstyle@altlist}%
10454 }%
```

Backward compatible listdotted style.

```
10455 \compatglossarystyle{listdotted}{%
10456   \renewcommand*{\glossaryentryfield}[5]{%
10457     \item[\makebox[\glslistdottedwidth][l]{%
10458       \glstentryitem{##1}\glstarget{##1}{##2}%
10459       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
10460   \renewcommand*{\glossarysubentryfield}[6]{%
10461     \item[\makebox[\glslistdottedwidth][l]{%
10462       \glssubentryitem{##2}%
10463       \glstarget{##2}{##3}%
10464       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
10465 }%
```

Backward compatible sublistdotted style.

```
10466 \compatglossarystyle{sublistdotted}{%
10467   \csuse{@glscompstyle@listdotted}%
10468   \renewcommand*{\glossaryentryfield}[5]{%
10469     \item[\glstentryitem{##1}\glstarget{##1}{##2}]}%
10470 }%
```

Backward compatible long style.

```
10471 \compatglossarystyle{long}{%
10472   \renewcommand*{\glossaryentryfield}[5]{%
10473     \glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
10474   \renewcommand*{\glossarysubentryfield}[6]{%
10475     \glssubentryitem{##2}\glstarget{##2}{##3} & ##4\glspostdescription\space ##5\\}%
10476 }%
```

```

10475      &
10476      \glssubentryitem{##2}%
10477      \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
10478 }%

```

Backward compatible longborder style.

```

10479 \compatglossarystyle{longborder}{%
10480   \csuse{@glscmpstyle@long}%
10481 }%

```

Backward compatible longheader style.

```

10482 \compatglossarystyle{longheader}{%
10483   \csuse{@glscmpstyle@long}%
10484 }%

```

Backward compatible longheaderborder style.

```

10485 \compatglossarystyle{longheaderborder}{%
10486   \csuse{@glscmpstyle@long}%
10487 }%

```

Backward compatible long3col style.

```

10488 \compatglossarystyle{long3col}{%
10489   \renewcommand*{\glossaryentryfield}[5]{%
10490     \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##5\\}%
10491     \renewcommand*{\glossarysubentryfield}[6]{%
10492       &
10493       \glssubentryitem{##2}%
10494       \glstarget{##2}{\strut}##4 & ##6\\}%
10495   }%

```

Backward compatible long3colborder style.

```

10496 \compatglossarystyle{long3colborder}{%
10497   \csuse{@glscmpstyle@long3col}%
10498 }%

```

Backward compatible long3colheader style.

```

10499 \compatglossarystyle{long3colheader}{%
10500   \csuse{@glscmpstyle@long3col}%
10501 }%

```

Backward compatible long3colheaderborder style.

```

10502 \compatglossarystyle{long3colheaderborder}{%
10503   \csuse{@glscmpstyle@long3col}%
10504 }%

```

Backward compatible long4col style.

```

10505 \compatglossarystyle{long4col}{%
10506   \renewcommand*{\glossaryentryfield}[5]{%
10507     \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
10508     \renewcommand*{\glossarysubentryfield}[6]{%
10509       &
10510       \glssubentryitem{##2}%

```


10511 \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
10512 }%

Backward compatible long4colheader style.

10513 \compatglossarystyle{long4colheader}{%
10514 \csuse{@glscompstyle@long4col}%
10515 }%

Backward compatible long4colborder style.

10516 \compatglossarystyle{long4colborder}{%
10517 \csuse{@glscompstyle@long4col}%
10518 }%

Backward compatible long4colheaderborder style.

10519 \compatglossarystyle{long4colheaderborder}{%
10520 \csuse{@glscompstyle@long4col}%
10521 }%

Backward compatible altlong4col style.

10522 \compatglossarystyle{altlong4col}{%
10523 \csuse{@glscompstyle@long4col}%
10524 }%

Backward compatible altlong4colheader style.

10525 \compatglossarystyle{altlong4colheader}{%
10526 \csuse{@glscompstyle@long4col}%
10527 }%

Backward compatible altlong4colborder style.

10528 \compatglossarystyle{altlong4colborder}{%
10529 \csuse{@glscompstyle@long4col}%
10530 }%

Backward compatible altlong4colheaderborder style.

10531 \compatglossarystyle{altlong4colheaderborder}{%
10532 \csuse{@glscompstyle@long4col}%
10533 }%

Backward compatible long style.

10534 \compatglossarystyle{longragged}{%
10535 \renewcommand*{\glossaryentryfield}[5]{%
10536 \glssentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
10537 \tabularnewline}%
10538 \renewcommand*{\glossarysubentryfield}[6]{%
10539 &
10540 \glssubentryitem{##2}%
10541 \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
10542 \tabularnewline}%
10543 }%

Backward compatible longraggedborder style.

10544 \compatglossarystyle{longraggedborder}{%
10545 \csuse{@glscompstyle@longragged}%
10546 }%

Backward compatible longraggedheader style.

```
10547 \compatglossarystyle{longraggedheader}{%  
10548 \csuse{@glscompstyle@longragged}%  
10549 }%
```

Backward compatible longraggedheaderborder style.

```
10550 \compatglossarystyle{longraggedheaderborder}{%  
10551 \csuse{@glscompstyle@longragged}%  
10552 }%
```

Backward compatible longragged3col style.

```
10553 \compatglossarystyle{longragged3col}{%  
10554 \renewcommand*{\glossaryentryfield}[5]{%  
10555 \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%  
10556 \renewcommand*{\glossarysubentryfield}[6]{%  
10557 &  
10558 \glssubentryitem{##2}%  
10559 \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%  
10560 }%
```

Backward compatible longragged3colborder style.

```
10561 \compatglossarystyle{longragged3colborder}{%  
10562 \csuse{@glscompstyle@longragged3col}%  
10563 }%
```

Backward compatible longragged3colheader style.

```
10564 \compatglossarystyle{longragged3colheader}{%  
10565 \csuse{@glscompstyle@longragged3col}%  
10566 }%
```

Backward compatible longragged3colheaderborder style.

```
10567 \compatglossarystyle{longragged3colheaderborder}{%  
10568 \csuse{@glscompstyle@longragged3col}%  
10569 }%
```

Backward compatible altlongragged4col style.

```
10570 \compatglossarystyle{altlongragged4col}{%  
10571 \renewcommand*{\glossaryentryfield}[5]{%  
10572 \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%  
10573 \renewcommand*{\glossarysubentryfield}[6]{%  
10574 &  
10575 \glssubentryitem{##2}%  
10576 \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%  
10577 }%
```

Backward compatible altlongragged4colheader style.

```
10578 \compatglossarystyle{altlongragged4colheader}{%  
10579 \csuse{@glscompstyle@altlong4col}%  
10580 }%
```

Backward compatible altlongragged4colborder style.

```
10581 \compatglossarystyle{altlongragged4colborder}{%
```

```

10582 \csuse{@glscompstyle@altlong4col}%
10583 }%

```

Backward compatible altlongragged4colheaderborder style.

```

10584 \compatglossarystyle{altlongragged4colheaderborder}{%
10585 \csuse{@glscompstyle@altlong4col}%
10586 }%

```

Backward compatible index style.

```

10587 \compatglossarystyle{index}{%
10588 \renewcommand*{\glossaryentryfield}[5]{%
10589 \item\glsentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10590 \ifx\relax##4\relax
10591 \else
10592 \space{##4}%
10593 \fi
10594 \space ##3\glspostdescription \space ##5}%
10595 \renewcommand*{\glossarysubentryfield}[6]{%
10596 \ifcase##1\relax
10597 % level 0
10598 \item
10599 \or
10600 % level 1
10601 \subitem
10602 \glssubentryitem{##2}%
10603 \else
10604 % all other levels
10605 \subsubitem
10606 \fi
10607 \textbf{\glstarget{##2}{##3}}%
10608 \ifx\relax##5\relax
10609 \else
10610 \space{##5}%
10611 \fi
10612 \space##4\glspostdescription\space ##6}%
10613 }%

```

Backward compatible indexgroup style.

```

10614 \compatglossarystyle{indexgroup}{%
10615 \csuse{@glscompstyle@index}%
10616 }%

```

Backward compatible indexhypergroup style.

```

10617 \compatglossarystyle{indexhypergroup}{%
10618 \csuse{@glscompstyle@index}%
10619 }%

```

Backward compatible tree style.

```

10620 \compatglossarystyle{tree}{%
10621 \renewcommand{\glossaryentryfield}[5]{%
10622 \hangindent0pt\relax

```

```

10623 \parindent0pt\relax
10624 \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10625 \ifx\relax##4\relax
10626 \else
10627 \space(##4)%
10628 \fi
10629 \space ##3\glspostdescription \space ##5\par}%
10630 \renewcommand{\glossarysubentryfield}[6]{%
10631 \hangindent##1\glstreeindent\relax
10632 \parindent##1\glstreeindent\relax
10633 \ifnum##1=1\relax
10634 \glssubentryitem{##2}%
10635 \fi
10636 \textbf{\glstarget{##2}{##3}}%
10637 \ifx\relax##5\relax
10638 \else
10639 \space(##5)%
10640 \fi
10641 \space##4\glspostdescription\space ##6\par}%
10642 }%

```

Backward compatible treegroup style.

```

10643 \compatglossarystyle{treegroup}{%
10644 \csuse{@glscmpstyle@tree}%
10645 }%

```

Backward compatible treehypergroup style.

```

10646 \compatglossarystyle{treehypergroup}{%
10647 \csuse{@glscmpstyle@tree}%
10648 }%

```

Backward compatible treenoname style.

```

10649 \compatglossarystyle{treenoname}{%
10650 \renewcommand{\glossaryentryfield}[5]{%
10651 \hangindent0pt\relax
10652 \parindent0pt\relax
10653 \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
10654 \ifx\relax##4\relax
10655 \else
10656 \space(##4)%
10657 \fi
10658 \space ##3\glspostdescription \space ##5\par}%
10659 \renewcommand{\glossarysubentryfield}[6]{%
10660 \hangindent##1\glstreeindent\relax
10661 \parindent##1\glstreeindent\relax
10662 \ifnum##1=1\relax
10663 \glssubentryitem{##2}%
10664 \fi
10665 \glstarget{##2}{\strut}%
10666 ##4\glspostdescription\space ##6\par}%
10667 }%

```

Backward compatible treenonamegroup style.

```
10668 \compatglossarystyle{treenonamegroup}{%
10669   \csuse{@glscompstyle@treenoname}%
10670 }%
```

Backward compatible treenonamehypergroup style.

```
10671 \compatglossarystyle{treenonamehypergroup}{%
10672   \csuse{@glscompstyle@treenoname}%
10673 }%
```

Backward compatible alttree style.

```
10674 \compatglossarystyle{alttree}{%
10675   \renewcommand{\glossaryentryfield}[5]{%
10676     \ifnum\@gls@prevlevel=0\relax
10677     \else
10678       \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
10679       \hangindent\glstreeindent
10680       \parindent\glstreeindent
10681     \fi
10682     \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
10683       \glssentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
10684     \ifx\relax##4\relax
10685     \else
10686       (##4)\space
10687     \fi
10688     ##3\glspostdescription \space ##5\par
10689     \def\@gls@prevlevel{0}%
10690   }%
10691   \renewcommand{\glossarysubentryfield}[6]{%
10692     \ifnum##1=1\relax
10693       \glssubentryitem{##2}%
10694     \fi
10695     \ifnum\@gls@prevlevel=##1\relax
10696     \else
10697       \@ifundefined{@glswidestname\romannumeral##1}{%
10698         \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
10699         \settowidth{\gls@tmplen}{\textbf{%
10700           \csname @glswidestname\romannumeral##1\endcsname\space}}}%
10701         \ifnum\@gls@prevlevel<##1\relax
10702           \setlength\glstreeindent\gls@tmplen
10703           \addtolength\glstreeindent\parindent
10704           \parindent\glstreeindent
10705         \else
10706           \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
10707             \settowidth{\glstreeindent}{\textbf{%
10708               \@glswidestname\space}}{%
10709             \settowidth{\glstreeindent}{\textbf{%
10710               \csname @glswidestname\romannumeral\@gls@prevlevel
10711               \endcsname\space}}}%
10712             \addtolength\parindent{-\glstreeindent}%

```

```

10713         \setlength\glstreeindent\parindent
10714     \fi
10715 \fi
10716 \hangindent\glstreeindent
10717 \makebox[0pt][r]{\makebox[\glstemplen][l]{%
10718     \textbf{\glstarget{##2}{##3}}}%
10719 \ifx##5\relax\relax
10720 \else
10721     (##5)\space
10722 \fi
10723 ##4\glspostdescription\space ##6\par
10724 \def\@gls@prevlevel{##1}%
10725 }%
10726 }%

```

Backward compatible alttreegroup style.

```

10727 \compatglossarystyle{alttreegroup}{%
10728 \csuse{@glscompstyle@almtree}%
10729 }%

```

Backward compatible almtreehypergroup style.

```

10730 \compatglossarystyle{almtreehypergroup}{%
10731 \csuse{@glscompstyle@almtree}%
10732 }%

```

Backward compatible mcolindex style.

```

10733 \compatglossarystyle{mcolindex}{%
10734 \csuse{@glscompstyle@index}%
10735 }%

```

Backward compatible mcolindexgroup style.

```

10736 \compatglossarystyle{mcolindexgroup}{%
10737 \csuse{@glscompstyle@index}%
10738 }%

```

Backward compatible mcolindexhypergroup style.

```

10739 \compatglossarystyle{mcolindexhypergroup}{%
10740 \csuse{@glscompstyle@index}%
10741 }%

```

Backward compatible mcoltree style.

```

10742 \compatglossarystyle{mcoltree}{%
10743 \csuse{@glscompstyle@tree}%
10744 }%

```

Backward compatible mcoltreegroup style.

```

10745 \compatglossarystyle{mcolindextreegroup}{%
10746 \csuse{@glscompstyle@tree}%
10747 }%

```

Backward compatible mcoltreehypergroup style.

```

10748 \compatglossarystyle{mcolindextreehypergroup}{%

```

10749 \csuse{@glscompstyle@tree}%
 10750 }%

Backward compatible mcoltreenoname style.

10751 \compatglossarystyle{mcoltreenoname}{%
 10752 \csuse{@glscompstyle@tree}%
 10753 }%

Backward compatible mcoltreenonamegroup style.

10754 \compatglossarystyle{mcoltreenonamegroup}{%
 10755 \csuse{@glscompstyle@tree}%
 10756 }%

Backward compatible mcoltreenonamehypergroup style.

10757 \compatglossarystyle{mcoltreenonamehypergroup}{%
 10758 \csuse{@glscompstyle@tree}%
 10759 }%

Backward compatible mcolalttree style.

10760 \compatglossarystyle{mcolalttree}{%
 10761 \csuse{@glscompstyle@alttree}%
 10762 }%

Backward compatible mcolalttreegroup style.

10763 \compatglossarystyle{mcolalttreegroup}{%
 10764 \csuse{@glscompstyle@alttree}%
 10765 }%

Backward compatible mcolalttreehypergroup style.

10766 \compatglossarystyle{mcolalttreehypergroup}{%
 10767 \csuse{@glscompstyle@alttree}%
 10768 }%

Backward compatible superragged style.

10769 \compatglossarystyle{superragged}{%
 10770 \renewcommand*{\glossaryentryfield}[5]{%
 10771 \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
 10772 \tabularnewline}%
 10773 \renewcommand*{\glossarysubentryfield}[6]{%
 10774 &
 10775 \glssubentryitem{##2}%
 10776 \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
 10777 \tabularnewline}%
 10778 }%

Backward compatible superraggedborder style.

10779 \compatglossarystyle{superraggedborder}{%
 10780 \csuse{@glscompstyle@superragged}%
 10781 }%

Backward compatible superraggedheader style.

10782 \compatglossarystyle{superraggedheader}{%
 10783 \csuse{@glscompstyle@superragged}%
 10784 }%

Backward compatible superraggedheaderborder style.

```
10785 \compatglossarystyle{superraggedheaderborder}{%
10786   \csuse{@glscompstyle@superragged}%
10787 }%
```

Backward compatible superragged3col style.

```
10788 \compatglossarystyle{superragged3col}{%
10789   \renewcommand*{\glossaryentryfield}[5]{%
10790     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
10791   \renewcommand*{\glossarysubentryfield}[6]{%
10792     &
10793     \glssubentryitem{##2}%
10794     \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
10795 }%
```

Backward compatible superragged3colborder style.

```
10796 \compatglossarystyle{superragged3colborder}{%
10797   \csuse{@glscompstyle@superragged3col}%
10798 }%
```

Backward compatible superragged3colheader style.

```
10799 \compatglossarystyle{superragged3colheader}{%
10800   \csuse{@glscompstyle@superragged3col}%
10801 }%
```

Backward compatible superragged3colheaderborder style.

```
10802 \compatglossarystyle{superragged3colheaderborder}{%
10803   \csuse{@glscompstyle@superragged3col}%
10804 }%
```

Backward compatible altsuperragged4col style.

```
10805 \compatglossarystyle{altsuperragged4col}{%
10806   \renewcommand*{\glossaryentryfield}[5]{%
10807     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
10808   \renewcommand*{\glossarysubentryfield}[6]{%
10809     &
10810     \glssubentryitem{##2}%
10811     \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
10812 }%
```

Backward compatible altsuperragged4colheader style.

```
10813 \compatglossarystyle{altsuperragged4colheader}{%
10814   \csuse{@glscompstyle@altsuperragged4col}%
10815 }%
```

Backward compatible altsuperragged4colborder style.

```
10816 \compatglossarystyle{altsuperragged4colborder}{%
10817   \csuse{@glscompstyle@altsuperragged4col}%
10818 }%
```

Backward compatible altsuperragged4colheaderborder style.

```
10819 \compatglossarystyle{altsuperragged4colheaderborder}{%
```



```
10820 \csuse{@glscompstyle@altsuperragged4col}%
10821 }%
```

Backward compatible super style.

```
10822 \compatglossarystyle{super}{%
10823   \renewcommand*{\glossaryentryfield}[5]{%
10824     \glstarget{##1}{\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
10825   \renewcommand*{\glossarysubentryfield}[6]{%
10826     &
10827     \glssubentryitem{##2}%
10828     \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
10829 }%
```

Backward compatible superborder style.

```
10830 \compatglossarystyle{superborder}{%
10831   \csuse{@glscompstyle@super}%
10832 }%
```

Backward compatible superheader style.

```
10833 \compatglossarystyle{superheader}{%
10834   \csuse{@glscompstyle@super}%
10835 }%
```

Backward compatible superheaderborder style.

```
10836 \compatglossarystyle{superheaderborder}{%
10837   \csuse{@glscompstyle@super}%
10838 }%
```

Backward compatible super3col style.

```
10839 \compatglossarystyle{super3col}{%
10840   \renewcommand*{\glossaryentryfield}[5]{%
10841     \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##5\\}%
10842   \renewcommand*{\glossarysubentryfield}[6]{%
10843     &
10844     \glssubentryitem{##2}%
10845     \glstarget{##2}{\strut}##4 & ##6\\}%
10846 }%
```

Backward compatible super3colborder style.

```
10847 \compatglossarystyle{super3colborder}{%
10848   \csuse{@glscompstyle@super3col}%
10849 }%
```

Backward compatible super3colheader style.

```
10850 \compatglossarystyle{super3colheader}{%
10851   \csuse{@glscompstyle@super3col}%
10852 }%
```

Backward compatible super3colheaderborder style.

```
10853 \compatglossarystyle{super3colheaderborder}{%
10854   \csuse{@glscompstyle@super3col}%
10855 }%
```

Backward compatible super4col style.

```
10856 \compatglossarystyle{super4col}{%
10857   \renewcommand*{\glossaryentryfield}[5]{%
10858     \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
10859   \renewcommand*{\glossarysubentryfield}[6]{%
10860     &
10861     \glssubentryitem{##2}%
10862     \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
10863 }%
```

Backward compatible super4colheader style.

```
10864 \compatglossarystyle{super4colheader}{%
10865   \csuse{@glscompstyle@super4col}%
10866 }%
```

Backward compatible super4colborder style.

```
10867 \compatglossarystyle{super4colborder}{%
10868   \csuse{@glscompstyle@super4col}%
10869 }%
```

Backward compatible super4colheaderborder style.

```
10870 \compatglossarystyle{super4colheaderborder}{%
10871   \csuse{@glscompstyle@super4col}%
10872 }%
```

Backward compatible altsuper4col style.

```
10873 \compatglossarystyle{altsuper4col}{%
10874   \csuse{@glscompstyle@super4col}%
10875 }%
```

Backward compatible altsuper4colheader style.

```
10876 \compatglossarystyle{altsuper4colheader}{%
10877   \csuse{@glscompstyle@super4col}%
10878 }%
```

Backward compatible altsuper4colborder style.

```
10879 \compatglossarystyle{altsuper4colborder}{%
10880   \csuse{@glscompstyle@super4col}%
10881 }%
```

Backward compatible altsuper4colheaderborder style.

```
10882 \compatglossarystyle{altsuper4colheaderborder}{%
10883   \csuse{@glscompstyle@super4col}%
10884 }%
```

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.

```
10885 \NeedsTeXFormat{LaTeX2e}
```

Package version number now in line with main glossaries package number.

```
10886 \ProvidesPackage{glossaries-accsupp}[2021/10/19 v4.48 (NLCT)]
```

```
10887 Experimental glossaries accessibility]
```

Pass all options to glossaries:

```
10888 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
10889 \ProcessOptions
```

This package should be loaded before glossaries-extra, so complain if that has already been loaded.

```
10890 \ifpackageloaded{glossaries-extra}
```

```
10891 {%
```

If the accsupp option was used, \@glsxtr@doaccsupp will have been set, otherwise it will be empty.

```
10892 \ifx\@glsxtr@doaccsupp\empty
```

```
10893 \GlossariesWarning{The ‘glossaries-accsupp’
```

```
10894 package has been loaded\MessageBreak
```

```
10895 after the ‘glossaries-extra’ package. This\MessageBreak
```

```
10896 can cause a failure to integrate both packages. \MessageBreak
```

```
10897 Either use the ‘accsupp’ option when you load\MessageBreak
```

```
10898 ‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
```

```
10899 before loading ‘glossaries-extra’}%
```

```
10900 \fi
```

```
10901 }
```

```
10902 {}
```

tibleglossentry Override style compatibility macros:

```
10903 \def\compatibleglossentry#1#2{%
```

```
10904 \toks0{#2}%
```

```
10905 \protected@edef\@do@glossentry{%
```

```
10906 \noexpand\accsuppglossaryentryfield{#1}%
```

```
10907 {\noexpand\glsnamefont
```

```
10908 {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
```

```

10909     {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
10910     {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
10911     {\the\toks@}%
10912 }%
10913 \@do@glossentry
10914 }

```

lesubglossentry

```

10915 \def\compatiblesubglossentry#1#2#3{%
10916   \toks@{#3}%
10917   \protected@edef\@do@subglossentry{%
10918     \noexpand\accsuppglossarysubentryfield{\number#1}%
10919     {#2}%
10920     {\noexpand\glsnamefont
10921      {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
10922     {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
10923     {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
10924     {\the\toks@}%
10925   }%
10926   \@do@subglossentry
10927 }

```

Required packages:

```

10928 \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.

```

10929 \providecommand{\gls@accsupp@engine}{accsupp}

```

s@accessibility `\gls@accessibility[\options]{\PDF element}{\value}{\content}`

```

10930 \providecommand{\gls@accessibility}[4]{#4}
10931 \ifdefstring\gls@accsupp@engine{accsupp}
10932 {
10933   \RequirePackage{accsupp}
10934   \renewcommand{\gls@accessibility}[4]{%
10935     \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp}%
10936   }
10937 }
10938 {}

```

lsaccessibility `\glsaccessibility[\options]{\PDF element}{\value}{\content}`

User-level command that includes debug info if required.

```

10939 \newcommand{\glsaccessibility}[4] [] {%
10940   \@glsshowaccsupp{#1}{#2}{#3}%
10941   \gls@accessibility{#1}{#2}{#3}{#4}%
10942 }

```

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:

```

10943 \define@key{glossentry}{access}{%
10944   \def\@glo@access{#1}%
10945 }

```

textaccess The replacement text corresponding to the text key:

```

10946 \define@key{glossentry}{textaccess}{%
10947   \def\@glo@textaccess{#1}%
10948 }

```

firstaccess The replacement text corresponding to the first key:

```

10949 \define@key{glossentry}{firstaccess}{%
10950   \def\@glo@firstaccess{#1}%
10951 }

```

pluralaccess The replacement text corresponding to the plural key:

```

10952 \define@key{glossentry}{pluralaccess}{%
10953   \def\@glo@pluralaccess{#1}%
10954 }

```

firstpluralaccess The replacement text corresponding to the firstplural key:

```

10955 \define@key{glossentry}{firstpluralaccess}{%
10956   \def\@glo@firstpluralaccess{#1}%
10957 }

```

symbolaccess The replacement text corresponding to the symbol key:

```

10958 \define@key{glossentry}{symbolaccess}{%
10959   \def\@glo@symbolaccess{#1}%
10960 }

```

symbolpluralaccess The replacement text corresponding to the symbolplural key:

```

10961 \define@key{glossentry}{symbolpluralaccess}{%
10962   \def\@glo@symbolpluralaccess{#1}%
10963 }

```

descriptionaccess The replacement text corresponding to the description key:

```
10964 \define@key{glossentry}{descriptionaccess}{%
10965   \def\@glo@descaccess{#1}%
10966 }
```

ionpluralaccess The replacement text corresponding to the descriptionplural key:

```
10967 \define@key{glossentry}{descriptionpluralaccess}{%
10968   \def\@glo@descpluralaccess{#1}%
10969 }
```

shortaccess The replacement text corresponding to the short key:

```
10970 \define@key{glossentry}{shortaccess}{%
10971   \def\@glo@shortaccess{#1}%
10972 }
```

ortpluralaccess The replacement text corresponding to the shortplural key:

```
10973 \define@key{glossentry}{shortpluralaccess}{%
10974   \def\@glo@shortpluralaccess{#1}%
10975 }
```

longaccess The replacement text corresponding to the long key:

```
10976 \define@key{glossentry}{longaccess}{%
10977   \def\@glo@longaccess{#1}%
10978 }
```

ongpluralaccess The replacement text corresponding to the longplural key:

```
10979 \define@key{glossentry}{longpluralaccess}{%
10980   \def\@glo@longpluralaccess{#1}%
10981 }
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
10982 \define@key{glossentry}{user1access}{%
10983   \def\@glo@useriaccess{#1}%
10984 }
```

user2access The replacement text corresponding to the user2 key:

```
10985 \define@key{glossentry}{user2access}{%
10986   \def\@glo@useriiaccess{#1}%
10987 }
```

user3access The replacement text corresponding to the user3 key:

```
10988 \define@key{glossentry}{user3access}{%
10989   \def\@glo@useriiiaccess{#1}%
10990 }
```

user4access The replacement text corresponding to the user4 key:

```
10991 \define@key{glossentry}{user4access}{%
10992   \def\@glo@userivaccess{#1}%
10993 }
```

user5access The replacement text corresponding to the user5 key:

```
10994 \define@key{glossentry}{user5access}{%
10995   \def\@glo@uservaccess{#1}%
10996 }
```

user6access The replacement text corresponding to the user6 key:

```
10997 \define@key{glossentry}{user6access}{%
10998   \def\@glo@userviaccess{#1}%
10999 }
```

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:

```
11000 \appto\@gls@keymap{,%
11001   {access}{access},%
11002   {textaccess}{textaccess},%
11003   {firstaccess}{firstaccess},%
11004   {pluralaccess}{pluralaccess},%
11005   {firstpluralaccess}{firstpluralaccess},%
11006   {symbolaccess}{symbolaccess},%
11007   {symbolpluralaccess}{symbolpluralaccess},%
11008   {descaccess}{descaccess},%
11009   {descpluralaccess}{descpluralaccess},%
11010   {shortaccess}{shortaccess},%
11011   {shortpluralaccess}{shortpluralaccess},%
11012   {longaccess}{longaccess},%
11013   {longpluralaccess}{longpluralaccess},%
11014   {user1access}{useriaccess},%
11015   {user2access}{useriiaccess},%
11016   {user3access}{useriiiaccess},%
11017   {user4access}{userivaccess},%
11018   {user5access}{uservaccess},%
11019   {user6access}{userviaccess}%
11020 }
```

\@gls@noaccess Indicates that no replacement text has been provided.

```
11021 \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.

```
11022 \let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
11023 \renewcommand*{\@newglossaryentryprehook}{%
```

```

11024 \@gls@oldnewglossaryentryprehook
11025 \def\@glo@access{\relax}%

```

Initialise the other keys:

```

11026 \def\@glo@textaccess{\@glo@access}%
11027 \def\@glo@firstaccess{\@glo@access}%
11028 \def\@glo@pluralaccess{\@glo@textaccess}%
11029 \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
11030 \def\@glo@symbolaccess{\relax}%
11031 \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
11032 \def\@glo@descaccess{\relax}%
11033 \def\@glo@descpluralaccess{\@glo@descaccess}%
11034 \def\@glo@shortaccess{\relax}%
11035 \def\@glo@shortpluralaccess{\@glo@shortaccess}%
11036 \def\@glo@longaccess{\relax}%
11037 \def\@glo@longpluralaccess{\@glo@longaccess}%
11038 \def\@glo@useriaccess{\relax}%
11039 \def\@glo@useriiaaccess{\relax}%
11040 \def\@glo@useriiiaaccess{\relax}%
11041 \def\@glo@userivaccess{\relax}%
11042 \def\@glo@uservaccess{\relax}%
11043 \def\@glo@userviiaaccess{\relax}%
11044 }

```

Add to the end hook:

```

11045 \let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
11046 \renewcommand*{\@newglossaryentryposthook}{%
11047 \@gls@oldnewglossaryentryposthook

```

Store the access information:

```

11048 \expandafter
11049 \protected@xdef\csname glo@\@glo@label @access\endcsname{%
11050 \@glo@access}%
11051 \expandafter
11052 \protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
11053 \@glo@textaccess}%
11054 \expandafter
11055 \protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
11056 \@glo@firstaccess}%
11057 \expandafter
11058 \protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
11059 \@glo@pluralaccess}%
11060 \expandafter
11061 \protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
11062 \@glo@firstpluralaccess}%
11063 \expandafter
11064 \protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
11065 \@glo@symbolaccess}%
11066 \expandafter
11067 \protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
11068 \@glo@symbolpluralaccess}%

```



```

11069 \expandafter
11070   \protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
11071     \@glo@descaccess}%
11072 \expandafter
11073   \protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
11074     \@glo@descpluralaccess}%
11075 \expandafter
11076   \protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
11077     \@glo@shortaccess}%
11078 \expandafter
11079   \protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
11080     \@glo@shortpluralaccess}%
11081 \expandafter
11082   \protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
11083     \@glo@longaccess}%
11084 \expandafter
11085   \protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
11086     \@glo@longpluralaccess}%
11087 \expandafter
11088   \protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
11089     \@glo@useriaccess}%
11090 \expandafter
11091   \protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
11092     \@glo@useriiaccess}%
11093 \expandafter
11094   \protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
11095     \@glo@useriiiaccess}%
11096 \expandafter
11097   \protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
11098     \@glo@userivaccess}%
11099 \expandafter
11100   \protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
11101     \@glo@uservaccess}%
11102 \expandafter
11103   \protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
11104     \@glo@userviaccess}%
11105 }

```

5.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

11106 \newcommand*{\glsentryaccess}[1]{%
11107   \@gls@entry@field{#1}{access}%
11108 }

```

`entrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

11109 \newcommand*{\glsentrytextaccess}[1]{%
11110   \@gls@entry@field{#1}{textaccess}%

```

```

11111 }

entryfirstaccess  Get the value of the firstaccess key for the entry with the given label:
11112 \newcommand*{\glentryfirstaccess}[1]{%
11113   \@gls@entry@field{#1}{firstaccess}%
11114 }

entrypluralaccess  Get the value of the pluralaccess key for the entry with the given label:
11115 \newcommand*{\glentrypluralaccess}[1]{%
11116   \@gls@entry@field{#1}{pluralaccess}%
11117 }

entryfirstpluralaccess  Get the value of the firstpluralaccess key for the entry with the given label:
11118 \newcommand*{\glentryfirstpluralaccess}[1]{%
11119   \@gls@entry@field{#1}{firstpluralaccess}%
11120 }

entrysymbolaccess  Get the value of the symbolaccess key for the entry with the given label:
11121 \newcommand*{\glentrysymbolaccess}[1]{%
11122   \@gls@entry@field{#1}{symbolaccess}%
11123 }

entrysymbolpluralaccess  Get the value of the symbolpluralaccess key for the entry with the given label:
11124 \newcommand*{\glentrysymbolpluralaccess}[1]{%
11125   \@gls@entry@field{#1}{symbolpluralaccess}%
11126 }

entrydescaccess  Get the value of the descriptionaccess key for the entry with the given label:
11127 \newcommand*{\glentrydescaccess}[1]{%
11128   \@gls@entry@field{#1}{descaccess}%
11129 }

entrydescpluralaccess  Get the value of the descriptionpluralaccess key for the entry with the given label:
11130 \newcommand*{\glentrydescpluralaccess}[1]{%
11131   \@gls@entry@field{#1}{descpluralaccess}%
11132 }

entryshortaccess  Get the value of the shortaccess key for the entry with the given label:
11133 \newcommand*{\glentryshortaccess}[1]{%
11134   \@gls@entry@field{#1}{shortaccess}%
11135 }

entryshortpluralaccess  Get the value of the shortpluralaccess key for the entry with the given label:
11136 \newcommand*{\glentryshortpluralaccess}[1]{%
11137   \@gls@entry@field{#1}{shortpluralaccess}%
11138 }

```

entrylongaccess Get the value of the longaccess key for the entry with the given label:

```
11139 \newcommand*{\glentrylongaccess}[1]{%
11140   \@gls@entry@field{#1}{longaccess}%
11141 }
```

ongpluralaccess Get the value of the longpluralaccess key for the entry with the given label:

```
11142 \newcommand*{\glentrylongpluralaccess}[1]{%
11143   \@gls@entry@field{#1}{longpluralaccess}%
11144 }
```

entryuseriaccess Get the value of the user1access key for the entry with the given label:

```
11145 \newcommand*{\glentryuseriaccess}[1]{%
11146   \@gls@entry@field{#1}{useriaccess}%
11147 }
```

tryuseriiaccess Get the value of the user2access key for the entry with the given label:

```
11148 \newcommand*{\glentryuseriiaccess}[1]{%
11149   \@gls@entry@field{#1}{useriiaccess}%
11150 }
```

ryuseriiiaccess Get the value of the user3access key for the entry with the given label:

```
11151 \newcommand*{\glentryuseriiiaccess}[1]{%
11152   \@gls@entry@field{#1}{useriiiaccess}%
11153 }
```

tryuserivaccess Get the value of the user4access key for the entry with the given label:

```
11154 \newcommand*{\glentryuserivaccess}[1]{%
11155   \@gls@entry@field{#1}{userivaccess}%
11156 }
```

entryuservaccess Get the value of the user5access key for the entry with the given label:

```
11157 \newcommand*{\glentryuservaccess}[1]{%
11158   \@gls@entry@field{#1}{uservaccess}%
11159 }
```

tryuserviaccess Get the value of the user6access key for the entry with the given label:

```
11160 \newcommand*{\glentryuserviaccess}[1]{%
11161   \@gls@entry@field{#1}{userviaccess}%
11162 }
```

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`.

```
11163 \newcommand{\glsfieldaccsupp}[4]{%
11164   \ifdef\glscategory
11165   {%
11166     \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
11167     {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
11168     {%
11169       \ifcsdef{glsxtr\glscategory{#4}accsupp}%
11170       {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
11171       {%
11172         \ifcsdef{gls#3accsupp}%
11173         {\csname gls#3accsupp\endcsname{#1}{#2}}%
11174         {\glsaccsupp{#1}{#2}}%
11175       }%
11176     }%
11177   }%
11178   {%
11179     \ifcsdef{gls#3accsupp}%
11180     {\csname gls#3accsupp\endcsname{#1}{#2}}%
11181     {\glsaccsupp{#1}{#2}}%
11182   }%
11183 }
```

glsfieldaccsupp

`\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}`

As `\glsfieldaccsupp` but fully expand replacement text.

```
11184 \newcommand{\xglsfieldaccsupp}[1]{%
11185   \protected@edef\@gls@replacementtext{#1}%
11186   \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
11187 }
```

glsshortaccsupp

`\glsshortaccsupp{<replacement text>}{<text>}`

```
11188 \newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

`\glsshortplaccsupp` `\glsshortplaccsupp{<replacement text>}{<text>}`

```
11189 \newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

`\glsaccsupp` `\glsaccsupp{<replacement text>}{<text>}`

```
11190 \newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
```

`\xglsaccsupp` Fully expands replacement text before calling `\glsaccsupp`

```
11191 \newcommand*{\xglsaccsupp}[2]{%
11192   \protected@edef\@gls@replacementtext{#1}%
11193   \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
11194 }
```

`@access@display` Deprecated. Use `\@gls@fieldaccess@display` instead.

```
11195 \newcommand*{\@gls@access@display}[2]{%
11196   \protected@edef\@glo@access{#2}%
11197   \ifx\@glo@access\@gls@noaccess
11198     #1%
11199   \else
11200     \xglsaccsupp{\@glo@access}{#1}%
11201   \fi
11202 }
```

`daccess@display` `\@gls@fieldaccess@display{<label>}{<field>}{<content>}{<replacement>}`

```
11203 \newcommand*{\@gls@fieldaccess@display}[4]{%
11204   \protected@edef\@glo@access{#4}%
11205   \ifdefequal\@glo@access\@gls@noaccess
11206     {#3}%
11207   {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
11208 }
```

`meaccessdisplay` Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```
11209 \newrobustcmd*{\glsnameaccessdisplay}[2]{%
11210   \ifcsundef{glo@glsdetoklabel{#2}@access}%
11211     {#1}%
11212   {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
11213 }
```

`xtaccessdisplay` As above but for the textaccess replacement text.

```
11214 \newrobustcmd*{\glsxtaccessdisplay}[2]{%
11215   \ifcsundef{glo@glsdetoklabel{#2}@textaccess}%
11216     {#1}%
11217 }
```

```

11217 {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
11218 }

```

alaccessdisplay As above but for the pluralaccess replacement text.

```

11219 \newrobustcmd*{\glspluralaccessdisplay}[2]{%
11220 \ifcsundef{glo@glsdetoklabel{#2}@pluralaccess}%
11221 {#1}%
11222 {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
11223 }

```

staccessdisplay As above but for the firstaccess replacement text.

```

11224 \newrobustcmd*{\glsfirstaccessdisplay}[2]{%
11225 \ifcsundef{glo@glsdetoklabel{#2}@firstaccess}%
11226 {#1}%
11227 {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
11228 }

```

alaccessdisplay As above but for the firstpluralaccess replacement text.

```

11229 \newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
11230 \ifcsundef{glo@glsdetoklabel{#2}@firstpluralaccess}%
11231 {#1}%
11232 {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
11233 }

```

olaccessdisplay As above but for the symbolaccess replacement text.

```

11234 \newrobustcmd*{\glsymbolaccessdisplay}[2]{%
11235 \ifcsundef{glo@glsdetoklabel{#2}@symbolaccess}%
11236 {#1}%
11237 {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
11238 }

```

alaccessdisplay As above but for the symbolpluralaccess replacement text.

```

11239 \newrobustcmd*{\glsymbolpluralaccessdisplay}[2]{%
11240 \ifcsundef{glo@glsdetoklabel{#2}@symbolpluralaccess}%
11241 {#1}%
11242 {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
11243 }

```

onaccessdisplay As above but for the descriptionaccess replacement text.

```

11244 \newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
11245 \ifcsundef{glo@glsdetoklabel{#2}@descaccess}%
11246 {#1}%
11247 {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
11248 }

```

alaccessdisplay As above but for the descriptionpluralaccess replacement text.

```

11249 \newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
11250 \ifcsundef{glo@glsdetoklabel{#2}@descpluralaccess}%

```

```

11251 {#1}%
11252 {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glentrydescpluralaccess{#2}}}%
11253 }

```

rtaccessdisplay As above but for the shortaccess replacement text.

```

11254 \newrobustcmd*{\glsshortaccessdisplay}[2]{%
11255 \ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
11256 {#1}%
11257 {\@gls@fieldaccess@display{#2}{short}{#1}{\glentryshortaccess{#2}}}%
11258 }

```

alaccessdisplay As above but for the shortpluralaccess replacement text.

```

11259 \newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
11260 \ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
11261 {#1}%
11262 {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glentryshortpluralaccess{#2}}}%
11263 }

```

ngaccessdisplay As above but for the longaccess replacement text.

```

11264 \newrobustcmd*{\glslongaccessdisplay}[2]{%
11265 \ifcsundef{glo@glsdetoklabel{#2}@longaccess}%
11266 {#1}%
11267 {\@gls@fieldaccess@display{#2}{long}{#1}{\glentrylongaccess{#2}}}%
11268 }

```

alaccessdisplay As above but for the longpluralaccess replacement text.

```

11269 \newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
11270 \ifcsundef{glo@glsdetoklabel{#2}@longpluralaccess}%
11271 {#1}%
11272 {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glentrylongpluralaccess{#2}}}%
11273 }

```

riaccessdisplay As above but for the user1access replacement text.

```

11274 \newrobustcmd*{\gluseriaccessdisplay}[2]{%
11275 \ifcsundef{glo@glsdetoklabel{#2}@useriaccess}%
11276 {#1}%
11277 {\@gls@fieldaccess@display{#2}{useri}{#1}{\glentryuseriaccess{#2}}}%
11278 }

```

iiaccessdisplay As above but for the user2access replacement text.

```

11279 \newrobustcmd*{\gluseriiaccessdisplay}[2]{%
11280 \ifcsundef{glo@glsdetoklabel{#2}@useriiaccess}%
11281 {#1}%
11282 {\@gls@fieldaccess@display{#2}{userii}{#1}{\glentryuseriiaccess{#2}}}%
11283 }

```

iiiaccessdisplay As above but for the user3access replacement text.

```

11284 \newrobustcmd*{\gluseriiiaccessdisplay}[2]{%

```

```

11285 \ifcsundef{glo@glsdetoklabel{#2}@useriiiaccess}%
11286 {#1}%
11287 {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glentryuseriiiaccess{#2}}}%
11288 }

```

ivaccessdisplay As above but for the user4access replacement text.

```

11289 \newrobustcmd*{\gluserivaccessdisplay}[2]{%
11290 \ifcsundef{glo@glsdetoklabel{#2}@userivaccess}%
11291 {#1}%
11292 {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glentryuserivaccess{#2}}}%
11293 }

```

rvaccessdisplay As above but for the user5access replacement text.

```

11294 \newrobustcmd*{\glservaccessdisplay}[2]{%
11295 \ifcsundef{glo@glsdetoklabel{#2}@uservaccess}%
11296 {#1}%
11297 {\@gls@fieldaccess@display{#2}{userv}{#1}{\glentryuservaccess{#2}}}%
11298 }

```

viaccessdisplay As above but for the user6access replacement text.

```

11299 \newrobustcmd*{\glserviaccessdisplay}[2]{%
11300 \ifcsundef{glo@glsdetoklabel{#2}@userviaccess}%
11301 {#1}%
11302 {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glentryuserviaccess{#2}}}%
11303 }

```

lsaccessdisplay Gets the replacement text corresponding to the named key given by the first argument and calls the appropriate command defined above.

```

11304 \newrobustcmd*{\glsaccessdisplay}[3]{%
11305 \ifcsundef{gls#1accessdisplay}%
11306 {%
11307 \PackageError{glossaries-accsupp}{No accessibility support
11308 for key ‘#1’}{}%
11309 }%
11310 {%
11311 \csname gls#1accessdisplay\endcsname{#2}{#3}%
11312 }%
11313 }

```

default@entryfmt Redefine the default entry format to use accessibility information

```

11314 \renewcommand*{\@gls@default@entryfmt}[2]{%
11315 \ifdefempty{glscustomtext}
11316 {%
11317 \glsifplural
11318 {%
11319 \glscapscase
11320 {%

```


Don't adjust case

```
11321      \ifglused\glslabel
11322      {%
```

Subsequent use

```
11323      #2{\glspluralaccessdisplay
11324          {\glentryplural{\glslabel}}{\glslabel}}%
11325          {\gldescriptionpluralaccessdisplay
11326              {\glentrydescplural{\glslabel}}{\glslabel}}%
11327          {\glssymbolpluralaccessdisplay
11328              {\glentrysymbolplural{\glslabel}}{\glslabel}}
11329          {\glinsert}%
11330      }%
11331      {%
```

First use

```
11332      #1{\glfirstpluralaccessdisplay
11333          {\glentryfirstplural{\glslabel}}{\glslabel}}%
11334          {\gldescriptionpluralaccessdisplay
11335              {\glentrydescplural{\glslabel}}{\glslabel}}%
11336          {\glssymbolpluralaccessdisplay
11337              {\glentrysymbolplural{\glslabel}}{\glslabel}}%
11338          {\glinsert}%
11339      }%
11340      }%
11341      {%
```

Make first letter upper case

```
11342      \ifglused\glslabel
11343      {%
```

Subsequent use.

```
11344      #2{\glspluralaccessdisplay
11345          {\Glentryplural{\glslabel}}{\glslabel}}%
11346          {\gldescriptionpluralaccessdisplay
11347              {\glentrydescplural{\glslabel}}{\glslabel}}%
11348          {\glssymbolpluralaccessdisplay
11349              {\glentrysymbolplural{\glslabel}}{\glslabel}}%
11350          {\glinsert}%
11351      }%
11352      {%
```

First use

```
11353      #1{\glfirstpluralaccessdisplay
11354          {\Glentryfirstplural{\glslabel}}{\glslabel}}%
11355          {\gldescriptionpluralaccessdisplay
11356              {\glentrydescplural{\glslabel}}{\glslabel}}%
11357          {\glssymbolpluralaccessdisplay
11358              {\glentrysymbolplural{\glslabel}}{\glslabel}}%
11359          {\glinsert}%
11360      }%
```

11361 }%
 11362 {%

Make all upper case

11363 \ifglsused\glslabel
 11364 {%

Subsequent use

11365 \MakeUppercase{%
 11366 #2{\glspluralaccessdisplay
 11367 {\glsentryplural{\glslabel}}{\glslabel}}%
 11368 {\glsdescriptionpluralaccessdisplay
 11369 {\glsentrydescplural{\glslabel}}{\glslabel}}%
 11370 {\glsymbolpluralaccessdisplay
 11371 {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
 11372 {\glsinsert}}%
 11373 }%
 11374 {%

First use

11375 \MakeUppercase{%
 11376 #1{\glsfirstpluralaccessdisplay
 11377 {\glsentryfirstplural{\glslabel}}{\glslabel}}%
 11378 {\glsdescriptionpluralaccessdisplay
 11379 {\glsentrydescplural{\glslabel}}{\glslabel}}%
 11380 {\glsymbolpluralaccessdisplay
 11381 {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
 11382 {\glsinsert}}%
 11383 }%
 11384 }%
 11385 }%
 11386 {%

Singular form

11387 \glscapscase
 11388 {%

Don't adjust case

11389 \ifglsused\glslabel
 11390 {%

Subsequent use

11391 #2{\glsfirsttextaccessdisplay
 11392 {\glsentrytext{\glslabel}}{\glslabel}}%
 11393 {\glsdescriptionaccessdisplay
 11394 {\glsentrydesc{\glslabel}}{\glslabel}}%
 11395 {\glsymbolaccessdisplay
 11396 {\glsentrysymbol{\glslabel}}{\glslabel}}%
 11397 {\glsinsert}}%
 11398 }%
 11399 {%

First use

```

11400      #1{\glsfirstaccessdisplay
11401          {\glsentryfirst{\glslabel}}{\glslabel}}%
11402          {\glsdescriptionaccessdisplay
11403              {\glsentrydesc{\glslabel}}{\glslabel}}%
11404          {\glsymbolaccessdisplay
11405              {\glsentrysymbol{\glslabel}}{\glslabel}}%
11406          {\glsinsert}}%
11407      }%
11408  }%
11409  {%

```

Make first letter upper case

```

11410      \ifglsused\glslabel
11411      {%

```

Subsequent use

```

11412      #2{\glstextaccessdisplay
11413          {\Glsentrytext{\glslabel}}{\glslabel}}%
11414          {\glsdescriptionaccessdisplay
11415              {\glsentrydesc{\glslabel}}{\glslabel}}%
11416          {\glsymbolaccessdisplay
11417              {\glsentrysymbol{\glslabel}}{\glslabel}}%
11418          {\glsinsert}}%
11419      }%
11420      {%

```

First use

```

11421      #1{\glsfirstaccessdisplay
11422          {\Glsentryfirst{\glslabel}}{\glslabel}}%
11423          {\glsdescriptionaccessdisplay
11424              {\glsentrydesc{\glslabel}}{\glslabel}}%
11425          {\glsymbolaccessdisplay
11426              {\glsentrysymbol{\glslabel}}{\glslabel}}%
11427          {\glsinsert}}%
11428      }%
11429  }%
11430  {%

```

Make all upper case

```

11431      \ifglsused\glslabel
11432      {%

```

Subsequent use

```

11433      \MakeUppercase{%
11434          #2{\glstextaccessdisplay
11435              {\glsentrytext{\glslabel}}{\glslabel}}%
11436              {\glsdescriptionaccessdisplay
11437                  {\glsentrydesc{\glslabel}}{\glslabel}}%
11438                  {\glsymbolaccessdisplay
11439                      {\glsentrysymbol{\glslabel}}{\glslabel}}%

```

```

11440         {\glsinsert}}}%
11441     }%
11442     {%

```

First use

```

11443     \MakeUppercase{%
11444         #1{\glsfirstaccessdisplay
11445             {\glsentryfirst{\glslabel}}{\glslabel}}}%
11446             {\glsdescriptionaccessdisplay
11447                 {\glsentrydesc{\glslabel}}{\glslabel}}}%
11448                 {\glsymbolaccessdisplay
11449                     {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11450                     {\glsinsert}}}%
11451     }%
11452 }%
11453 }%
11454 }%
11455 {%

```

Custom text provided in \glsdisp

```

11456     \ifglsused{\glslabel}%
11457     {%

```

Subsequent use

```

11458         #2{\glscustomtext}%
11459         {\glsdescriptionaccessdisplay
11460             {\glsentrydesc{\glslabel}}{\glslabel}}}%
11461             {\glsymbolaccessdisplay
11462                 {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11463                 {\glsinsert}}}%
11464     }%
11465     {%

```

First use

```

11466         #1{\glscustomtext}%
11467         {\glsdescriptionaccessdisplay
11468             {\glsentrydesc{\glslabel}}{\glslabel}}}%
11469             {\glsymbolaccessdisplay
11470                 {\glsentrysymbol{\glslabel}}{\glslabel}}}%
11471                 {\glsinsert}}}%
11472     }%
11473 }%
11474 }

```

\glsгенentryfmt Redefine to use accessibility information.

```

11475 \renewcommand*{\glsгенentryfmt}{%
11476     \ifdefempty\glscustomtext
11477     {%
11478         \glsifplural
11479         {%

```

Plural form

```
11480      \glscapscase
11481      {%
```

Don't adjust case

```
11482      \ifglused\glslabel
11483      {%
```

Subsequent use

```
11484      \glspluralaccessdisplay
11485          {\glentryplural{\glslabel}}{\glslabel}%
11486      \glinsert
11487      }%
11488      {%
```

First use

```
11489      \glfirstpluralaccessdisplay
11490          {\glentryfirstplural{\glslabel}}{\glslabel}%
11491      \glinsert
11492      }%
11493      }%
11494      {%
```

Make first letter upper case

```
11495      \ifglused\glslabel
11496      {%
```

Subsequent use.

```
11497      \glspluralaccessdisplay
11498          {\Glsentryplural{\glslabel}}{\glslabel}%
11499      \glinsert
11500      }%
11501      {%
```

First use

```
11502      \glfirstpluralaccessdisplay
11503          {\Glsentryfirstplural{\glslabel}}{\glslabel}%
11504      \glinsert
11505      }%
11506      }%
11507      {%
```

Make all upper case

```
11508      \ifglused\glslabel
11509      {%
```

Subsequent use

```
11510      \glspluralaccessdisplay
11511          {\mfirstucMakeUppercase{\glentryplural{\glslabel}}}%
11512          {\glslabel}%
11513      \mfirstucMakeUppercase{\glinsert}%
11514      }%
11515      {%
```

First use

```
11516      \glsfirstpluralaccessdisplay
11517      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
11518      {\glslabel}%
11519      \mfirstucMakeUppercase{\glsinsert}%
11520      }%
11521      }%
11522      }%
11523      {%
```

Singular form

```
11524      \glscapscase
11525      {%
```

Don't adjust case

```
11526      \ifglsused\glslabel
11527      {%
```

Subsequent use

```
11528      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
11529      \glsinsert
11530      }%
11531      {%
```

First use

```
11532      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
11533      \glsinsert
11534      }%
11535      }%
11536      {%
```

Make first letter upper case

```
11537      \ifglsused\glslabel
11538      {%
```

Subsequent use

```
11539      \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
11540      \glsinsert
11541      }%
11542      {%
```

First use

```
11543      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
11544      \glsinsert
11545      }%
11546      }%
11547      {%
```

Make all upper case

```
11548      \ifglsused\glslabel
11549      {%
```

Subsequent use

```
11550      \glstextaccessdisplay
11551      {\mfirstucMakeUppercase{\glentrytext{\glslabel}}}{\glslabel}%
11552      \mfirstucMakeUppercase{\glsinsert}%
11553      }%
11554      {%
```

First use

```
11555      \glsfirstaccessdisplay
11556      {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
11557      \mfirstucMakeUppercase{\glsinsert}%
11558      }%
11559      }%
11560      }%
11561      }%
11562      {%
```

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.

```
11563      \glscustomtext\glsinsert
11564      }%
11565 }
```

`\glsgenacfmt` Redefine to include accessibility information.

```
11566 \renewcommand*{\glsgenacfmt}{%
11567   \ifdefempty\glscustomtext
11568   {%
11569     \ifglsused\glslabel
11570     {%
```

Subsequent use:

```
11571      \glsifplural
11572      {%
```

Subsequent plural form:

```
11573      \glscapscase
11574      {%
```

Subsequent plural form, don't adjust case:

```
11575      \acronymfont
11576      {\glsshortpluralaccessdisplay
11577       {\glentryshortpl{\glslabel}}{\glslabel}}%
11578      \glsinsert
11579      }%
11580      {%
```

Subsequent plural form, make first letter upper case:

```
11581      \acronymfont
11582      {\glsshortpluralaccessdisplay
11583       {\Glsentryshortpl{\glslabel}}{\glslabel}}%
11584      \glsinsert
```

11585 }%
 11586 {%

Subsequent plural form, all caps:

11587 \mfirstucMakeUppercase
 11588 {\acronymfont
 11589 {\glsshortpluralaccessdisplay
 11590 {\glentryshortpl{\glslabel}}{\glslabel}}}%
 11591 \glsinsert}%
 11592 }%
 11593 }%
 11594 {%

Subsequent singular form

11595 \glscapscase
 11596 {%

Subsequent singular form, don't adjust case:

11597 \acronymfont
 11598 {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}}%
 11599 \glsinsert
 11600 }%
 11601 {%

Subsequent singular form, make first letter upper case:

11602 \acronymfont
 11603 {\glsshortaccessdisplay{\Glentryshort{\glslabel}}{\glslabel}}}%
 11604 \glsinsert
 11605 }%
 11606 {%

Subsequent singular form, all caps:

11607 \mfirstucMakeUppercase
 11608 {\acronymfont{%
 11609 \glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}}%
 11610 \glsinsert}%
 11611 }%
 11612 }%
 11613 }%
 11614 {%

First use:

11615 \glsifplural
 11616 {%

First use plural form:

11617 \glscapscase
 11618 {%

First use plural form, don't adjust case:

11619 \genplacrformat{\glslabel}{\glsinsert}%
 11620 }%
 11621 {%

First use plural form, make first letter upper case:

```
11622      \Genplacrfullformat{\glslabel}{\glsinsert}%
11623      }%
11624      {%
```

First use plural form, all caps:

```
11625      \mfirstucMakeUppercase
11626      {\genplacrfullformat{\glslabel}{\glsinsert}}%
11627      }%
11628      }%
11629      {%
```

First use singular form

```
11630      \glscapscase
11631      {%
```

First use singular form, don't adjust case:

```
11632      \genacrfullformat{\glslabel}{\glsinsert}%
11633      }%
11634      {%
```

First use singular form, make first letter upper case:

```
11635      \Genacrfullformat{\glslabel}{\glsinsert}%
11636      }%
11637      {%
```

First use singular form, all caps:

```
11638      \mfirstucMakeUppercase
11639      {\genacrfullformat{\glslabel}{\glsinsert}}%
11640      }%
11641      }%
11642      }%
11643      }%
11644      {%
```

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.

```
11645      \glscustomtext
11646      }%
11647 }
```

`\genacrfullformat` Redefine to include accessibility information.

```
11648 \renewcommand*{\genacrfullformat}[2]{%
11649   \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
11650   (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1}}%
11651 }
```

`\genacrfullformat` Redefine to include accessibility information.

```
11652 \renewcommand*{\Genacrfullformat}[2]{%
11653   \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
11654   (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1}}%
11655 }
```

placrfullformat Redefine to include accessibility information.

```
11656 \renewcommand*{\genplacrfullformat}[2]{%
11657   \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
11658   (\glsshortpluralaccessdisplay
11659     {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
11660 }
```

placrfullformat Redefine to include accessibility information.

```
11661 \renewcommand*{\Genplacrfullformat}[2]{%
11662   \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
11663   (\glsshortpluralaccessdisplay
11664     {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
11665 }
```

\@acrshort

```
11666 \def\@acrshort#1#2[#3]{%
11667   \glsdoifexists{#2}%
11668   {%
11669     \let\do@gls@link@checkfirsthyper\relax

11670     \let\glsifplural\@secondoftwo
11671     \let\glscapscase\@firstofthree
11672     \let\glsinsert\@empty
11673     \def\glscustomtext{%
11674       \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
11675     }%

    Call \@gls@link

11676     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11677   }%

11678   \glspostlinkhook
11679 }
```

\@Acrshort

```
11680 \def\@Acrshort#1#2[#3]{%
11681   \glsdoifexists{#2}%
11682   {%
11683     \let\do@gls@link@checkfirsthyper\relax

11684     \let\glsifplural\@secondoftwo
11685     \let\glscapscase\@secondofthree
11686     \let\glsinsert\@empty
11687     \def\glscustomtext{%
11688       \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
11689     }%

    Call \@gls@link

11690     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11691   }%
```

```

11692 \glspostlinkhook
11693 }

```

\@ACRshort

```

11694 \def\@ACRshort#1#2[#3]{%
11695   \glstoifexists{#2}%
11696   {%
11697     \let\do@gl@link@checkfirsthyper\relax

11698     \let\gl@sifplural\@secondoftwo
11699     \let\gl@scapscase\@thirdofthree
11700     \let\gl@insert\@empty
11701     \def\glscustomtext{%
11702       \acronymfont{\glsshortaccessdisplay
11703         {\MakeUppercase{\glentryshort{#2}}}{#2}}#3%
11704     }%

    Call\@gl@link

11705     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11706     }%

11707   \glspostlinkhook
11708 }

```

\@acrlong

```

11709 \def\@acrlong#1#2[#3]{%
11710   \glstoifexists{#2}%
11711   {%
11712     \let\do@gl@link@checkfirsthyper\relax

11713     \let\gl@sifplural\@secondoftwo
11714     \let\gl@scapscase\@firstofthree
11715     \let\gl@insert\@empty
11716     \def\glscustomtext{%
11717       \acronymfont{\glslongaccessdisplay{\glentrylong{#2}}{#2}}#3%
11718     }%

    Call\@gl@link

11719     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
11720     }%

11721   \glspostlinkhook
11722 }

```

\@Acrlong

```

11723 \def\@Acrlong#1#2[#3]{%
11724   \glstoifexists{#2}%
11725   {%
11726     \let\do@gl@link@checkfirsthyper\relax

```

```

11727 \let\glsifplural\@secondoftwo
11728 \let\glsifcaps\@firstofthree
11729 \let\glsinsert\@empty
11730 \def\glsfont{\fontfamily\glsfontfamily\fontseries\glsfontseries\fontsize\glsfontsize\fontstyle\glsfontstyle}
11731 \def\glsfontfamily{\fontfamily\glsfontfamily}
11732 }%

```

Call \@gls@link

```

11733 \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
11734 }%

11735 \glspostlinkhook
11736 }

```

\@ACRlong

```

11737 \def\@ACRlong#1#2[#3]{%
11738 \glsdoifexists{#2}%
11739 {%
11740 \let\do@gls@link@checkfirsthyper\relax

11741 \let\glsifplural\@secondoftwo
11742 \let\glsifcaps\@firstofthree
11743 \let\glsinsert\@empty
11744 \def\glsfont{\fontfamily\glsfontfamily\fontseries\glsfontseries\fontsize\glsfontsize\fontstyle\glsfontstyle}
11745 \def\glsfontfamily{\fontfamily\glsfontfamily}
11746 \MakeUppercase{\glsentrylong{#2}}{#2}#3}%
11747 }%

```

Call \@gls@link

```

11748 \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
11749 }%

11750 \glspostlinkhook
11751 }

```

\@gls@text@

```

11752 \def\@gls@text@#1#2[#3]{%
11753 \gls@field@link{#1}{#2}{\gls@text@accessdisplay{\glsentrytext{#2}}{#2}#3}%
11754 }

```

\@Gls@text@

```

11755 \def\@Gls@text@#1#2[#3]{%
11756 \gls@field@link{#1}{#2}{\Gls@text@accessdisplay{\Glsentrytext{#2}}{#2}#3}%
11757 }

```

\@GLStext@

```

11758 \def\@GLStext@#1#2[#3]{%
11759 \gls@field@link{#1}{#2}%
11760 {\gls@text@accessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}{#2}%
11761 \mfirstucMakeUppercase{#3}}}%
11762 }

```

```

\@glsfirst@
11763 \def \@glsfirst@#1#2[#3]{%
11764   \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glentryfirst{#2}}{#2}#3}%
11765 }

\@Glsfirst@
11766 \def \@Glsfirst@#1#2[#3]{%
11767   \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
11768 }

\@GLSfirst@
11769 \def \@GLSfirst@#1#2[#3]{%
11770   \@gls@field@link{#1}{#2}%
11771   {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glentryfirst{#2}}}{#2}%
11772     \mfirstucMakeUppercase{#3}}%
11773 }

\@glsplural@
11774 \def \@glsplural@#1#2[#3]{%
11775   \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glentryplural{#2}}{#2}#3}%
11776 }

\@Glsplural@
11777 \def \@Glsplural@#1#2[#3]{%
11778   \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
11779 }

\@GLSplural@
11780 \def \@GLSplural@#1#2[#3]{%
11781   \@gls@field@link{#1}{#2}%
11782   {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glentryplural{#2}}}{#2}%
11783     \mfirstucMakeUppercase{#3}}%
11784 }

glsfirstplural@
11785 \def \@glsfirstplural@#1#2[#3]{%
11786   \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glentryfirstplural{#2}}{#2}#3}%
11787 }

Glsfirstplural@
11788 \def \@glsfirstplural@#1#2[#3]{%
11789   \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
11790 }

GLSfirstplural@
11791 \def \@GLSfirstplural@#1#2[#3]{%
11792   \@gls@field@link{#1}{#2}%
11793   {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glentryfirstplural{#2}}}{#2}%

```

```

11794     \mfirstucMakeUppercase{#3}}}%
11795 }

\@glsname@
11796 \def\@glsname@#1#2[#3]{%
11797   \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
11798 }

\@Glsname@
11799 \def\@Glsname@#1#2[#3]{%
11800   \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
11801 }

\@GLSname@
11802 \def\@GLSname@#1#2[#3]{%
11803   \@gls@field@link{#1}{#2}%
11804   {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
11805     \mfirstucMakeUppercase{#3}}}%
11806 }

\@glsdesc@
11807 \def\@glsdesc@#1#2[#3]{%
11808   \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
11809 }

\@Glsdesc@
11810 \def\@Glsdesc@#1#2[#3]{%
11811   \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
11812 }

\@GLSdesc@
11813 \def\@GLSdesc@#1#2[#3]{%
11814   \@gls@field@link{#1}{#2}%
11815   {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
11816     \mfirstucMakeUppercase{#3}}}%
11817 }

@glsdescplural@
11818 \def\@glsdescplural@#1#2[#3]{%
11819   \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
11820 }

@Glsdescplural@
11821 \def\@Glsdescplural@#1#2[#3]{%
11822   \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
11823 }

```

@GLSdescplural@

```
11824 \def\@GLSdescplural@#1#2[#3]{%
11825   \@gls@field@link{#1}{#2}%
11826   {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
11827     \mfirstucMakeUppercase{#3}}%
11828 }
```

\@glssymbol@

```
11829 \def\@glssymbol@#1#2[#3]{%
11830   \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
11831 }
```

\@GLssymbol@

```
11832 \def\@GLssymbol@#1#2[#3]{%
11833   \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
11834 }
```

\@GLSsymbol@

```
11835 \def\@GLSsymbol@#1#2[#3]{%
11836   \@gls@field@link{#1}{#2}%
11837   {\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
11838     \mfirstucMakeUppercase{#3}}%
11839 }
```

lssymbolplural@

```
11840 \def\@glssymbolplural@#1#2[#3]{%
11841   \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
11842 }
```

lssymbolplural@

```
11843 \def\@Glssymbolplural@#1#2[#3]{%
11844   \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
11845 }
```

LSymbolplural@

```
11846 \def\@GLSsymbolplural@#1#2[#3]{%
11847   \@gls@field@link{#1}{#2}%
11848   {\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
11849     \mfirstucMakeUppercase{#3}}%
11850 }
```

\@glsuseri@

```
11851 \def\@glsuseri@#1#2[#3]{%
11852   \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
11853 }
```

\@Glsuseri@

```
11854 \def \@Glsuser@i#1#2[#3]{%
11855   \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
11856 }
```

\@GLSuseri@

```
11857 \def \@GLSuseri@#1#2[#3]{%
11858   \@gls@field@link{#1}{#2}%
11859   {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
11860    \mfirstucMakeUppercase{#3}}%
11861 }
```

\@glsuserii@

```
11862 \def \@glsuserii@#1#2[#3]{%
11863   \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
11864 }
```

\@Glsuserii@

```
11865 \def \@Glsuser@i#1#2[#3]{%
11866   \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
11867 }
```

\@GLSuserii@

```
11868 \def \@GLSuserii@#1#2[#3]{%
11869   \@gls@field@link{#1}{#2}%
11870   {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
11871    \mfirstucMakeUppercase{#3}}%
11872 }
```

\@glsuseriii@

```
11873 \def \@glsuseriii@#1#2[#3]{%
11874   \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
11875 }
```

\@Glsuseriii@

```
11876 \def \@Glsuser@i#1#2[#3]{%
11877   \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
11878 }
```

\@GLSuseriii@

```
11879 \def \@GLSuseriii@#1#2[#3]{%
11880   \@gls@field@link{#1}{#2}%
11881   {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
11882    \mfirstucMakeUppercase{#3}}%
11883 }
```



```

\@glsuseriv@
11884 \def \@glsuseriv@#1#2[#3]{%
11885   \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
11886 }

\@Glsuseriv@
11887 \def \@Glsuser@i#1#2[#3]{%
11888   \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
11889 }

\@GLSuseriv@
11890 \def \@GLSuseriv@#1#2[#3]{%
11891   \@gls@field@link{#1}{#2}%
11892   {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}{#2}%
11893     \mfirstucMakeUppercase{#3}}}%
11894 }

\@glsuserv@
11895 \def \@glsuserv@#1#2[#3]{%
11896   \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
11897 }

\@Glsuserv@
11898 \def \@Glsuser@i#1#2[#3]{%
11899   \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
11900 }

\@GLSuserv@
11901 \def \@GLSuserv@#1#2[#3]{%
11902   \@gls@field@link{#1}{#2}%
11903   {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}{#2}%
11904     \mfirstucMakeUppercase{#3}}}%
11905 }

\@glsuservi@
11906 \def \@glsuservi@#1#2[#3]{%
11907   \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
11908 }

\@Glsuservi@
11909 \def \@Glsuser@i#1#2[#3]{%
11910   \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
11911 }

\@GLSuservi@
11912 \def \@GLSuservi@#1#2[#3]{%
11913   \@gls@field@link{#1}{#2}%
11914   {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#2}}{#2}%
11915     \mfirstucMakeUppercase{#3}}}%
11916 }

```

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.

```
11917 \renewcommand*{\glossentryname}[1]{%
11918   \glsdoifexists{#1}%
11919   {%
11920     \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
11921   }%
11922 }

11923 \renewcommand*{\glossentryname}[1]{%
11924   \glsdoifexists{#1}%
11925   {%
11926     \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
11927   }%
11928 }

11929 \renewcommand*{\glossentrydesc}[1]{%
11930   \glsdoifexists{#1}%
11931   {%
11932     \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
11933   }%
11934 }

11935 \renewcommand*{\Glossentrydesc}[1]{%
11936   \glsdoifexists{#1}%
11937   {%
11938     \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
11939   }%
11940 }

11941 \renewcommand*{\glossentrysymbol}[1]{%
11942   \glsdoifexists{#1}%
11943   {%
11944     \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
11945   }%
11946 }

11947 \renewcommand*{\Glossentrysymbol}[1]{%
11948   \glsdoifexists{#1}%
11949   {%
11950     \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
11951   }%
11952 }
```

ssaryentryfield

```

11953 \newcommand*{\accsuppglossaryentryfield}[5]{%
11954   \glossaryentryfield{#1}%
11955   {\glslnameaccessdisplay{#2}{#1}}%
11956   {\glsdescriptionaccessdisplay{#3}{#1}}%
11957   {\glssymbolaccessdisplay{#4}{#1}}{#5}%
11958 }

```

rysubentryfield

```

11959 \newcommand*{\accsuppglossarysubentryfield}[6]{%
11960   \glossarysubentryfield{#1}{#2}%
11961   {\glslnameaccessdisplay{#3}{#2}}%
11962   {\glsdescriptionaccessdisplay{#4}{#2}}%
11963   {\glssymbolaccessdisplay{#5}{#2}}{#6}%
11964 }

```

5.4 Acronyms

Redefine acronym styles provided by glossaries:

long-short *<long>* (*<short>*) acronym style.

```

11965 \renewacronymstyle{long-short}%
11966 {}%

```

Check for long form in case this is a mixed glossary.

```

11967 \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
11968 }%
11969 {}%
11970 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
11971 \renewcommand*{\genacrfullformat}[2]{%
11972   \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
11973   (\glsshortaccessdisplay
11974     {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
11975   }%
11976 \renewcommand*{\Genacrfullformat}[2]{%
11977   \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
11978   (\glsshortaccessdisplay
11979     {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
11980   }%
11981 \renewcommand*{\genplacrfullformat}[2]{%
11982   \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
11983   (\glsshortpluralaccessdisplay
11984     {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
11985   }%
11986 \renewcommand*{\Genplacrfullformat}[2]{%
11987   \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
11988   (\glsshortpluralaccessdisplay
11989     {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
11990   }%
11991 \renewcommand*{\acronymentry}[1]{%

```

```

11992 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
11993 \renewcommand*{\acronymsort}[2]{##1}%
11994 \renewcommand*{\acronymfont}[1]{##1}%
11995 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
11996 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
11997 }

```

short-long *<short>* (*<long>*) acronym style.

```

11998 \renewacronymstyle{short-long}%
11999 {%

```

Check for long form in case this is a mixed glossary.

```

12000 \ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
12001 }%
12002 {%
12003 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
12004 \renewcommand*{\genacrfullformat}[2]{%
12005 \glsshortaccessdisplay
12006 {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
12007 (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
12008 }%
12009 \renewcommand*{\Genacrfullformat}[2]{%
12010 \glsshortaccessdisplay
12011 {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
12012 (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
12013 }%
12014 \renewcommand*{\genplacrfullformat}[2]{%
12015 \glsshortpluralaccessdisplay
12016 {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
12017 (\glslongpluralaccessdisplay
12018 {\glentrylongpl{##1}}{##1})%
12019 }%
12020 \renewcommand*{\Genplacrfullformat}[2]{%
12021 \glsshortpluralaccessdisplay
12022 {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
12023 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
12024 }%
12025 \renewcommand*{\acronymentry}[1]{%
12026 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
12027 \renewcommand*{\acronymsort}[2]{##1}%
12028 \renewcommand*{\acronymfont}[1]{##1}%
12029 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
12030 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
12031 }

```

long-short-desc *<long>* (*<short>*) acronym style that has an accompanying description (which the user needs to supply).

```

12032 \renewacronymstyle{long-short-desc}%
12033 {%

```

```

12034 \GlsUseAcrEntryDisplayStyle{long-short}%
12035 }%
12036 {%
12037 \GlsUseAcrStyleDefs{long-short}%
12038 \renewcommand*{\GenericAcronymFields}{}%
12039 \renewcommand*{\acronymsort}[2]{##2}%
12040 \renewcommand*{\acronymentry}[1]{%
12041 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
12042 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1)}}%
12043 }

```

g-sc-short-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12044 \renewacronymstyle{long-sc-short-desc}%
12045 {%
12046 \GlsUseAcrEntryDisplayStyle{long-sc-short}%
12047 }%
12048 {%
12049 \GlsUseAcrStyleDefs{long-sc-short}%
12050 \renewcommand*{\GenericAcronymFields}{}%
12051 \renewcommand*{\acronymsort}[2]{##2}%
12052 \renewcommand*{\acronymentry}[1]{%
12053 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
12054 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1)}}%
12055 }

```

g-sm-short-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12056 \renewacronymstyle{long-sm-short-desc}%
12057 {%
12058 \GlsUseAcrEntryDisplayStyle{long-sm-short}%
12059 }%
12060 {%
12061 \GlsUseAcrStyleDefs{long-sm-short}%
12062 \renewcommand*{\GenericAcronymFields}{}%
12063 \renewcommand*{\acronymsort}[2]{##2}%
12064 \renewcommand*{\acronymentry}[1]{%
12065 \glslongaccessdisplay{\gl Sentrylong{##1}}{##1}\space
12066 (\glsshortaccessdisplay{\acronymfont{\gl Sentryshort{##1}}}{##1)}}%
12067 }

```

short-long-desc *<short>* ({<long>}) acronym style that has an accompanying description (which the user needs to supply).

```

12068 \renewacronymstyle{short-long-desc}%
12069 {%
12070 \GlsUseAcrEntryDisplayStyle{short-long}%
12071 }%
12072 {%
12073 \GlsUseAcrStyleDefs{short-long}%

```

```

12074 \renewcommand*{\GenericAcronymFields}{}%
12075 \renewcommand*{\acronymsort}[2]{##2}%
12076 \renewcommand*{\acronymentry}[1]{%
12077     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12078     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
12079 }

```

short-long-desc *<long>* (\textsc{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12080 \renewacronymstyle{sc-short-long-desc}%
12081 {%
12082     \GlsUseAcrEntryDispStyle{sc-short-long}%
12083 }%
12084 {%
12085     \GlsUseAcrStyleDefs{sc-short-long}%
12086     \renewcommand*{\GenericAcronymFields}{}%
12087     \renewcommand*{\acronymsort}[2]{##2}%
12088     \renewcommand*{\acronymentry}[1]{%
12089         \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12090         (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
12091     }

```

short-long-desc *<long>* (\textsmaller{<short>}) acronym style that has an accompanying description (which the user needs to supply).

```

12092 \renewacronymstyle{sm-short-long-desc}%
12093 {%
12094     \GlsUseAcrEntryDispStyle{sm-short-long}%
12095 }%
12096 {%
12097     \GlsUseAcrStyleDefs{sm-short-long}%
12098     \renewcommand*{\GenericAcronymFields}{}%
12099     \renewcommand*{\acronymsort}[2]{##2}%
12100     \renewcommand*{\acronymentry}[1]{%
12101         \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12102         (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
12103     }

```

dua *<long>* only acronym style.

```

12104 \renewacronymstyle{dua}%
12105 {%

```

Check for long form in case this is a mixed glossary.

```

12106 \ifdefempty\glscustomtext
12107 {%
12108     \ifglshaslong{\glslabel}%
12109     {%
12110         \glsifplural
12111         {%

```

Plural form:

12112 \glscapscase
12113 {%

Plural form, don't adjust case:

12114 \glslongpluralaccessdisplay{\glentrylongpl{\glslabel}}{\glslabel}%
12115 \glsinsert
12116 }%
12117 {%

Plural form, make first letter upper case:

12118 \glslongpluralaccessdisplay{\Glentrylongpl{\glslabel}}{\glslabel}%
12119 \glsinsert
12120 }%
12121 {%

Plural form, all caps:

12122 \glslongpluralaccessdisplay
12123 {\mfirstucMakeUppercase{\glentrylongpl{\glslabel}}}{\glslabel}%
12124 \mfirstucMakeUppercase{\glsinsert}%
12125 }%
12126 }%
12127 {%

Singular form

12128 \glscapscase
12129 {%

Singular form, don't adjust case:

12130 \glslongaccessdisplay{\glentrylong{\glslabel}}{\glslabel}\glsinsert
12131 }%
12132 {%

Subsequent singular form, make first letter upper case:

12133 \glslongaccessdisplay{\Glentrylong{\glslabel}}{\glslabel}\glsinsert
12134 }%
12135 {%

Subsequent singular form, all caps:

12136 \glslongaccessdisplay
12137 {\mfirstucMakeUppercase
12138 {\glentrylong{\glslabel}\glsinsert}}{\glslabel}%
12139 \mfirstucMakeUppercase{\glsinsert}%
12140 }%
12141 }%
12142 }%
12143 {%

Not an acronym:

12144 \glsgenentryfmt
12145 }%
12146 }%

```

12147 {\glscustomtext\glsinsert}%
12148 }%
12149 {%
12150 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
12151 \renewcommand*{\acrfullfmt}[3]{%
12152   \glslink[##1]{##2}{%
12153     \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
12154     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
12155 \renewcommand*{\Acrfullfmt}[3]{%
12156   \glslink[##1]{##2}{%
12157     \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
12158     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
12159 \renewcommand*{\ACRfullfmt}[3]{%
12160   \glslink[##1]{##2}{%
12161     \glslongaccessdisplay
12162     {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
12163     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}}%
12164 \renewcommand*{\acrfullplfmt}[3]{%
12165   \glslink[##1]{##2}{%
12166     \glslongpluralaccessdisplay
12167     {\glsentrylongpl{##2}}{##2}##3\space
12168     (\glsshortpluralaccessdisplay
12169     {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
12170 \renewcommand*{\Acrfullplfmt}[3]{%
12171   \glslink[##1]{##2}{%
12172     \glslongpluralaccessdisplay
12173     {\Glsentrylongpl{##2}}{##2}##3\space
12174     (\glsshortpluralaccessdisplay
12175     {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
12176 \renewcommand*{\ACRfullplfmt}[3]{%
12177   \glslink[##1]{##2}{%
12178     \glslongpluralaccessdisplay
12179     {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
12180     (\glsshortpluralaccessdisplay
12181     {\acronymfont{\glsentryshortpl{##2}}}{##2})}}}%
12182 \renewcommand*{\glsentryfull}[1]{%
12183   \glslongaccessdisplay{\glsentrylong{##1}}\space
12184   (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
12185 }%
12186 \renewcommand*{\Glsentryfull}[1]{%
12187   \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
12188   (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
12189 }%
12190 \renewcommand*{\glsentryfullpl}[1]{%
12191   \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
12192   (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
12193 }%
12194 \renewcommand*{\Glsentryfullpl}[1]{%
12195   \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space

```



```

12196 (\glsshortpluralaccessdisplay{\acronymfont{\glentryshortpl{##1}}}{##1})%
12197 }%
12198 \renewcommand*{\acronymentry}[1]{%
12199 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
12200 \renewcommand*{\acronymsort}[2]{##1}%
12201 \renewcommand*{\acronymfont}[1]{##1}%
12202 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
12203 }

```

dua-desc *<long>* only acronym style with user-supplied description.

```

12204 \renewacronymstyle{dua-desc}%
12205 {%
12206 \GlsUseAcrEntryDispStyle{dua}%
12207 }%
12208 {%
12209 \GlsUseAcrStyleDefs{dua}%
12210 \renewcommand*{\GenericAcronymFields}{}%
12211 \renewcommand*{\acronymentry}[1]{%
12212 \glslongaccessdisplay{\acronymfont{\glentrylong{##1}}}{##1}}%
12213 \renewcommand*{\acronymsort}[2]{##2}%
12214 }%

```

footnote *<short>*\footnote{*<long>*} acronym style.

```

12215 \renewacronymstyle{footnote}%
12216 {%
12217 \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
12218 }%
12219 {%
12220 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

```

Need to ensure hyperlinks are switched off on first use:

```

12221 \glshyperfirstfalse
12222 \renewcommand*{\genacrfullformat}[2]{%
12223 \glsshortaccessdisplay
12224 {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2%
12225 \protect\footnote{\glslongaccessdisplay{\glentrylong{##1}}}{##1}}%
12226 }%
12227 \renewcommand*{\Genacrfullformat}[2]{%
12228 \glsshortaccessdisplay
12229 {\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
12230 \protect\footnote{\glslongaccessdisplay{\glentrylong{##1}}}{##1}}%
12231 }%
12232 \renewcommand*{\genplacrfullformat}[2]{%
12233 \glsshortpluralaccessdisplay
12234 {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2%
12235 \protect\footnote{\glslongpluralaccessdisplay{\glentrylongpl{##1}}}{##1}}%
12236 }%
12237 \renewcommand*{\Genplacrfullformat}[2]{%

```

```

12238 \glsshortpluralaccessdisplay
12239 {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
12240 \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
12241 }%
12242 \renewcommand*{\acronymentry}[1]{%
12243 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
12244 \renewcommand*{\acronymsort}[2]{##1}%
12245 \renewcommand*{\acronymfont}[1]{##1}%
12246 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%

```

Don't use footnotes for \acrfull:

```

12247 \renewcommand*{\acrfullfmt}[3]{%
12248 \glslink[##1]{##2}{%
12249 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
12250 (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
12251 \renewcommand*{\Acrfullfmt}[3]{%
12252 \glslink[##1]{##2}{%
12253 \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
12254 (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
12255 \renewcommand*{\ACRfullfmt}[3]{%
12256 \glslink[##1]{##2}{%
12257 \glsshortaccessdisplay
12258 {\mfirstucMakeUppercase
12259 {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
12260 (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}}%
12261 \renewcommand*{\acrfullplfmt}[3]{%
12262 \glslink[##1]{##2}{%
12263 \glsshortpluralaccessdisplay
12264 {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
12265 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
12266 \renewcommand*{\ACRfullplfmt}[3]{%
12267 \glslink[##1]{##2}{%
12268 \glsshortpluralaccessdisplay
12269 {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
12270 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
12271 \renewcommand*{\ACRfullplplfmt}[3]{%
12272 \glslink[##1]{##2}{%
12273 \glsshortpluralaccessdisplay
12274 {\mfirstucMakeUppercase
12275 {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
12276 (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%

```

Similarly for \glsentryfull etc:

```

12277 \renewcommand*{\glsentryfull}[1]{%
12278 \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
12279 (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
12280 \renewcommand*{\Glsentryfull}[1]{%
12281 \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
12282 (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
12283 \renewcommand*{\glsentryfullpl}[1]{%

```

```

12284 \glsshortpluralaccessdisplay
12285 {\acronymfont{\glentryshortpl{##1}}{##1}\space
12286 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}}%
12287 \renewcommand*\Glsentryfullpl[1]{%
12288 \glsshortpluralaccessdisplay
12289 {\acronymfont{\Glsentryshortpl{##1}}{##1}\space
12290 (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}}%
12291 }

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

12292 \renewacronymstyle{footnote-sc}%
12293 {%
12294 \GlsUseAcrEntryDisplayStyle{footnote}%
12295 }%
12296 {%
12297 \GlsUseAcrStyleDefs{footnote}%
12298 \renewcommand{\acronymentry}[1]{%
12299 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}
12300 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
12301 \renewcommand*\acrpluralsuffix{\glstextup{\glspluralsuffix}}%
12302 }%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```

12303 \renewacronymstyle{footnote-sm}%
12304 {%
12305 \GlsUseAcrEntryDisplayStyle{footnote}%
12306 }%
12307 {%
12308 \GlsUseAcrStyleDefs{footnote}%
12309 \renewcommand{\acronymentry}[1]{%
12310 \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}
12311 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
12312 \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
12313 }%

```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

12314 \renewacronymstyle{footnote-desc}%
12315 {%
12316 \GlsUseAcrEntryDisplayStyle{footnote}%
12317 }%
12318 {%
12319 \GlsUseAcrStyleDefs{footnote}%
12320 \renewcommand*\GenericAcronymFields{}%
12321 \renewcommand*\acronymsort[2]{##2}%
12322 \renewcommand*\acronymentry[1]{%
12323 \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
12324 (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}}%
12325 }

```

ootnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

12326 \renewacronymstyle{footnote-sc-desc}%
12327 {%
12328   \GlsUseAcrEntryDispStyle{footnote-sc}%
12329 }%
12330 {%
12331   \GlsUseAcrStyleDefs{footnote-sc}%
12332   \renewcommand*{\GenericAcronymFields}{}%
12333   \renewcommand*{\acronymsort}[2]{##2}%
12334   \renewcommand*{\acronymentry}[1]{%
12335     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12336     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1)}}%
12337 }

```

ootnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

12338 \renewacronymstyle{footnote-sm-desc}%
12339 {%
12340   \GlsUseAcrEntryDispStyle{footnote-sm}%
12341 }%
12342 {%
12343   \GlsUseAcrStyleDefs{footnote-sm}%
12344   \renewcommand*{\GenericAcronymFields}{}%
12345   \renewcommand*{\acronymsort}[2]{##2}%
12346   \renewcommand*{\acronymentry}[1]{%
12347     \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
12348     (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1)}}%
12349 }

```

aultshortaccess \glsdefaultshortaccess{<long>}{<short>}

Default shortaccess value.

```
12350 \newcommand*{\glsdefaultshortaccess}[2]{#1}
```

Use \newacronymhook to modify the key list to set the access text to the long version by default.

```

12351 \renewcommand*{\newacronymhook}{%
12352   \protected@edef\@gls@keylist{%
12353     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12354     shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12355     \the\glskeylisttok}%
12356   \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
12357 }

```

ltNewAcronymDef Modify default style to use access text:

```
12358 \renewcommand*{\DefaultNewAcronymDef}{%
```

```

12359 \edef\@do@newglossaryentry{%
12360   \noexpand\newglossaryentry{\the\glslabeltok}%
12361   {%
12362     type=\acronymtype,%
12363     name={\the\glsshorttok},%
12364     description={\the\glslongtok},%
12365     descriptionaccess=\relax,
12366     text={\the\glsshorttok},%
12367     access={\noexpand\@glo@textaccess},%
12368     sort={\the\glsshorttok},%
12369     short={\the\glsshorttok},%
12370     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12371     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12372     long={\the\glslongtok},%
12373     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12374     descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12375     first={\noexpand\glslongaccessdisplay
12376       {\the\glslongtok}{\the\glslabeltok}\space
12377       (\noexpand\glsshortaccessdisplay
12378         {\the\glsshorttok}{\the\glslabeltok})},%
12379     plural={\the\glsshorttok\acrpluralsuffix},%
12380     firstplural={\noexpand\glslongpluralaccessdisplay
12381       {\noexpand\@glo@longpl}{\the\glslabeltok}\space
12382       (\noexpand\glsshortpluralaccessdisplay
12383         {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
12384     firstaccess=\relax,
12385     firstpluralaccess=\relax,
12386     textaccess={\noexpand\@glo@shortaccess},%
12387     \the\glskeylisttok
12388   }%
12389 }%
12390 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12391 \let\@org@gls@assign@plural\gls@assign@plural
12392 \let\@org@gls@assign@descplural\gls@assign@descplural
12393 \def\gls@assign@firstpl##1##2{%
12394   \@gls@expand@field{##1}{firstpl}{##2}%
12395 }%
12396 \def\gls@assign@plural##1##2{%
12397   \@gls@expand@field{##1}{plural}{##2}%
12398 }%
12399 \def\gls@assign@descplural##1##2{%
12400   \@gls@expand@field{##1}{descplural}{##2}%
12401 }%
12402 \@do@newglossaryentry
12403 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12404 \let\gls@assign@plural\@org@gls@assign@plural
12405 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12406 \@gls@noexpand@assign@default@plurals
12407 }

```

teNewAcronymDef

```

12408 \renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
12409   \edef\@do@newglossaryentry{%
12410     \noexpand\newglossaryentry{\the\glslabeltok}%
12411     {%
12412       type=\acronymtype,%
12413       name={\noexpand\acronymfont{\the\glsshorttok}},%
12414       sort={\the\glsshorttok},%
12415       text={\the\glsshorttok},%
12416       short={\the\glsshorttok},%
12417       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12418       shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12419       long={\the\glslongtok},%
12420       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12421       access={\noexpand\@glo@textaccess},%
12422       plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12423       symbol={\the\glslongtok},%
12424       symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12425       firstpluralaccess=\relax,
12426       textaccess={\noexpand\@glo@shortaccess},%
12427       \the\glskeylisttok
12428     }%
12429   }%
12430   \let\@org@gls@assign@firstpl\gls@assign@firstpl
12431   \let\@org@gls@assign@plural\gls@assign@plural
12432   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12433   \def\gls@assign@firstpl##1##2{%
12434     \@@gls@expand@field{##1}{firstpl}{##2}%
12435   }%
12436   \def\gls@assign@plural##1##2{%
12437     \@@gls@expand@field{##1}{plural}{##2}%
12438   }%
12439   \def\gls@assign@symbolplural##1##2{%
12440     \@@gls@expand@field{##1}{symbolplural}{##2}%
12441   }%
12442   \@do@newglossaryentry
12443   \let\gls@assign@plural\@org@gls@assign@plural
12444   \let\gls@assign@firstpl\@org@gls@assign@firstpl
12445   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12446 }

```

onNewAcronymDef

```

12447 \renewcommand*{\DescriptionNewAcronymDef}{%
12448   \edef\@do@newglossaryentry{%
12449     \noexpand\newglossaryentry{\the\glslabeltok}%
12450     {%
12451       type=\acronymtype,%
12452       name={\noexpand
12453         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%

```

```

12454     access={\noexpand\@glo@textaccess},%
12455     sort={\the\glsshorttok},%
12456     short={\the\glsshorttok},%
12457     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12458     shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12459     long={\the\glslongtok},%
12460     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12461     first={\the\glslongtok},%
12462     firstaccess=\relax,
12463     firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12464     text={\the\glsshorttok},%
12465     textaccess={\the\glslongtok},%
12466     plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12467     symbol={\noexpand\@glo@text},%
12468     symbolaccess={\noexpand\@glo@textaccess},%
12469     symbolplural={\noexpand\@glo@plural},%
12470     firstpluralaccess=\relax,
12471     textaccess={\noexpand\@glo@shortaccess},%
12472     \the\glskeylisttok}%
12473 }%
12474 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12475 \let\@org@gls@assign@plural\gls@assign@plural
12476 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12477 \def\gls@assign@firstpl##1##2{%
12478   \@gls@expand@field{##1}{firstpl}{##2}%
12479 }%
12480 \def\gls@assign@plural##1##2{%
12481   \@gls@expand@field{##1}{plural}{##2}%
12482 }%
12483 \def\gls@assign@symbolplural##1##2{%
12484   \@gls@expand@field{##1}{symbolplural}{##2}%
12485 }%
12486 \do@newglossaryentry
12487 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12488 \let\gls@assign@plural\@org@gls@assign@plural
12489 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12490 }

```

teNewAcronymDef

```

12491 \renewcommand*{\FootnoteNewAcronymDef}{%
12492   \edef\@do@newglossaryentry{%
12493     \noexpand\newglossaryentry{\the\glslabeltok}%
12494     {%
12495       type=\acronymtype,%
12496       name={\noexpand\acronymfont{\the\glsshorttok}},%
12497       sort={\the\glsshorttok},%
12498       text={\the\glsshorttok},%
12499       textaccess={\the\glslongtok},%
12500       access={\noexpand\@glo@textaccess},%

```

```

12501 plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12502 short={\the\glsshorttok},%
12503 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12504 long={\the\glslongtok},%
12505 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12506 description={\the\glslongtok},%
12507 descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12508 \the\glskeylisttok
12509 }%
12510 }%
12511 \let\@org@gls@assign@plural\gls@assign@plural
12512 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12513 \let\@org@gls@assign@descplural\gls@assign@descplural
12514 \def\gls@assign@firstpl##1##2{%
12515   \@@gls@expand@field{##1}{firstpl}{##2}%
12516 }%
12517 \def\gls@assign@plural##1##2{%
12518   \@@gls@expand@field{##1}{plural}{##2}%
12519 }%
12520 \def\gls@assign@descplural##1##2{%
12521   \@@gls@expand@field{##1}{descplural}{##2}%
12522 }%
12523 \do@newglossaryentry
12524 \let\gls@assign@plural\@org@gls@assign@plural
12525 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12526 \let\gls@assign@descplural\@org@gls@assign@descplural
12527 }

```

11NewAcronymDef

```

12528 \renewcommand*{\SmallNewAcronymDef}{%
12529   \edef\@do@newglossaryentry{%
12530     \noexpand\newglossaryentry{\the\glslabeltok}%
12531     {%
12532       type=\acronymtype,%
12533       name={\noexpand\acronymfont{\the\glsshorttok}},%
12534       access={\noexpand\@glo@symbolaccess},%
12535       sort={\the\glsshorttok},%
12536       short={\the\glsshorttok},%
12537       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12538       shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
12539       long={\the\glslongtok},%
12540       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12541       text={\noexpand\@glo@short},%
12542       textaccess={\noexpand\@glo@shortaccess},%
12543       plural={\noexpand\@glo@shortpl},%
12544       first={\the\glslongtok},%
12545       firstaccess=\relax,%
12546       firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
12547       description={\noexpand\@glo@first},%

```



```

12548     descriptionplural={\noexpand\@glo@firstplural},%
12549     symbol={\the\glsshorttok},%
12550     symbolaccess={\the\glslongtok},%
12551     symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
12552     \the\glskeylisttok
12553   }%
12554 }%
12555 \let\@org@gls@assign@firstpl\gls@assign@firstpl
12556 \let\@org@gls@assign@plural\gls@assign@plural
12557 \let\@org@gls@assign@descplural\gls@assign@descplural
12558 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
12559 \def\gls@assign@firstpl##1##2{%
12560   \@@gls@expand@field{##1}{firstpl}{##2}%
12561 }%
12562 \def\gls@assign@plural##1##2{%
12563   \@@gls@expand@field{##1}{plural}{##2}%
12564 }%
12565 \def\gls@assign@descplural##1##2{%
12566   \@@gls@expand@field{##1}{descplural}{##2}%
12567 }%
12568 \def\gls@assign@symbolplural##1##2{%
12569   \@@gls@expand@field{##1}{symbolplural}{##2}%
12570 }%
12571 \do@newglossaryentry
12572 \let\gls@assign@firstpl\@org@gls@assign@firstpl
12573 \let\gls@assign@plural\@org@gls@assign@plural
12574 \let\gls@assign@descplural\@org@gls@assign@descplural
12575 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
12576 }

```

The following are kept for compatibility with versions before 3.0:

sshortaccesskey

```
12577 \newcommand*{\glsshortaccesskey}{\glsshortkey access}%
```

pluralaccesskey

```
12578 \newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
```

lslongaccesskey

```
12579 \newcommand*{\glslongaccesskey}{\glslongkey access}%
```

pluralaccesskey

```
12580 \newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

5.5 Debugging Commands

owglonameaccess

```
12581 \newcommand*{\showglonameaccess}[1]{%
```

```

12582 \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
12583 }

```

owglotextaccess

```

12584 \newcommand*{\showglotextaccess}[1]{%
12585 \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
12586 }

```

glopluralaccess

```

12587 \newcommand*{\showglopluralaccess}[1]{%
12588 \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
12589 }

```

wglofirstaccess

```

12590 \newcommand*{\showwglofirstaccess}[1]{%
12591 \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
12592 }

```

rstpluralaccess

```

12593 \newcommand*{\showglofirstpluralaccess}[1]{%
12594 \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
12595 }

```

glosymbolaccess

```

12596 \newcommand*{\showglosymbolaccess}[1]{%
12597 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
12598 }

```

bolpluralaccess

```

12599 \newcommand*{\showglosymbolpluralaccess}[1]{%
12600 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
12601 }

```

owglodescaccess

```

12602 \newcommand*{\showglodescaccess}[1]{%
12603 \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
12604 }

```

escpluralaccess

```

12605 \newcommand*{\showglodescpluralaccess}[1]{%
12606 \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
12607 }

```

wgloshortaccess

```

12608 \newcommand*{\showgloshortaccess}[1]{%
12609 \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
12610 }

```

ortpluralaccess

```
12611 \newcommand*{\showgloshortpluralaccess}[1]{%  
12612   \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname  
12613 }
```

owglolongaccess

```
12614 \newcommand*{\showglolongaccess}[1]{%  
12615   \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname  
12616 }
```

ongpluralaccess

```
12617 \newcommand*{\showglolongpluralaccess}[1]{%  
12618   \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname  
12619 }
```

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.

```
12620 \NeedsTeXFormat{LaTeX2e}
12621 \ProvidesPackage{glossaries-babel}[2021/10/19 v4.48 (NLCT)]
```

Load tracklang to obtain language settings.

```
12622 \RequirePackage{tracklang}
12623 \let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
12624 \AnyTrackedLanguages
12625 {%
12626   \ForEachTrackedDialect{\this@dialect}{%
12627     \IfTrackedLanguageFileExists{\this@dialect}%
12628       {glossaries-}% prefix
12629       {.ldf}%
12630       {%
12631         \RequireGlossariesLang{\CurrentTrackedTag}%
12632       }%
12633       {%
12634         \PackageWarningNoLine{glossaries}%
12635           {No language module detected for ‘\this@dialect’.\MessageBreak
12636             Language modules need to be installed separately.\MessageBreak
12637             Please check on CTAN for a bundle called\MessageBreak
12638             ‘glossaries-\CurrentTrackedLanguage’ or similar}%
12639       }%
12640     }%
12641   }%
12642 }
```

6.1 Polyglossia Captions

Language support has now been split off into independent language modules.

```
12643 \NeedsTeXFormat{LaTeX2e}
12644 \ProvidesPackage{glossaries-polyglossia}[2021/10/19 v4.48 (NLCT)]
```

Load tracklang to obtain language settings.

```
12645 \RequirePackage{tracklang}
12646 \let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
12647 \AnyTrackedLanguages
```

```

12648 {%
12649   \ForEachTrackedDialect{\this@dialect}{%
12650     \IfTrackedLanguageFileExists{\this@dialect}%
12651     {glossaries-}% prefix
12652     {.ldf}%
12653     {%
12654       \RequireGlossariesLang{\CurrentTrackedTag}%
12655     }%
12656     {%
12657       \PackageWarningNoLine{glossaries}%
12658       {No language module detected for ‘\this@dialect’.\MessageBreak
12659       Language modules need to be installed separately.\MessageBreak
12660       Please check on CTAN for a bundle called\MessageBreak
12661       ‘glossaries-\CurrentTrackedLanguage’ or similar}%
12662     }%
12663   }%
12664 }%
12665 {}%

```

6.2 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

12666 \NeedsTeXFormat{LaTeX2e}
12667 \ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]
12668 \RequirePackage{ifthen}
12669 \RequirePackage{xkeyval}[2006/11/18]
12670 \RequirePackage{mfirstuc}
12671 \RequirePackage{textcase}
12672 \renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
12673 \RequirePackage{xfor}
12674 \RequirePackage{datatool-base}
12675 \RequirePackage{amsgen}
12676 \RequirePackage{etoolbox}
12677 \newif\if@gls@docloaded
12678 \@ifpackageloaded{doc}%
12679 {%
12680   \@gls@docloadedtrue
12681 }%
12682 {%
12683   \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
12684 }
12685 \if@gls@docloaded
12686   \let\glsorg@theglossary\theglossary
12687   \let\glsorg@endtheglossary\endtheglossary
12688   \let\glsorg@PrintChanges\PrintChanges
12689   \renewcommand{\PrintChanges}{%
12690     \begingroup
12691     \let\theglossary\glsorg@theglossary

```

```

12692     \let\endtheglossary\glsorg@endtheglossary
12693     \glsorg@PrintChanges
12694 \endgroup
12695 }
12696 \fi
12697 \newif\if@gls@debug
12698 \define@choicekey{glossaries.sty}{debug}{\@gls@debug@val\@gls@debug@nr}%
12699 {true,false,showtargets,showaccsupp}[true]{%
12700   \ifcase\@gls@debug@nr\relax
12701     % debug=true
12702     \@gls@debugtrue
12703     \renewcommand*\GlossariesWarning{[1]{%
12704       \PackageWarning{glossaries}{##1}%
12705     }}%
12706     \renewcommand*\GlossariesWarningNoLine{[1]{%
12707       \PackageWarningNoLine{glossaries}{##1}%
12708     }}%
12709     \let\@glsshowtarget\@gobble
12710     \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12711   \or
12712     % debug=false
12713     \@gls@debugfalse
12714     \let\@glsshowtarget\@gobble
12715     \let\@glsshowaccsupp\@gobblethree
12716     \PackageInfo{glossaries}{debug mode OFF}%
12717   \or
12718     % debug=showtargets
12719     \@gls@debugtrue
12720     \renewcommand*\GlossariesWarning{[1]{%
12721       \PackageWarning{glossaries}{##1}%
12722     }}%
12723     \renewcommand*\GlossariesWarningNoLine{[1]{%
12724       \PackageWarningNoLine{glossaries}{##1}%
12725     }}%
12726     \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12727     \renewcommand{\@glsshowtarget}{\@glsshowtarget}%
12728   \or
12729     % debug=showaccsupp
12730     \@gls@debugtrue
12731     \renewcommand*\GlossariesWarning{[1]{%
12732       \PackageWarning{glossaries}{##1}%
12733     }}%
12734     \renewcommand*\GlossariesWarningNoLine{[1]{%
12735       \PackageWarningNoLine{glossaries}{##1}%
12736     }}%
12737     \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
12738     \renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
12739   \fi
12740 }

```

```

12741 \newcommand*{\glsshowtarget}[1]{%
12742   \ifmode
12743     \nfss@text{\glsshowtargetfont [#1]}%
12744   \else
12745     \ifinner
12746       {\glsshowtargetfont [#1]}%
12747     \else
12748       \glsshowtargetouter{#1}%
12749     \fi
12750   \fi
12751 }
12752 \newcommand*{\glsshowtargetouter}[1]{%
12753   \glsshowtargetsymbols\marginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
12754 \newcommand*{\glsshowtargetsymbols}{\tiny$\triangleright$}
12755 \newcommand*{\glsshowtargetfont}{\ttfamily\footnotesize}
12756 \newcommand*{\@glsshowtarget}[1]{%
12757   \newrobustcmd*{\@glsshowtarget}[1]{%
12758     \begingroup
12759     \protected@edef\@gls@tmp{#1}%
12760     \@onelevel@sanitize\@gls@tmp
12761     \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
12762   \endgroup
12763 }
12764 \newcommand*{\@glsshowaccsupp}[3]{%
12765   \newrobustcmd*{\glsshowaccsupp}[3]{%
12766     \ifstrepty{#1}%
12767       {\@glsshowtarget{/#2 (#3)}}%
12768       {\@glsshowtarget{/#2 (#3) [#1]}}%
12769 }
12770 \newcommand*{\@gls@see@noindex}{%
12771   \PackageError{glossaries}%
12772   {'\gls@xr@key' key may only be used after \string\makeglossaries\space
12773   or \string\makenoidxglossaries\space (or move
12774   \string\newglossaryentry\space
12775   definitions into the preamble))%
12776   {You must use \string\makeglossaries\space
12777   or \string\makenoidxglossaries\space before defining
12778   any entries that have a '\gls@xr@key' key. It may
12779   be that the 'see' key has been written to the .glsdefs
12780   file from the previous run, in which case you need to
12781   move your definitions
12782   to the preamble if you don't want to use
12783   \string\makeglossaries\space
12784   or \string\makenoidxglossaries}%
12785 }
12786 \define@choicekey{glossaries.sty}{seenindex}%
12787   [\gls@seenindex@val\gls@seenindex@nr]{error,warn,ignore}{%
12788   \ifcase\gls@seenindex@nr
12789     \renewcommand*{\@gls@see@noindex}{%

```

```

12790 \PackageError{glossaries}%
12791 {'\gls@xr@key' key may only be used after \string\makeglossaries\space
12792 or \string\makenoidxglossaries}%
12793 {You must use \string\makeglossaries\space
12794 or \string\makenoidxglossaries\space before defining
12795 any entries that have a '\gls@xr@key' key}%
12796 }%
12797 \or
12798 \renewcommand*\@gls@see@noindex{%
12799 \GlossariesWarning{'\gls@xr@key' key ignored}%
12800 }%
12801 \or
12802 \renewcommand*\@gls@see@noindex{}%
12803 \fi
12804 }
12805 \define@boolkey{glossaries.sty}[gls]{toc}[true]{}
12806 \define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
12807 \ifcsundef{chapter}%
12808 {\newcommand*\@@glossarysec{section}}%
12809 {\newcommand*\@@glossarysec{chapter}}
12810 \define@choicekey{glossaries.sty}{section}{part,chapter,section,%
12811 subsection,subsubsection,paragraph,subparagraph}[section]{}
12812 \renewcommand*\@@glossarysec{#1}}
12813 \newcommand*\@@glossarysecstar{*}
12814 \newcommand*\@@glossaryseclabel{}
12815 \newcommand*\glsautoprefix{}
12816 \define@choicekey{glossaries.sty}{numberedsection}%
12817 [\gls@numberedsection@val\gls@numberedsection@nr]{}
12818 false,nolabel,autolabel,nameref}[nolabel]{}
12819 \ifcase\gls@numberedsection@nr\relax
12820 \renewcommand*\@@glossarysecstar{*}%
12821 \renewcommand*\@@glossaryseclabel{}%
12822 \or
12823 \renewcommand*\@@glossarysecstar{}%
12824 \renewcommand*\@@glossaryseclabel{}%
12825 \or
12826 \renewcommand*\@@glossarysecstar{}%
12827 \renewcommand*\@@glossaryseclabel{%
12828 \label{\glsautoprefix\@glo@type}}%
12829 \or
12830 \renewcommand*\@@glossarysecstar{*}%
12831 \renewcommand*\@@glossaryseclabel{%
12832 \protected@edef\@currentlabelname{\glossarytoctitle}%
12833 \label{\glsautoprefix\@glo@type}}%
12834 \fi
12835 }
12836 \ifpackageloaded{classicthesis}
12837 {\newcommand*\@glossary@default@style{index}}
12838 {\newcommand*\@glossary@default@style{list}}

```



```

12839 \define@key{glossaries.sty}{style}{%
12840   \def\@glossary@default@style{#1}%
12841 }
12842 \newcommand*\@gls@declareoption}[2]{%
12843   \DeclareOptionX{#1}{#2}%
12844   \DeclareOption{#1}{#2}%
12845 }
12846 \newcommand*\@glossaryentrynumbers}[1]{#1\@gls@save@numberlist{#1}}
12847 \@gls@declareoption{nonumberlist}{%
12848   \renewcommand*\@glossaryentrynumbers}[1]{\@gls@save@numberlist{#1}}%
12849 }
12850 \define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
12851   \glssavenumberlistfalse
12852 \newcommand*\@glo@seeautonumberlist{%
12853   \@gls@declareoption{seeautonumberlist}{%
12854     \renewcommand*\@glo@seeautonumberlist{%
12855       \def\@glo@prefix{\glsnextpages}%
12856     }%
12857 }
12858 \define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
12859   \glsclocationstrue
12860 \newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
12861 \@gls@declareoption{nolong}{\renewcommand*\@gls@loadlong{}}
12862 \IfFileExists{supertabular.sty}{%
12863   \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}%
12864   \newcommand*\@gls@loadsuper{}%
12865 \@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}
12866 \newcommand*\@gls@loadlist{\RequirePackage{glossary-list}[v4.46]}
12867 \@gls@declareoption{nolist}{%
12868   \renewcommand*\@gls@loadlist{%
12869     \ifdefstring{\@glossary@default@style}{list}%
12870     {\let\@glossary@default@style\relax}%
12871   }%
12872 }%
12873 }
12874 \newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}}
12875 \@gls@declareoption{notree}{\renewcommand*\@gls@loadtree{}}
12876 \@gls@declareoption{nostyles}{%
12877   \renewcommand*\@gls@loadlong}{%
12878   \renewcommand*\@gls@loadsuper}{%
12879   \renewcommand*\@gls@loadlist}{%
12880   \renewcommand*\@gls@loadtree}{%
12881   \let\@glossary@default@style\relax
12882 }
12883 \newcommand*\@gls@postdescription{%
12884   \ifglspostdot\else.\spacefactor\sfcode'\. \fi
12885 }
12886 \define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
12887   \glspostdotfalse

```

```

12888 \define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
12889 \glsnogroupskipfalse
12890 \define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
12891 \@ifclassloaded{memoir}
12892 {%
12893   \glsucmarktrue
12894 }%
12895 {%
12896   \glsucmarkfalse
12897 }
12898 \newcommand*{\@gls@define@glossaryentrycounter}{%
12899   \ifglssentrycounter
12900     \ifundef\c@glossaryentry
12901     {%
12902       \ifx\@gls@counterwithin\@empty
12903         \newcounter{glossaryentry}%
12904       \else
12905         \newcounter{glossaryentry}[\@gls@counterwithin]%
12906       \fi
12907       \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
12908     }%
12909   {}%
12910 \fi
12911 }
12912 \define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
12913 \glssentrycounterfalse
12914 \define@key{glossaries.sty}{counterwithin}{%
12915   \renewcommand*{\@gls@counterwithin}{#1}%
12916   \glssentrycountertrue
12917   \@gls@define@glossaryentrycounter
12918 }
12919 \newcommand*{\@gls@counterwithin}{}
12920 \newcommand{\@gls@define@glossarysubentrycounter}{%
12921   \ifundef\c@glossarysubentry
12922   {%
12923     \ifglsssubentrycounter
12924     \ifglssentrycounter
12925       \newcounter{glossarysubentry}[glossaryentry]%
12926     \else
12927       \newcounter{glossarysubentry}%
12928     \fi
12929     \def\theHglossarysubentry{\currentglossary.\currentglsssubentry.\theglossarysubentry}%
12930   \fi
12931   }%
12932   {}%
12933 }
12934 \define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
12935 \glsssubentrycounterfalse
12936 \newcommand*{\@gls@default@sorttype}{standard}

```

```

12937 \define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
12938   \renewcommand*{\@glo@default@sorttype}{#1}%
12939   \csname @gls@setupsort@#1\endcsname
12940 }
12941 \newcommand*{\glsprestandardsort}[3]{%
12942   \glsdosanitizesort
12943 }
12944 \newcommand*{\@glo@check@sortallowed}[1]{%
12945 \newcommand*{\@gls@setupsort@standard}{%
12946   \def\do@glo@storeentry{\@glo@storeentry}%
12947   \def\@gls@defsortcount##1{%
12948     \def\@gls@defsort##1##2{%
12949       \ifx\@glo@sort\@glsdefaultsort
12950         \let\@glo@sort\@glo@name
12951       \fi
12952       \let\glsdosanitizesort\@gls@sanitizesort
12953       \glsprestandardsort{\@glo@sort}{##1}{##2}%
12954       \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
12955     }%
12956     \def\@gls@setsort##1{%
12957       \let\@glo@check@sortallowed\@gobble
12958     }
12959     \@gls@setupsort@standard
12960 \newcommand*{\glssortnumberfmt}[1]{%
12961   \ifnum#1<100000 0\fi
12962   \ifnum#1<10000 0\fi
12963   \ifnum#1<1000 0\fi
12964   \ifnum#1<100 0\fi
12965   \ifnum#1<10 0\fi
12966   \number#1%
12967 }
12968 \newcommand*{\@gls@setupsort@def}{%
12969   \def\do@glo@storeentry{\@glo@storeentry}%
12970   \def\@gls@defsortcount##1{%
12971     \expandafter\global
12972     \expandafter\newcount\csname glossary@##1@sortcount\endcsname
12973   }%
12974   \def\@gls@defsort##1##2{%
12975     \ifcsundef{glossary@##1@sortcount}%
12976     {\@gls@defsortcount{##1}}%
12977     {}%
12978     \expandafter\global\expandafter
12979     \advance\csname glossary@##1@sortcount\endcsname by 1\relax
12980     \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
12981       \expandafter\glssortnumberfmt
12982       {\csname glossary@##1@sortcount\endcsname}}%
12983   }%
12984   \def\@gls@setsort##1{%
12985     \let\@glo@check@sortallowed\@gobble

```

```

12986 }
12987 \newcommand*{\@gls@setupsort@use}{%
12988   \let\do@glo@storeentry\@gobble
12989   \def\@gls@defsortcount##1{%
12990     \expandafter\global
12991     \expandafter\newcount\csname glossary@##1@sortcount\endcsname
12992   }%
12993   \def\@gls@defsort##1##2{%
12994     \expandafter\gdef\csname glo@##2@sort\endcsname{%
12995     }%
12996   \def\@gls@setsort##1{%
12997     \edef\@glo@parent{\csname glo@##1@parent\endcsname}%
12998     \ifx\@glo@parent\@empty
12999     \else
13000       \expandafter\@gls@setsort\expandafter{\@glo@parent}%
13001     \fi
13002     \edef\@glo@type{\csname glo@##1@type\endcsname}%
13003     \edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
13004     \ifx\@gls@tmp\@empty
13005       \expandafter\global\expandafter
13006       \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
13007       \expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
13008         \expandafter\glssortnumberfmt
13009         {\csname glossary@\@glo@type @sortcount\endcsname}}%
13010       \@glo@storeentry{##1}%
13011     \fi
13012   }%
13013   \let\@glo@check@sortallowed\@gobble
13014 }
13015 \newcommand*{\@gls@setupsort@none}{%
13016   \def\do@glo@storeentry##1{}%
13017   \def\@gls@defsortcount##1{}%
13018   \def\@gls@defsort##1##2{%
13019     \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
13020   }%
13021   \def\@gls@setsort##1{}%
13022   \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
13023     {Option sort=none not allowed with \string##1}%
13024     {(Use sort=def instead)}}%
13025 }
13026 \newcommand*{\glsdefmain}{%
13027   \if@gls@docloaded
13028     \newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
13029   \else
13030     \newglossary{main}{gls}{glo}{\glossaryname}%
13031   \fi
13032   \newcommand*{\gls@tr@set@main@toctitle}{%
13033     \translatelet{\glossarytoctitle}{Glossary}%
13034   }%

```

```

13035 }
13036 \newcommand*{\glsdefaulttype}{main}
13037 \newcommand*{\acronymtype}{\glsdefaulttype}
13038 \@gls@declareoption{nomain}{%
13039   \let\glsdefaulttype\relax
13040   \renewcommand*{\glsdefmain}{}}%
13041 }
13042 \define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
13043   \ifglsacronym
13044     \renewcommand*{\@gls@do@acronymsdef}{%
13045       \DeclareAcronymList{acronym}%
13046       \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
13047       \renewcommand*{\acronymtype}{acronym}%
13048       \newcommand*{\gls@tr@set@acronym@toctitle}{%
13049         \translatelet{\glossarytoctitle}{Acronyms}%
13050       }%
13051     }%
13052   \else
13053     \let\@gls@do@acronymsdef\relax
13054   \fi
13055 }
13056 \AtBeginDocument{%
13057   \ifglsacronym
13058     \ifbool{glscompatible-3.07}%
13059     {}%
13060     {%
13061       \providecommand*{\printacronyms}[1][]{%
13062         \printglossary[type=\acronymtype,#1]}%
13063     }%
13064   \fi
13065 }
13066 \newcommand*{\@gls@do@acronymsdef}{}
13067 \@gls@declareoption{acronyms}{%
13068   \glsacronymtrue
13069   \def\@gls@do@acronymsdef{%
13070     \DeclareAcronymList{acronym}%
13071     \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
13072     \renewcommand*{\acronymtype}{acronym}%
13073     \newcommand*{\gls@tr@set@acronym@toctitle}{%
13074       \translatelet{\glossarytoctitle}{Acronyms}%
13075     }%
13076   }%
13077 }
13078 \newcommand*{\@glsacronymlists}{}
13079 \newcommand*{\@addtoacronymlists}[1]{%
13080   \ifx\@glsacronymlists\@empty
13081     \protected@xdef\@glsacronymlists{#1}%
13082   \else
13083     \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%

```

```

13084 \fi
13085 }
13086 \newcommand*{\DeclareAcronymList}[1]{%
13087   \glsIfListOfAcronyms{#1}{}{\@addtoacronymlists{#1}}%
13088 }
13089 \newcommand{\glsIfListOfAcronyms}[1]{%
13090   \edef\@do@gl@s@islistofacronyms{%
13091     \noexpand\@gl@s@islistofacronyms{#1}{\@gl@sacronymlists}}%
13092   \@do@gl@s@islistofacronyms
13093 }
13094 \newcommand{\@gl@s@islistofacronyms}[4]{%
13095   \def\gl@s@islistofacronyms##1,#1,##2\end@gl@s@islistofacronyms{%
13096     \def\@gl@s@before{##1}\def\@gl@s@after{##2}}%
13097   \gl@s@islistofacronyms,#2,#1,\@nil\end@gl@s@islistofacronyms
13098   \ifx\@gl@s@after\@nnil
13099     #4%
13100   \else
13101     #3%
13102   \fi
13103 }
13104 \newif\if@gl@sisacronymlist
13105 \newcommand*{\gl@s@checkisacronymlist}[1]{%
13106   \glsIfListOfAcronyms{#1}%
13107   {\@gl@sisacronymlisttrue}{\@gl@sisacronymlistfalse}%
13108 }
13109 \newcommand*{\SetAcronymLists}[1]{%
13110   \renewcommand*{\@gl@sacronymlists}{#1}%
13111 }
13112 \define@key{glossaries.sty}{acronymlists}{%
13113   \DeclareAcronymList{#1}%
13114 }
13115 \newcommand{\glscounter}{page}
13116 \define@key{glossaries.sty}{counter}{%
13117   \renewcommand*{\glscounter}{#1}%
13118 }
13119 \newcommand*{\@gl@s@nohyperlist}{}
13120 \newcommand*{\GlsDeclareNoHyperList}[1]{%
13121   \ifdefempty\@gl@s@nohyperlist
13122     {%
13123       \renewcommand*{\@gl@s@nohyperlist}{#1}%
13124     }%
13125     {%
13126       \appto\@gl@s@nohyperlist{,#1}%
13127     }%
13128 }
13129 \define@key{glossaries.sty}{nohypertypes}{%
13130   \GlsDeclareNoHyperList{#1}%
13131 }
13132 \newcommand*{\GlossariesWarning}[1]{%

```

```

13133 \PackageWarning{glossaries}{#1}%
13134 }
13135 \newcommand*\GlossariesWarningNoLine}[1]{%
13136 \PackageWarningNoLine{glossaries}{#1}%
13137 }
13138 \newcommand{\glosortentrieswarning}{%
13139 \typeout{Using TeX to sort glossary entries---this may
13140 take a while}%
13141 }
13142 \@gls@declareoption{nowarn}{%
13143 \if@gls@debug
13144 \GlossariesWarning{Warnings can't be suppressed in debug mode}%
13145 \else
13146 \renewcommand*\GlossariesWarning}[1]{}%
13147 \renewcommand*\GlossariesWarningNoLine}[1]{}%
13148 \renewcommand*\glosortentrieswarning}{}%
13149 \renewcommand*\@gls@missinglang@warn}[2]{}%
13150 \fi
13151 }
13152 \newcommand*\@gls@missinglang@warn}[2]{%
13153 \PackageWarningNoLine{glossaries}%
13154 {No language module detected for '#1'.\MessageBreak
13155 Language modules need to be installed separately.\MessageBreak
13156 Please check on CTAN for a bundle called\MessageBreak
13157 'glossaries-#2' or similar}%
13158 }
13159 \@gls@declareoption{nolangwarn}{%
13160 \renewcommand*\@gls@missinglang@warn}[2]{}%
13161 }
13162 \newcommand*\@gls@warnonglossdefined}{%
13163 \GlossariesWarning{Overriding \string\printglossary}%
13164 }
13165 \newcommand*\@gls@warnontheglossdefined}{%
13166 \GlossariesWarning{Overriding 'theglossary' environment}%
13167 }
13168 \@gls@declareoption{noredefwarn}{%
13169 \renewcommand*\@gls@warnonglossdefined}{}%
13170 \renewcommand*\@gls@warnontheglossdefined}{}%
13171 }
13172 \newcommand*\@gls@sanitizedesc}{%
13173 }
13174 \newcommand*\glssetexpandfield}[1]{%
13175 \csdef{gls@assign@#1@field}##1##2{%
13176 \@@gls@expand@field{##1}{#1}{##2}%
13177 }%
13178 }
13179 \newcommand*\glssetnoexpandfield}[1]{%
13180 \csdef{gls@assign@#1@field}##1##2{%
13181 \@@gls@noexpand@field{##1}{#1}{##2}%

```

```

13182 }%
13183 }
13184 \glssetexpandfield{type}
13185 \glssetnoexpandfield{desc}
13186 \glssetnoexpandfield{descplural}
13187 \newcommand*{\@gls@sanitizename}{%
13188 \glssetnoexpandfield{name}
13189 \newcommand*{\@gls@sanitizesymbol}{%
13190 \glssetnoexpandfield{symbol}
13191 \glssetnoexpandfield{symbolplural}
13192 \newcommand*{\@gls@sanitizesort}{%
13193 \ifglssanitizesort
13194 \@@gls@sanitizesort
13195 \else
13196 \@@gls@nosanitizesort
13197 \fi
13198 }
13199 \newcommand*{\@@gls@sanitizesort{%
13200 \@onelevel@sanitize\@glo@sort
13201 }
13202 \newcommand*{\@@gls@nosanitizesort}{%
13203 \newcommand*{\@gls@noidx@sanitizesort{%
13204 \ifdefvoid\@glo@sort
13205 }%
13206 {%
13207 \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
13208 }%
13209 }
13210 \def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
13211 \def\@glo@sort{#1#2}%
13212 \@onelevel@sanitize\@glo@sort
13213 }
13214 \newcommand*{\@@gls@noidx@nosanitizesort}{%
13215 \ifdefvoid\@glo@sort
13216 }%
13217 {%
13218 \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
13219 }%
13220 }
13221 \def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
13222 \bgroup
13223 \glsnoidxstripaccents
13224 \protected@xdef\@glo@sort{#1#2}%
13225 \egroup
13226 \let\@glo@sort\@glo@sort
13227 }
13228 \newcommand*\glsnoidxstripaccents{%
13229 \let\IeC\@firstofone
13230 \let\add@accent@\@secondoftwo

```



```

13231 \let\@text@composite@x\@secondoftwo
13232 \let\@tabacckludge\@secondoftwo
13233 \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
13234 \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
13235 \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
13236 \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
13237 \let\'\'@firstofone
13238 \let\'\'@firstofone
13239 \let\~\@firstofone
13240 \let\"\"@firstofone
13241 \let\u\"@firstofone
13242 \let\t\"@firstofone
13243 \let\d\"@firstofone
13244 \let\r\"@firstofone
13245 \let\=\"@firstofone
13246 \let\.\"@firstofone
13247 \let\~\"@firstofone
13248 \let\v\"@firstofone
13249 \let\H\"@firstofone
13250 \let\c\"@firstofone
13251 \let\b\"@firstofone
13252 \let\a\"@secondoftwo
13253 \def\AE{AE}%
13254 \def\ae{ae}%
13255 \def\OE{OE}%
13256 \def\oe{oe}%
13257 \def\AA{AA}%
13258 \def\aa{aa}%
13259 \def\L{L}%
13260 \def\l{l}%
13261 \def\O{O}%
13262 \def\o{o}%
13263 \def\SS{SS}%
13264 \def\ss{ss}%
13265 \def\th{th}%
13266 \def\TH{TH}%
13267 \def\dh{dh}%
13268 \def\DH{DH}%
13269 }
13270 \@ifl@t@r\fmtversion{2019/10/01}
13271 {%
13272 \appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
13273 }
13274 {}
13275 \define@boolkey[gl]{sanitize}{description}[true]{%
13276 \GlossariesWarning{sanitize={description} package option deprecated}%
13277 \ifgl@sanitize@description
13278 \glssetnoexpandfield{desc}%
13279 \glssetnoexpandfield{descplural}%

```

```

13280 \else
13281   \glsssetexpandfield{desc}%
13282   \glsssetexpandfield{descplural}%
13283 \fi
13284 }
13285 \define@boolkey[glS]{sanitize}{name}[true]{%
13286   \GlossariesWarning{sanitize={name} package option deprecated}%
13287   \ifglS@sanitize@name
13288     \glsssetnoexpandfield{name}%
13289   \else
13290     \glsssetexpandfield{name}%
13291   \fi
13292 }
13293 \define@boolkey[glS]{sanitize}{symbol}[true]{%
13294   \GlossariesWarning{sanitize={symbol} package option deprecated}%
13295   \ifglS@sanitize@symbol
13296     \glsssetnoexpandfield{symbol}%
13297     \glsssetnoexpandfield{symbolplural}%
13298   \else
13299     \glsssetexpandfield{symbol}%
13300     \glsssetexpandfield{symbolplural}%
13301   \fi
13302 }
13303 \define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
13304   \ifglsssanitizesort
13305     \glsssetnoexpandfield{sortvalue}%
13306     \renewcommand*{\@glS@noidx@setsanitizesort}{%
13307       \glsssanitizesorttrue
13308       \glsssetnoexpandfield{sortvalue}%
13309     }%
13310   \else
13311     \glsssetexpandfield{sortvalue}%
13312     \renewcommand*{\@glS@noidx@setsanitizesort}{%
13313       \glsssanitizesortfalse
13314       \glsssetexpandfield{sortvalue}%
13315     }%
13316   \fi
13317 }
13318 \glsssanitizesorttrue
13319 \glsssetnoexpandfield{sortvalue}%
13320 \newcommand*{\@glS@noidx@setsanitizesort}{%
13321   \glsssanitizesortfalse
13322   \glsssetexpandfield{sortvalue}%
13323 }
13324 \define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
13325   \setbool{glsssanitizesort}{#1}%
13326   \ifglsssanitizesort
13327     \glsssetnoexpandfield{sortvalue}%
13328   \else

```

```

13329 \glssetexpandfield{sortvalue}%
13330 \fi
13331 \GlossariesWarning{sanitize={sort} package option
13332 deprecated. Use sanitizesort instead}%
13333 }
13334 \define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
13335 \ifthenelse{\equal{#1}{none}}{%
13336 {%
13337 \GlossariesWarning{sanitize package option deprecated}%
13338 \glssetexpandfield{name}%
13339 \glssetexpandfield{symbol}%
13340 \glssetexpandfield{symbolplural}%
13341 \glssetexpandfield{desc}%
13342 \glssetexpandfield{descplural}%
13343 }%
13344 {%
13345 \setkeys{gls}{sanitize}{#1}%
13346 }%
13347 }
13348 \newif\ifglstranslate
13349 \newcommand*\@gls@usetranslator{%
13350 \@ifpackageloaded{polyglossia}%
13351 {%
13352 \let\glsifusetranslator\@secondoftwo
13353 }%
13354 {%
13355 \@ifpackageloaded{babel}%
13356 {%
13357 \IfFileExists{translator.sty}%
13358 {%
13359 \RequirePackage{translator}%
13360 \let\glsifusetranslator\@firstoftwo
13361 }%
13362 {}%
13363 }%
13364 {}%
13365 }%
13366 }
13367 \newcommand{\glsifusedtranslatordict}[3]{%
13368 \glsifusetranslator
13369 {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
13370 {#3}%
13371 }
13372 \@gls@declareoption{nottranslate}{%
13373 \glstranslatefalse
13374 \let\@gls@usetranslator\relax
13375 \let\glsifusetranslator\@secondoftwo
13376 }
13377 \define@choicekey{glossaries.sty}{translate}%

```

```

13378 [\gls@translate@val\gls@translate@nr]%
13379 {true,false,babel}[true]%
13380 {%
13381   \ifcase\gls@translate@nr\relax
13382     \glstranslatetrue
13383     \renewcommand*\@gls@usetranslator{%
13384       \@ifpackageloaded{polyglossia}%
13385       {%
13386         \let\glsifusetranslator\@secondoftwo
13387       }%
13388       {%
13389         \@ifpackageloaded{babel}%
13390         {%
13391           \IfFileExists{translator.sty}%
13392           {%
13393             \RequirePackage{translator}%
13394             \let\glsifusetranslator\@firstoftwo
13395           }%
13396           {}%
13397         }%
13398         {}%
13399       }%
13400     }%
13401   \or
13402     \glstranslatefalse
13403     \let\@gls@usetranslator\relax
13404     \let\glsifusetranslator\@secondoftwo
13405   \or
13406     \glstranslatetrue
13407     \let\@gls@usetranslator\relax
13408     \let\glsifusetranslator\@secondoftwo
13409   \fi
13410 }
13411 \glstranslatefalse
13412 \let\glsifusetranslator\@secondoftwo
13413 \@ifpackageloaded{translator}%
13414 {%
13415   \glstranslatetrue
13416   \let\glsifusetranslator\@firstoftwo
13417 }%
13418 {%
13419   \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
13420   {
13421     \@ifpackageloaded{\gls@thissty}%
13422     {%
13423       \glstranslatetrue
13424       \@endfortrue
13425     }%
13426     {}%

```

```

13427 }
13428 }
13429 \define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
13430 \glsindexonlyfirstfalse
13431 \define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
13432 \glshyperfirsttrue
13433 \newcommand*{\@gls@setacrstyle}{}
13434 \define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
13435 \ifbool{glsacrdescription}%
13436 {}%
13437 {%
13438 \renewcommand*{\@gls@sanitizedesc}{}%
13439 }%
13440 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13441 }
13442 \define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
13443 \renewcommand*{\@gls@sanitizesymbol}{}%
13444 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13445 }
13446 \define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
13447 \renewcommand*{\@gls@sanitizesymbol}{}%
13448 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13449 }
13450 \define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
13451 \renewcommand*{\@gls@sanitizesymbol}{}%
13452 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13453 }
13454 \define@boolkey{glossaries.sty}[glsacr]{dua}[true]{}
13455 \renewcommand*{\@gls@sanitizesymbol}{}%
13456 \renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
13457 }
13458 \define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
13459 \newcommand*{\glsorder}{word}
13460 \newcommand*{\@glsorder}[1]{}
13461 \define@choicekey{glossaries.sty}{order}{word,letter}{}
13462 \def\glsorder{#1}
13463 \newif\ifglxindy
13464 \glxindyfalse
13465 \@gls@declareoption{makeindex}{\glxindyfalse}
13466 \define@boolkey[gls]{xindy}{glsnumbers}[true]{}
13467 \gls@xindy@glsnumberstrue
13468 \def\@xdy@main@language{\language}%
13469 \define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
13470 \ifcsundef{inputencodingname}{}%
13471 \def\gls@codepage{}{}%
13472 \def\gls@codepage{\inputencodingname}
13473 }
13474 \define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
13475 \define@key{glossaries.sty}{xindy}[]{}

```

```

13476 \glxindytrue
13477 \setkeys[glx]{indy}{#1}%
13478 }
13479 \@glx@declareoption{xindygloss}{%
13480 \glxindytrue
13481 }
13482 \@glx@declareoption{xindynoglsnumbers}{%
13483 \glxindytrue
13484 \glx@xindy@glxnumbersfalse
13485 }
13486 \providecommand{\@domakeglossaries}[1]{#1}
13487 \@glx@declareoption{disablemakegloss}{%
13488 \ifdefequal\makeglossaries\@no@makeglossaries
13489 {%
13490 \GlossariesWarning{Option ‘disablemakegloss’ has no effect
13491 (\string\makenoidxglossaries\space has already been used)}%
13492 }%
13493 {%
13494 \ifdefequal\@makeglossary\@gobble
13495 {%
13496 \GlossariesWarning{Option ‘disablemakegloss’ has no effect
13497 (\string\makeglossaries\space has already been used)}%
13498 }%
13499 {%
13500 \renewcommand{\@domakeglossaries}[1]{%
13501 \PackageInfo{glossaries}{\string\makeglossaries\space and
13502 \string\makenoidxglossaries\space have been disabled}%
13503 }%
13504 }%
13505 }%
13506 }
13507 \@glx@declareoption{restoremakegloss}{%
13508 \ifdefequal\makeglossaries\@no@makeglossaries
13509 {%
13510 \GlossariesWarning{Option ‘restoremakegloss’ has no effect
13511 (\string\makenoidxglossaries\space has already been used)}%
13512 }%
13513 {%
13514 \ifdefequal\@makeglossary\@gobble
13515 {%
13516 \GlossariesWarning{Option ‘restoremakegloss’ has no effect
13517 (\string\makeglossaries\space has already been used)}%
13518 }%
13519 {%
13520 \PackageInfo{glossaries}{\string\makeglossaries\space and
13521 \string\makenoidxglossaries\space have been restored}%
13522 \let\@domakeglossaries\@firstofone
13523 }%
13524 }%

```

```

13525 }
13526 \newcommand*{\@do@write@glslabels}{%
13527   \AtEndDocument{\@do@write@glslabels}%
13528   \let\@do@write@glslabels\relax
13529 }
13530 \newcommand*{\@do@do@write@glslabels}{%
13531   \newwrite\@glsl@labelsfile
13532   \immediate\openout\@glsl@labelsfile=\jobname.glslabels
13533   \forallglstentries[\@glo@types,\@ignored@glossaries]{\@glstentry}%
13534   {\ifdefempty{\@glstentry}{\immediate\write\@glsl@labelsfile{\@glstentry}}}%
13535   \immediate\closeout\@glsl@labelsfile
13536 }
13537 \@glsl@declareoption{writeglslabels}{\@do@write@glslabels}
13538 \newif\ifglslautomake
13539 \newcommand{\glsl@automake@nr}{1}
13540 \define@choicekey{glossaries.sty}{automake}%
13541   [\glsl@automake@val\glsl@automake@nr]{true,false,immediate}[true]{%
13542   \ifnum\glsl@automake@nr=1\relax
13543     \glslautomakefalse
13544   \else
13545     \glslautomaketrue
13546   \fi
13547   \ifglslautomake
13548     \renewcommand*{\@glsl@doautomake}{%
13549       \PackageError{glossaries}{You must use
13550       \string\makeglossaries\space with automake=true}
13551       {%
13552         Either remove the automake=true setting or
13553         add \string\makeglossaries\space to your document preamble.%
13554       }%
13555     }%
13556   \else
13557     \renewcommand*{\@glsl@doautomake}{}%
13558   \fi
13559 }
13560 \glslautomakefalse
13561 \newcommand*{\@glsl@doautomake}{}
13562 \AtEndDocument{\@glsl@doautomake}
13563 \define@boolkey{glossaries.sty}[glsl]{savewrites}[true]{%
13564   \ifglssavewrites
13565     \renewcommand*{\glslwritefiles}{\@glslwritefiles}%
13566   \else
13567     \let\glslwritefiles\@empty
13568   \fi
13569 }
13570 \glssavewritesfalse
13571 \let\glslwritefiles\@empty
13572 \define@boolkey{glossaries.sty}[glsl]{compatible-3.07}[true]{}
13573 \boolfalse{glslcompatible-3.07}

```

```

13574 \define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
13575   \ifbool{glscompatible-2.07}%
13576   {%
13577     \booltrue{glscompatible-3.07}%
13578   }%
13579   {}%
13580 }
13581 \boolfalse{glscompatible-2.07}
13582 \let\gls@original@makeglossary\makeglossary
13583 \let\gls@original@glossary\glossary
13584 \def\makeglossary{%
13585   \GlossariesWarning{Use of \string\makeglossary\space with
13586   glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
13587   instead. If you \MessageBreak need the original definition of
13588   \string\makeglossary\space use \MessageBreak the package options
13589   kernelglossredefs=false (to \MessageBreak restore the former definition of
13590   \string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
13591   conflict)}}%
13592   \makeglossaries
13593 }
13594 \newcommand*{\@gls@override@glossary}[1][main]{%
13595   \GlossariesWarning{Use of \string\glossary\space with
13596   glossaries.sty is deprecated. \MessageBreak Indexing should be performed
13597   with the user level \MessageBreak commands, such as \string\gls\space or
13598   \string\glsadd. If you need the \MessageBreak original definition of
13599   \string\glossary\space use the package \MessageBreak options
13600   kernelglossredefs=false (to restore the \MessageBreak former definition of
13601   \string\glossary) and nomain (if the \MessageBreak file extensions cause a
13602   conflict)}}%
13603   \gls@glossary{#1}%
13604 }
13605 \if@gls@docloaded
13606 \else
13607   \def\glossary{\@gls@override@glossary}
13608 \fi
13609 \define@choicekey{glossaries.sty}{kernelglossredefs}%
13610   [\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
13611 {%
13612   \ifcase\gls@debug@nr\relax
13613     \def\glossary{\@gls@override@glossary}%
13614     \def\makeglossary{%
13615       \GlossariesWarning{Use of \string\makeglossary\space with
13616       glossaries.sty is deprecated. Use \string\makeglossaries\space
13617       instead. If you need the original definition of
13618       \string\makeglossary\space use the package options
13619       kernelglossredefs=false (to prevent redefinition of
13620       \string\makeglossary) and nomain (if the file extensions cause a
13621       conflict)}}%
13622       \makeglossaries

```



```

13623 }%
13624 \or
13625 \let\glossary\gls@original@glossary
13626 \let\makeglossary\gls@original@makeglossary
13627 \or
13628 \def\makeglossary{\makeglossaries}%
13629 \renewcommand*{\@gls@override@glossary}[1][main]{%
13630 \gls@glossary{##1}%
13631 }%
13632 \fi
13633 }
13634 \@gls@declareoption{symbols}{%
13635 \let\@gls@do@symbolsdef\@gls@symbolsdef
13636 }
13637 \newcommand*{\@gls@do@symbolsdef}{%
13638 \newcommand*{\@gls@symbolsdef}{%
13639 \newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
13640 \newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]]{%
13641 \newcommand*{\gls@tr@set@symbols@toctitle}{%
13642 \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
13643 }%
13644 }%
13645 \@gls@declareoption{numbers}{%
13646 \let\@gls@do@numbersdef\@gls@numbersdef
13647 }
13648 \newcommand*{\@gls@do@numbersdef}{%
13649 \newcommand*{\@gls@numbersdef}{%
13650 \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
13651 \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
13652 \newcommand*{\gls@tr@set@numbers@toctitle}{%
13653 \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
13654 }%
13655 }%
13656 \@gls@declareoption{index}{%
13657 \ifx\@gls@do@indexdef\@empty
13658 \let\@gls@do@indexdef\@gls@indexdef
13659 \fi
13660 }
13661 \@gls@declareoption{noglossaryindex}{%
13662 \let\@gls@do@indexdef\relax
13663 }
13664 \newcommand*{\@gls@do@indexdef}{%
13665 \newcommand*{\@gls@indexdef}{%
13666 \newglossary[ilg]{index}{ind}{idx}{\indexname}%
13667 \newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
13668 \newcommand*{\newterm}[2][\%
13669 \newglossaryentry{##2}%
13670 {type={index},name={##2},description={\nopostdesc},##1}}
13671 \let\@gls@do@indexdef\relax

```

```

13672 }%
13673 \@for\CurrentOption := \@declaredoptions\do{%
13674   \ifx\CurrentOption\@empty
13675   \else
13676     \@expandtwoargs
13677     \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
13678     \ifin@
13679     \@use@ption
13680     \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
13681   \fi
13682 \fi
13683 }
13684 \ProcessOptionsX
13685 \RequirePackage{glossaries-compatible-307}
13686 \disable@keys{glossaries.sty}{compatible-2.07,%
13687 xindy,xindygloss,xindynoglsnumbers,makeindex,%
13688 acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
13689 nomain,noglossaryindex}
13690 \newcommand*{\setupglossaries}[1]{%
13691   \renewcommand*{@gls@setacrstyle}{}%
13692   \ifglsacrshortcuts
13693     \def@gls@setupshortcuts{glsacrshortcutstrue}%
13694   \else
13695     \def@gls@setupshortcuts{%
13696       \ifglsacrshortcuts
13697         \DefineAcronymSynonyms
13698       \fi
13699     }%
13700   \fi
13701   \glsacrshortcutsfalse
13702   \let@gls@do@numbersdef\relax
13703   \let@gls@do@symbolssdef\relax
13704   \let@gls@do@indexdef\relax
13705   \let@gls@do@acronymsdef\relax
13706   \ifgl Sentrycounter
13707     \let@gls@doentrycounterdef\relax
13708   \else
13709     \let@gls@doentrycounterdef@gls@define@glossaryentrycounter
13710   \fi
13711   \ifglssubentrycounter
13712     \let@gls@dosubentrycounterdef\relax
13713   \else
13714     \let@gls@dosubentrycounterdef@gls@define@glossarysubentrycounter
13715   \fi
13716   \setkeys{glossaries.sty}{#1}%
13717   \@gls@setacrstyle
13718   \@gls@setupshortcuts
13719   \@gls@do@acronymsdef
13720   \@gls@do@numbersdef

```

```

13721 \@gls@do@symbolssdef
13722 \@gls@do@indexdef
13723 \@gls@doentrycounterdef
13724 \@gls@dosubentrycounterdef
13725 }
13726 \ifthenelse{\equal{\glscounter}{section}}%
13727 {%
13728   \ifcsundef{chapter}{}%
13729   {%
13730     \let\@gls@old@chapter\@chapter
13731     \def\@chapter[#1]#2{\@gls@old@chapter[#1]{#2}%
13732       \ifcsundef{hyperdef}{}{\hyperdef{section}{\thesection}{}}}%
13733   }%
13734 }%
13735 {}
13736 \newcommand*{\@gls@onlypremakeg}{}
13737 \newcommand*{\@onlypremakeg}[1]{%
13738   \ifx\@gls@onlypremakeg\@empty
13739     \def\@gls@onlypremakeg{#1}%
13740   \else
13741     \expandafter\toks@{\expandafter{\@gls@onlypremakeg}%
13742       \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
13743   \fi
13744 }
13745 \newcommand*{\@disable@onlypremakeg}{%
13746 \@for\@thiscs:=\@gls@onlypremakeg\do{%
13747   \expandafter\@disable@premakecs\@thiscs%
13748 }}
13749 \newcommand*{\@disable@premakecs}[1]{%
13750   \def#1{\PackageError{glossaries}{\string#1\space may only be
13751     used before \string\makeglossaries}{You can't use
13752     \string#1\space after \string\makeglossaries}}%
13753 }
13754 \providecommand*{\glossaryname}{Glossary}
13755 \providecommand*{\acronymname}{Acronyms}
13756 \newcommand*{\glssettocitle}[1]{%
13757   \def\glossarytocitle{\csname @glotype@#1@title\endcsname}}
13758 \providecommand*{\entryname}{Notation}
13759 \providecommand*{\descriptionname}{Description}
13760 \providecommand*{\symbolname}{Symbol}
13761 \providecommand*{\pagelistname}{Page List}
13762 \providecommand*{\glssymbolsgroupname}{Symbols}
13763 \providecommand*{\glsnumbersgroupname}{Numbers}
13764 \newcommand*{\glspluralsuffix}{s}
13765 \newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
13766 \newcommand*{\glsupacrpluralsuffix}{\glsacrpluralsuffix}
13767 \providecommand*{\seename}{see}
13768 \providecommand*{\andname}{\&}
13769 \newcommand*{\RequireGlossariesLang}[1]{%

```

```

13770 \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
13771 }
13772 \newcommand*{\ProvidesGlossariesLang}[1]{%
13773   \ProvidesFile{glossaries-#1.ldf}%
13774 }
13775 \newcommand*{\addglossarytocaptions}[1]{%
13776   \ifglstranslate
13777     \RequirePackage{tracklang}
13778     \@gls@usetranslator
13779     \ifpackageloaded{translator}
13780     {%
13781       \ifboolexpr
13782       {
13783         test {\ifdefstring{\trans@languages}{English}}
13784         and not
13785         test {\ifdefstring{bbl@loaded}{english}}
13786       }
13787       {%
13788         \let\glsifusetranslator\@secondoftwo
13789       }%
13790       {%
13791         \usedictionary{glossaries-dictionary}%
13792         \renewcommand*{\addglossarytocaptions}[1]{%
13793           \ifcsundef{captions#1}{}%
13794           {%
13795             \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
13796             \expandafter\toks@\expandafter{\@gls@tmp
13797               \renewcommand*{\glossaryname}{\translate{Glossary}}}%
13798             }%
13799             \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
13800           }%
13801         }%
13802       }%
13803     }%
13804   }{}%
13805   \AnyTrackedLanguages
13806   {%
13807     \ForEachTrackedDialect{\this@dialect}{%
13808       \IfTrackedLanguageFileExists{\this@dialect}%
13809       {glossaries-}% prefix
13810       {.ldf}%
13811       {%
13812         \RequireGlossariesLang{\CurrentTrackedTag}%
13813       }%
13814       {%
13815         \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
13816       }%
13817     }%
13818   }%

```

```

13819 {}%
13820 \glsifusetranslator
13821 {%
13822   \renewcommand*{\glssettoctitle}[1]{%
13823     \ifcsdef{gls@tr@set@#1@toctitle}%
13824     {%
13825       \csuse{gls@tr@set@#1@toctitle}%
13826     }%
13827     {%
13828       \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
13829     }%
13830   }%
13831   \renewcommand*{\glossaryname}{\translate{Glossary}}%
13832   \renewcommand*{\acronymname}{\translate{Acronyms}}%
13833   \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
13834   \renewcommand*{\descriptionname}{%
13835     \translate{Description (glossaries)}}%
13836   \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
13837   \renewcommand*{\pagelistname}{%
13838     \translate{Page List (glossaries)}}%
13839   \renewcommand*{\glssymbolsgroupname}{%
13840     \translate{Symbols (glossaries)}}%
13841   \renewcommand*{\glsnumbersgroupname}{%
13842     \translate{Numbers (glossaries)}}%
13843   }{}%
13844 \fi
13845 \DeclareRobustCommand*{\nopostdesc}{%
13846 \newcommand*{\@nopostdesc}{%
13847   \let\org@glspostdescription\glspostdescription
13848   \def\glspostdescription{%
13849     \let\glspostdescription\org@glspostdescription}%
13850 }
13851 \newcommand*{\@no@post@desc}{\nopostdesc}
13852 \newcommand{\glspar}{\par}
13853 \newcommand{\setStyleFile}[1]{%
13854   \renewcommand*{\gls@istfilebase}{#1}%
13855   \ifglsxindy
13856     \def\istfilename{\gls@istfilebase.xdy}
13857   \else
13858     \def\istfilename{\gls@istfilebase.ist}
13859   \fi
13860 }
13861 \@onlypremakeg\setStyleFile
13862 \ifglsxindy
13863   \def\istfilename{\gls@istfilebase.xdy}
13864 \else
13865   \def\istfilename{\gls@istfilebase.ist}
13866 \fi
13867 \newcommand*{\gls@istfilebase}{\jobname}

```

```

13868 \newcommand*{\@istfilename}[1]{}
13869 \newcommand*{\glscompositor}{.}
13870 \newcommand*{\glsSetCompositor}[1]{%
13871   \renewcommand*{\glscompositor}{#1}}
13872 \@onlypremakeg\glsSetCompositor
13873 \newcommand*{\@glsAlphacompositor}{\glscompositor}
13874 \ifglsexindy
13875   \newcommand*{\glsSetAlphaCompositor}[1]{%
13876     \renewcommand*{\@glsAlphacompositor}{#1}}
13877 \else
13878   \newcommand*{\glsSetAlphaCompositor}[1]{%
13879     \glsnoxywarning\glsSetAlphaCompositor}
13880 \fi
13881 \@onlypremakeg\glsSetAlphaCompositor
13882 \newcommand*{\gls@suffixF}{}
13883 \newcommand*{\glsSetSuffixF}[1]{%
13884   \renewcommand*{\gls@suffixF}{#1}}
13885 \@onlypremakeg\glsSetSuffixF
13886 \newcommand*{\gls@suffixFF}{}
13887 \newcommand*{\glsSetSuffixFF}[1]{%
13888   \renewcommand*{\gls@suffixFF}{#1}}
13889 }
13890 \ifcsundef{hyperlink}%
13891 {%
13892   \newcommand*{\glsnumberformat}[1]{#1}%
13893 }%
13894 {%
13895   \newcommand*{\glsnumberformat}[1]{\glshypernumber{#1}}%
13896 }
13897 \newcommand{\delimN}{, }
13898 \newcommand{\delimR}{--}
13899 \newcommand*{\glossarypreamble}{%
13900   \csuse{@glossarypreamble@\currentglossary}%
13901 }
13902 \newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
13903   \ifglossaryexists*{#1}%
13904     {\csgdef{@glossarypreamble@#1}{#2}}%
13905     {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
13906 }
13907 \newcommand*{\glossarypostamble}{}
13908 \newcommand*{\glossarysection}[2][\@gls@title]{%
13909   \def\@gls@title{#2}%
13910   \ifcsundef{phantomsection}%
13911     {%
13912       \@glossarysection{#1}{#2}%
13913     }%
13914     {%
13915       \@pglossarysection{#1}{#2}%
13916     }%

```

```

13917 \glsglossarymark{\glossarytoctitle}%
13918 }
13919 \ifcsundef{glossarymark}%
13920 {%
13921 \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
13922 }%
13923 {%
13924 \@ifclassloaded{memoir}
13925 {%
13926 \newcommand{\glsglossarymark}[1]{%
13927 \ifglsucmark
13928 \markboth{\memUHead{#1}}{\memUHead{#1}}%
13929 \else
13930 \markboth{#1}{#1}%
13931 \fi
13932 }
13933 }%
13934 {%
13935 \newcommand{\glsglossarymark}[1]{%
13936 \ifglsucmark
13937 \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
13938 \else
13939 \@mkboth{#1}{#1}%
13940 \fi
13941 }
13942 }
13943 }
13944 \providecommand{\glossarymark}[1]{%
13945 \ifglsucmark
13946 \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
13947 \else
13948 \@mkboth{#1}{#1}%
13949 \fi
13950 }
13951 \newcommand*{\setglossarysection}[1]{%
13952 \setkeys{glossaries.sty}{section=#1}}
13953 \newcommand*{\@glossarysection}[2]{%
13954 \ifdefempty\@glossarysecstar
13955 {%
13956 \csname\@glossarysec\endcsname[#1]{#2}%
13957 }%
13958 {%
13959 \csname\@glossarysec\endcsname*{#2}%
13960 \@gls@toc{#1}{\@glossarysec}%
13961 }%
13962 \@glossaryseclabel
13963 }
13964 \newcommand*{\@p@glossarysection}[2]{%
13965 \glsclearpage

```

```

13966 \phantomsection
13967 \ifdefempty\@@glossarysecstar
13968 {%
13969   \csname\@@glossarysec\endcsname{#2}%
13970 }%
13971 {%
13972   \@gls@toc{#1}{\@@glossarysec}%
13973   \csname\@@glossarysec\endcsname*{#2}%
13974 }%
13975 \@@glossaryseclabel
13976 }
13977 \newcommand*\@gls@docclearpage{%
13978   \ifthenelse{\equal{\@@glossarysec}{chapter}}%
13979   {%
13980     \ifcsundef{cleardoublepage}%
13981     {%
13982       \clearpage
13983     }%
13984     {%
13985       \ifcsdef{if@openright}%
13986       {%
13987         \if@openright
13988           \cleardoublepage
13989         \else
13990           \clearpage
13991         \fi
13992       }%
13993       {%
13994         \cleardoublepage
13995       }%
13996     }%
13997   }%
13998   {}%
13999 }
14000 \newcommand*\@gls@clearpage{\@gls@docclearpage}
14001 \newcommand*\@gls@toc[2]{%
14002   \ifglstoc
14003     \ifglslnumberline
14004       \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
14005     \else
14006       \addcontentsline{toc}{#2}{#1}%
14007     \fi
14008   \fi
14009 }
14010 \newcommand*\@gls@noxindywarning[1]{%
14011   \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
14012 }
14013 \newcommand*\@gls@nomakeindexwarning[1]{%
14014   \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%

```



```

14015 }
14016 \ifglxsindy
14017   \edef\xdyattributes{\string"default\string"}%
14018 \fi
14019 \ifglxsindy
14020   \edef\xdyattributelist{}%
14021 \fi
14022 \ifglxsindy
14023   \def\xdylocref{}
14024 \fi
14025 \newcommand*{\@gls@ifinlist}[4]{%
14026   \def\@do@ifinlist##1,##2\end@ifinlist{%
14027     \def\@gls@listsuffix{##2}%
14028     \ifx\@gls@listsuffix\@empty
14029       #4%
14030     \else
14031       #3%
14032     \fi
14033   }%
14034   \@do@ifinlist,##1,##2,\end@ifinlist
14035 }
14036 \ifglxsindy
14037   \newcommand*{\@xdycounters}{\glscounter}
14038   \newcommand*\GlsAddXdyCounters[1]{%
14039     \@for\@gls@ctr:=#1\do{%
14040       \edef\@do@addcounter{%
14041         \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
14042         {%
14043           \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
14044             \noexpand\@gls@ctr}%
14045         }%
14046       }%
14047       \@do@addcounter
14048     }
14049   }
14050   \@onlypremakeg\GlsAddXdyCounters
14051 \else
14052   \newcommand*\GlsAddXdyCounters[1]{%
14053     \glsnoxindywarning\GlsAddXdyAttribute
14054   }
14055 \fi
14056 \newcommand*\@disabled@glsaddxdycounters{%
14057   \PackageError{glossaries}{\string\GlsAddXdyCounters\space
14058     can't be used after \string\GlsAddXdyAttribute}{Move all
14059     occurrences of \string\GlsAddXdyCounters\space before the first
14060     instance of \string\GlsAddXdyAttribute}%
14061 }
14062 \ifglxsindy
14063   \newcommand*\@glsaddxdyattribute[2]{%

```

```

14064 \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
14065 \string"#2#1\string"}%
14066 \expandafter\toks@\expandafter{\@xdylocref}%
14067 \edef\@xdylocref{\the\toks@ ^^J%
14068 (markup-locref
14069 :open \string"\glstildechar n%
14070 \expandafter\string\csname glsX#2X#1\endcsname
14071 \string" ^^J
14072 :close \string"\string" ^^J
14073 :attr \string"#2#1\string")}%
14074 \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
14075 \setentrycounter[##1]{##2}\csname #1\endcsname{##2}%
14076 }%
14077 }
14078 \newcommand*\GlsAddXdyAttribute[1]{%
14079 \ifx\@xdyattributelist\@empty
14080 \edef\@xdyattributelist{#1}%
14081 \else
14082 \edef\@xdyattributelist{\@xdyattributelist,#1}%
14083 \fi
14084 \@for\@this@counter:=\@xdycounters\do{%
14085 \protected@edef\gls@do@addxdyattribute{%
14086 \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
14087 }
14088 \gls@do@addxdyattribute
14089 }%
14090 \let\GlsAddXdyCounters\@disabled@glsaddxdycounters
14091 }
14092 \@onlypremakeg\GlsAddXdyAttribute
14093 \else
14094 \newcommand*\GlsAddXdyAttribute[1]{%
14095 \glsnoxywarning\GlsAddXdyAttribute}
14096 \fi
14097 \ifglsxindy
14098 \newcommand*{\@gls@addpredefinedattributes}{%
14099 \GlsAddXdyAttribute{glsnumberformat}
14100 \GlsAddXdyAttribute{textrm}
14101 \GlsAddXdyAttribute{textsf}
14102 \GlsAddXdyAttribute{texttt}
14103 \GlsAddXdyAttribute{textbf}
14104 \GlsAddXdyAttribute{textmd}
14105 \GlsAddXdyAttribute{textit}
14106 \GlsAddXdyAttribute{textup}
14107 \GlsAddXdyAttribute{textsl}
14108 \GlsAddXdyAttribute{textsc}
14109 \GlsAddXdyAttribute{emph}
14110 \GlsAddXdyAttribute{glshypernumber}
14111 \GlsAddXdyAttribute{hyperrm}
14112 \GlsAddXdyAttribute{hypersf}

```

```

14113 \GlsAddXdyAttribute{hypertt}
14114 \GlsAddXdyAttribute{hyperbf}
14115 \GlsAddXdyAttribute{hypermd}
14116 \GlsAddXdyAttribute{hyperit}
14117 \GlsAddXdyAttribute{hyperup}
14118 \GlsAddXdyAttribute{hypersl}
14119 \GlsAddXdyAttribute{hypersc}
14120 \GlsAddXdyAttribute{hyperemph}
14121 \GlsAddXdyAttribute{glsignore}
14122 }
14123 \else
14124 \let\@gls@addpredefinedattributes\relax
14125 \fi
14126 \def\@xdyuseralphabets{}
14127 \ifglxsindy
14128 \newcommand*{\GlsAddXdyAlphabet}[2]{%
14129 \edef\@xdyuseralphabets{%
14130 \@xdyuseralphabets ^^J
14131 (define-alphabet "#1" (#2))}}
14132 \else
14133 \newcommand*{\GlsAddXdyAlphabet}[2]{%
14134 \glsnoxywarning\GlsAddXdyAlphabet}
14135 \fi
14136 \ifglxsindy
14137 \newcommand*{\@gls@xdy@locationlist}{%
14138 roman-page-numbers,%
14139 Roman-page-numbers,%
14140 arabic-page-numbers,%
14141 alpha-page-numbers,%
14142 Alpha-page-numbers,%
14143 Appendix-page-numbers,%
14144 arabic-section-numbers%
14145 }
14146 \protected@edef\@gls@roman{\@roman{0}\string"
14147 \string"roman-numbers-lowercase\string" :sep \string"}}%
14148 \@onelevel@sanitize\@gls@roman
14149 \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
14150 :sep \string"}%
14151 \@onelevel@sanitize\@tmp
14152 \ifx\@tmp\@gls@roman
14153 \expandafter
14154 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
14155 \string"roman-numbers-lowercase\string"%
14156 }%
14157 \else
14158 \expandafter
14159 \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
14160 :sep \string"\@gls@roman\string"%
14161 }%

```

```

14162 \fi
14163 \expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
14164   \string"roman-numbers-uppercase\string"%
14165 }%
14166 \expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
14167   \string"arabic-numbers\string"%
14168 }%
14169 \expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
14170   \string"alpha\string"%
14171 }%
14172 \expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
14173   \string"ALPHA\string"%
14174 }%
14175 \expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
14176   \string"ALPHA\string"
14177   :sep \string"@glsAlphacompositor\string"
14178   \string"arabic-numbers\string"%
14179 }
14180 \expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
14181   \string"arabic-numbers\string"
14182   :sep \string"\glscompositor\string"
14183   \string"arabic-numbers\string"%
14184 }%
14185 \def\@xdyuserlocationdefs{}
14186 \def\@xdyuserlocationnames{}
14187 \fi
14188 \ifglsxindy
14189 \newcommand\@xdycrossrefhook{}
14190 \fi
14191 \ifglsxindy
14192 \newcommand*{\GlsAddXdyLocation}[3][[]]{%
14193   \def\@gls@tmp{#1}%
14194   \ifx\@gls@tmp\@empty
14195     \edef\@xdyuserlocationdefs{%
14196       \@xdyuserlocationdefs ^^J%
14197       (define-location-class \string"#2\string"^^J\space\space
14198       \space(:sep \string"{}\glsopenbrace\string" #3
14199       :sep \string"\glsclosebrace\string"))
14200   }%
14201   \else
14202     \edef\@xdyuserlocationdefs{%
14203       \@xdyuserlocationdefs ^^J%
14204       (define-location-class \string"#2\string"^^J\space\space
14205       \space(:sep "\glsopenbrace"
14206       #1
14207       :sep "\glsclosebrace\glsopenbrace" #3
14208       :sep "\glsclosebrace"))
14209   }%
14210 \fi

```

```

14211 \edef\@xdyuserlocationnames{%
14212 \@xdyuserlocationnames^^J\space\space\space
14213 \string"#2\string"}%
14214 }
14215 \@onlypremakeg\GlsAddXdyLocation
14216 \else
14217 \newcommand*\GlsAddXdyLocation[2]{%
14218 \glsnnoxindywarning\GlsAddXdyLocation}
14219 \fi
14220 \ifglxindy
14221 \def\@xdylocationclassorder{^^J\space\space\space
14222 \string"roman-page-numbers\string"^^J\space\space\space
14223 \string"arabic-page-numbers\string"^^J\space\space\space
14224 \string"arabic-section-numbers\string"^^J\space\space\space
14225 \string"alpha-page-numbers\string"^^J\space\space\space
14226 \string"Roman-page-numbers\string"^^J\space\space\space
14227 \string"Alpha-page-numbers\string"^^J\space\space\space
14228 \string"Appendix-page-numbers\string"
14229 \@xdyuserlocationnames^^J\space\space\space
14230 \string"see\string"
14231 }
14232 \fi
14233 \ifglxindy
14234 \newcommand*\GlsSetXdyLocationClassOrder[1]{%
14235 \def\@xdylocationclassorder{#1}}
14236 \else
14237 \newcommand*\GlsSetXdyLocationClassOrder[1]{%
14238 \glsnnoxindywarning\GlsSetXdyLocationClassOrder}
14239 \fi
14240 \ifglxindy
14241 \def\@xdysortrules{}
14242 \fi
14243 \ifglxindy
14244 \newcommand*\GlsAddSortRule[2]{%
14245 \expandafter\toks@\expandafter{\@xdysortrules}%
14246 \protected@edef\@xdysortrules{\the\toks@ ^^J
14247 (sort-rule \string"#1\string" \string"#2\string")}%
14248 }
14249 \else
14250 \newcommand*\GlsAddSortRule[2]{%
14251 \glsnnoxindywarning\GlsAddSortRule}
14252 \fi
14253 \ifglxindy
14254 \def\@xdyrequiredstyles{tex}
14255 \fi
14256 \ifglxindy
14257 \newcommand*\GlsAddXdyStyle[1]{%
14258 \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
14259 \else

```

```

14260 \newcommand*\GlsAddXdyStyle[1]{%
14261   \glsnoxywarning\GlsAddXdyStyle}
14262 \fi
14263 \ifglxindy
14264   \newcommand*\GlsSetXdyStyles[1]{%
14265     \edef\@xdyrequiredstyles{#1}}
14266 \else
14267   \newcommand*\GlsSetXdyStyles[1]{%
14268     \glsnoxywarning\GlsSetXdyStyles}
14269 \fi
14270 \newcommand*\findrootlanguage{}
14271 \def\@xdy@language#1#2{}
14272 \ifglxindy
14273   \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
14274     \ifglossaryexists{#1}{%
14275       \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
14276     }{%
14277       \PackageError{glossaries}{Can't set language type for
14278         glossary type '#1' --- no such glossary}{%
14279         You have specified a glossary type that doesn't exist}}
14280 \else
14281   \newcommand*\GlsSetXdyLanguage[2][]{%
14282     \glsnoxywarning\GlsSetXdyLanguage}
14283 \fi
14284 \def\@gls@codepage#1#2{}
14285 \ifglxindy
14286   \newcommand*\GlsSetXdyCodePage[1]{%
14287     \renewcommand*\@gls@codepage{#1}%
14288   }
14289   \AtBeginDocument{%
14290     \ifx\@gls@codepage\@empty
14291       \@ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
14292     \fi
14293   }
14294 \else
14295   \newcommand*\GlsSetXdyCodePage[1]{%
14296     \glsnoxywarning\GlsSetXdyCodePage}
14297 \fi
14298 \ifglxindy
14299   \ifglxindy@glxnumbers
14300     \def\@xdy@lettergroups{(define-letter-group
14301       \string"glxnumbers\string"^^J\space\space\space
14302       :prefixes (\string"0\string" \string"1\string"
14303       \string"2\string" \string"3\string" \string"4\string"
14304       \string"5\string" \string"6\string" \string"7\string"
14305       \string"8\string" \string"9\string")^^J\space\space\space
14306       \@xdynumbergrouporder)}
14307   \else
14308     \def\@xdy@lettergroups{}

```

```

14309 \fi
14310 \fi
14311 \newcommand*\GlsAddLetterGroup[2]{%
14312   \expandafter\toks@\expandafter{\@xdylettergroups}%
14313   \protected@edef\@xdylettergroups{\the\toks@^^J%
14314   (define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
14315 }%
14316 \newcommand*\forallglossaries[3][\@glo@types]{%
14317   \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
14318 }
14319
14320 \newcommand*\forallacronyms[2]{%
14321   \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
14322 }
14323 \newcommand*\forallglsentries[3][\glsdefaulttype]{%
14324   \edef\@glo@list{\csname glolist@#1\endcsname}%
14325   \@for#2:=\@glo@list\do
14326   {%
14327     \ifdefempty{#2}{#3}%
14328   }%
14329 }
14330 \newcommand*\forallglsentries[3][\@glo@types]{%
14331   \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
14332   {%
14333     \forallglsentries[\@this@glo@]{#2}{#3}%
14334   }%
14335 }
14336 \newcommand{\ifglossaryexists}{%
14337   \@ifstar\s@ifglossaryexists\@ifglossaryexists
14338 }
14339 \newcommand{\@ifglossaryexists}[3]{%
14340   \ifcsundef{@glo@type@#1@out}{#3}{#2}%
14341 }
14342 \newcommand{\s@ifglossaryexists}[3]{%
14343   \ifcsundef{glolist@#1}{#3}{#2}%
14344 }
14345 \newcommand*\glsdetoklabel[1]{#1}
14346 \newcommand{\ifglsentryexists}[3]{%
14347   \ifcsundef{glo@glsdetoklabel{#1}@name}{#3}{#2}%
14348 }
14349 \newcommand*\ifglsused[3]{%
14350   \ifbool{glo@glsdetoklabel{#1}@flag}{#2}{#3}%
14351 }
14352 \newcommand{\glsdoifexists}[2]{%
14353   \ifglsentryexists{#1}{#2}{%
14354     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
14355       has not been defined}{You need to define a glossary entry before you
14356       can use it.}}%
14357 }

```

```

14358 \newcommand{\glsdoifnoexists}[2]{%
14359   \ifglentryexists{#1}{%
14360     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
14361       been defined}{-}{#2}%
14362 }
14363 \newcommand{\glsdoifexistsorwarn}[2]{%
14364   \ifglentryexists{#1}{#2}{%
14365     \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
14366       has not been defined}%
14367   }%
14368 }
14369 \newcommand{\glsdoifexistsordo}[3]{%
14370   \ifglentryexists{#1}{#2}{%
14371     \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
14372       has not been defined}{You need to define a glossary entry before you
14373       can use it.}%
14374     #3%
14375   }%
14376 }
14377 \newcommand{\doifglossarynoexistsordo}[3]{%
14378   \ifglossaryexists*{#1}%
14379   {%
14380     \PackageError{glossaries}{Glossary type ‘#1’ already exists}{-}%
14381     #3%
14382   }%
14383   {#2}%
14384 }
14385 \newrobustcmd{\ifglshaschildren}[3]{%
14386   \glsdoifexists{#1}%
14387   {%
14388     \def\do@glshaschildren{#3}%
14389     \edef\@gls@thislabel{\glsdetoklabel{#1}}%
14390     \expandafter\forlentries\expandafter
14391     [\csname glo@\@gls@thislabel @type\endcsname]
14392     {\glo@label}%
14393     {%
14394       \letcs\glo@parent{glo@\glo@label @parent}%
14395       \ifdefequal\@gls@thislabel\glo@parent
14396       {%
14397         \def\do@glshaschildren{#2}%
14398         \@endfortrue
14399       }%
14400     }%
14401   }%
14402   \do@glshaschildren
14403 }%
14404 }
14405 \newcommand{\ifglshasparent}[3]{%
14406   \glsdoifexists{#1}%

```



```

14407  {%
14408    \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
14409  }%
14410 }
14411 \newcommand*{\ifglshasdesc}[3]{%
14412   \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
14413   {#3}%
14414   {#2}%
14415 }
14416 \newcommand*{\ifglshasdescsuppressed}[3]{%
14417   \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
14418   {#2}%
14419   {#3}%
14420 }
14421 \newrobustcmd*{\ifglshassymbol}[3]{%
14422   \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
14423   \ifdefempty\@glo@symbol
14424   {#3}%
14425   {%
14426     \ifdefequal\@glo@symbol\@gls@default@value
14427     {#3}%
14428     {#2}%
14429   }%
14430 }
14431 \newrobustcmd*{\ifglshaslong}[3]{%
14432   \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
14433   \ifdefempty\@glo@long
14434   {#3}%
14435   {%
14436     \ifdefequal\@glo@long\@gls@default@value
14437     {#3}%
14438     {#2}%
14439   }%
14440 }
14441 \newrobustcmd*{\ifglshasshort}[3]{%
14442   \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
14443   \ifdefempty\@glo@short
14444   {#3}%
14445   {%
14446     \ifdefequal\@glo@short\@gls@default@value
14447     {#3}%
14448     {#2}%
14449   }%
14450 }
14451 \newrobustcmd*{\ifglshasfield}[4]{%
14452   \glsdoifexists{#2}%
14453   {%
14454     \letcs{\@glo@thisvalue}{glo@\glsdetoklabel{#2}@#1}%
14455     \ifdef\@glo@thisvalue

```

```

14456 {%
14457     \ifdefempty\@glo@thisvalue
14458     {%
14459         #4%
14460     }%
14461     {%
14462         \ifdefequal\@glo@thisvalue\@gls@default@value
14463         {%
14464             #4%
14465         }%
14466         {%
14467             \let\glscurrentfieldvalue\@glo@thisvalue
14468             #3%
14469         }%
14470     }%
14471 }%
14472 {%
14473     \@gls@fetchfield{\@gls@thisfield}{#1}%
14474     \ifdef\@gls@thisfield
14475     {%
14476         \letcs{\@glo@thisvalue}{glo\@glsdetoklabel{#2}\@gls@thisfield}%
14477         \ifdefempty\@glo@thisvalue
14478         {%
14479             #4%
14480         }%
14481         {%
14482             \ifdefequal\@glo@thisvalue\@gls@default@value
14483             {%
14484                 #4%
14485             }%
14486             {%
14487                 \let\glscurrentfieldvalue\@glo@thisvalue
14488                 #3%
14489             }%
14490         }%
14491     }%
14492     {%
14493         \GlossariesWarning{Unknown entry field '#1'}%
14494         #4%
14495     }%
14496 }%
14497 }%
14498 }
14499 \newcommand*{\glscurrentfieldvalue}{}
14500 \newcommand*{\@glo@types}{,}
14501 \newcommand*{\@gls@provide@newglossary}%
14502     \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
14503     \let\@gls@provide@newglossary\relax
14504 }

```

```

14505 \newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
14506   \csgdef{gls@#1@entryfmt}{#2}%
14507 }
14508 \newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
14509 \newcommand*{\@gls@forbidtext}[1]{%
14510   \ifboolexpr{test {\ifdefstring{#1}{tex}}
14511     or test {\ifdefstring{#1}{TEX}}}
14512   {%
14513     \def#1{nottex}%
14514     \PackageError{glossaries}%
14515       {Forbidden '.tex' extension replaced with '.nottex'}%
14516       {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
14517         Don't use '.tex' as an extension for a temporary file.}%
14518   }%
14519   {%
14520   }%
14521 }
14522 \newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
14523 \def\@gls@gobbleopt[#1]{}
14524 \newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
14525 \newcommand*{\s@newglossary}[2]{%
14526   \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
14527 }
14528 \newcommand*{\ns@newglossary}[5][glg]{%
14529   \doifglossarynoexistsordo{#2}%
14530   {%
14531     \ifundef\glsdefaulttype
14532     {%
14533       \gdef\glsdefaulttype{#2}%
14534     }{}%
14535     \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
14536     \expandafter\gdef\csname glolist@#2\endcsname{,}%
14537     \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
14538     \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
14539     \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
14540     \expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
14541     \expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
14542     \expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname
14543     \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
14544     \@gls@provide@newglossary
14545     \protected@write\auxout{{\string\@newglossary{#2}{#1}{#3}{#4}}}%
14546     \ifcsundef{gls@#2@entryfmt}%
14547     {%
14548       \defglsentryfmt[#2]{\glsentryfmt}%
14549     }%
14550     {}%
14551     \@gls@defsortcount{#2}%
14552     \@ifnextchar[{\@gls@setcounter{#2}}%
14553     {\@gls@setcounter{#2}[\glscounter]}%

```

```

14554 }%
14555 {%
14556   \gls@gobbleopt
14557 }%
14558 }
14559 \newcommand*{\altnewglossary}[3]{%
14560   \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
14561 }
14562 \@onlypreamble{\newglossary}
14563 \@onlypremakeg\newglossary
14564 \newcommand*{\@newglossary}[4]{%
14565   \def\@gls@setcounter#1[#2]{%
14566     \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
14567     \ifglxindy
14568       \GlsAddXdyCounters{#2}%
14569     \fi
14570 }
14571 \newcommand*{\@gls@getcounter}[1]{%
14572   \csname @glotype@#1@counter\endcsname
14573 }
14574 \glsdefmain
14575 \@gls@do@acronymsdef
14576 \@gls@do@symbolsdef
14577 \@gls@do@numbersdef
14578 \@gls@do@indexdef
14579 \newcommand*{\newignoredglossary}[1]{%
14580   \ifdefempty\@ignored@glossaries
14581     {%
14582       \edef\@ignored@glossaries{#1}%
14583     }%
14584     {%
14585       \eappto\@ignored@glossaries{, #1}%
14586     }%
14587     \csgdef{glolist@#1}{,}%
14588     \ifcsundef{gls@#1@entryfmt}%
14589     {%
14590       \defglsentryfmt[#1]{\glsentryfmt}%
14591     }%
14592     {}%
14593     \ifdefempty\@gls@nohyperlist
14594     {%
14595       \renewcommand*{\@gls@nohyperlist}{#1}%
14596     }%
14597     {%
14598       \eappto\@gls@nohyperlist{, #1}%
14599     }%
14600 }
14601 \newcommand*{\@ignored@glossaries}{}
14602 \newcommand*{\ifignoredglossary}[3]{%

```

```

14603 \edef\@gls@igtype{#1}%
14604 \expandafter\DTLifinlist\expandafter
14605   {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
14606 }
14607 \define@key{glossentry}{name}{%
14608 \def\@glo@name{#1}%
14609 }
14610 \define@key{glossentry}{description}{%
14611 \def\@glo@desc{#1}%
14612 }
14613 \define@key{glossentry}{descriptionplural}{%
14614 \def\@glo@descplural{#1}%
14615 }
14616 \define@key{glossentry}{sort}{%
14617 \def\@glo@sort{#1}}
14618 \define@key{glossentry}{text}{%
14619 \def\@glo@text{#1}%
14620 }
14621 \define@key{glossentry}{plural}{%
14622 \def\@glo@plural{#1}%
14623 }
14624 \define@key{glossentry}{first}{%
14625 \def\@glo@first{#1}%
14626 }
14627 \define@key{glossentry}{firstplural}{%
14628 \def\@glo@firstplural{#1}%
14629 }
14630 \newcommand*{\@gls@default@value}{\relax}
14631 \define@key{glossentry}{symbol}{%
14632 \def\@glo@symbol{#1}%
14633 }
14634 \define@key{glossentry}{symbolplural}{%
14635 \def\@glo@symbolplural{#1}%
14636 }
14637 \define@key{glossentry}{type}{%
14638 \def\@glo@type{#1}}
14639 \define@key{glossentry}{counter}{%
14640 \ifcsundef{c@#1}%
14641   {%
14642     \PackageError{glossaries}%
14643     {There is no counter called ‘#1’}%
14644     {%
14645       The counter key should have the name of a valid counter
14646       as its value%
14647     }%
14648   }%
14649   {%
14650     \def\@glo@counter{#1}%
14651   }%

```

```

14652 }
14653 \define@key{glossentry}{see}{%
14654   \gls@set@xr@key{see}{\@glo@see}{#1}%
14655 }
14656 \newcommand*{\gls@set@xr@key}[3]{%
14657   \renewcommand*{\gls@xr@key}{#1}%
14658   \gls@checkseeallowed
14659   \def#2{#3}%
14660   \@glo@seeautonumberlist
14661 }
14662 \newcommand*{\gls@xr@key}{see}
14663 \newcommand*{\gls@checkseeallowed}{%
14664   \@gls@see@noindex
14665 }
14666 \newcommand*{\gls@checkseeallowed@preambleonly}{%
14667   \GlossariesWarning{glossaries}%
14668   {'\gls@xr@key' key doesn't have any effect when used in the document
14669     environment. Move the definition to the preamble
14670     after \string\makeglossaries\space
14671     or \string\makenoidxglossaries}%
14672 }
14673 \define@key{glossentry}{parent}{%
14674   \def\@glo@parent{#1}}
14675 \define@choicekey{glossentry}{nonumberlist}{%
14676   [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
14677 }%
14678   \ifcase\gls@nonumberlist@nr\relax
14679     \def\@glo@prefix{\glsnonextpages}%
14680     \@gls@savenonumberlist{true}%
14681   \else
14682     \def\@glo@prefix{\glsnextpages}%
14683     \@gls@savenonumberlist{false}%
14684   \fi
14685 }
14686 \newcommand*{\@gls@savenonumberlist}[1]{%
14687   \newcommand*{\@gls@initnonumberlist}{}%
14688   \newcommand*{\@gls@storenonumberlist}[1]{%
14689     \newcommand*{\@gls@enablesavenonumberlist}{%
14690       \renewcommand*{\@gls@initnonumberlist}{%
14691         \undef\@glo@nonumberlist
14692       }%
14693       \renewcommand*{\@gls@savenonumberlist}[1]{%
14694         \def\@glo@nonumberlist{##1}%
14695       }%
14696       \renewcommand*{\@gls@storenonumberlist}[1]{%
14697         \ifdef\@glo@nonumberlist
14698           {%
14699             \cslet{glo@\glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
14700           }%

```

```

14701     {}%
14702 }%
14703 \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
14704 }
14705 \define@key{glossentry}{user1}{%
14706   \def\@glo@useri{#1}%
14707 }
14708 \define@key{glossentry}{user2}{%
14709   \def\@glo@userii{#1}%
14710 }
14711 \define@key{glossentry}{user3}{%
14712   \def\@glo@useriii{#1}%
14713 }
14714 \define@key{glossentry}{user4}{%
14715   \def\@glo@useriv{#1}%
14716 }
14717 \define@key{glossentry}{user5}{%
14718   \def\@glo@userv{#1}%
14719 }
14720 \define@key{glossentry}{user6}{%
14721   \def\@glo@uservi{#1}%
14722 }
14723 \define@key{glossentry}{short}{%
14724   \def\@glo@short{#1}%
14725 }
14726 \define@key{glossentry}{shortplural}{%
14727   \def\@glo@shortpl{#1}%
14728 }
14729 \define@key{glossentry}{long}{%
14730   \def\@glo@long{#1}%
14731 }
14732 \define@key{glossentry}{longplural}{%
14733   \def\@glo@longpl{#1}%
14734 }
14735 \newcommand*{\@glsnoname}{%
14736   \PackageError{glossaries}{name key required in
14737     \string\newglossaryentry\space for entry '\@glo@label'}{You
14738     haven't specified the entry name}}
14739 \newcommand*{\@glsnodelsc{%
14740   \PackageError{glossaries}
14741     {%
14742       description key required in \string\newglossaryentry\space
14743       for entry '\@glo@label'%
14744     }%
14745   {%
14746     You haven't specified the entry description%
14747   }%
14748 }%
14749 \newcommand*{\@glsdefaultplural}{%

```

```

14750 \newcommand*{\@gls@missingnumberlist}[1]{%
14751   ??%
14752   \ifglssavenumberlist
14753     \GlossariesWarning{Missing number list for entry ‘#1’.
14754     Maybe makeglossaries + rerun required}%
14755   \else
14756     \PackageError{glossaries}%
14757     {Package option ‘savenumberlist=true’ required}%
14758     {%
14759     You must use the ‘savenumberlist’ package option
14760     to reference location lists.%
14761     }%
14762   \fi
14763 }
14764 \newcommand*{\@gls@defaultsort}{\@glo@name}
14765 \newcount\gls@level
14766 \newcommand{\@gls@noexpand@field}[3]{%
14767   \expandafter\global\expandafter
14768   \let\csname glo@#1@#2\endcsname#3%
14769 }
14770 \newcommand{\@gls@noexpand@fields}[4]{%
14771   \ifcsdef{gls@assign@#3@field}
14772     {%
14773       \ifdefequal{#4}{\@gls@default@value}%
14774       {%
14775         \edef\@gls@value{\expandonce{#1}}%
14776         \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
14777       }%
14778       {%
14779         \csuse{gls@assign@#3@field}{#2}{#4}%
14780       }%
14781     }%
14782     {%
14783       \ifdefequal{#4}{\@gls@default@value}%
14784       {%
14785         \edef\@gls@value{\expandonce{#1}}%
14786         \@gls@noexpand@field{#2}{#3}{\@gls@value}%
14787       }%
14788       {%
14789         \@gls@noexpand@field{#2}{#3}{#4}%
14790       }%
14791     }%
14792 }
14793 \newcommand{\@gls@expand@field}[3]{%
14794   \expandafter
14795   \protected@xdef\csname glo@#1@#2\endcsname{#3}%
14796 }
14797 \newcommand{\@gls@expand@fields}[4]{%
14798   \ifcsdef{gls@assign@#3@field}

```



```

14799 {%
14800   \ifdefequal{#4}{\@gls@default@value}%
14801   {%
14802     \edef\@gls@value{\expandonce{#1}}%
14803     \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
14804   }%
14805   {%
14806     \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
14807     {%
14808       \@@gls@expand@field{#2}{#3}{#4}%
14809     }%
14810     {%
14811       \csuse{gls@assign@#3@field}{#2}{#4}%
14812     }%
14813   }%
14814 }%
14815 {%
14816   \ifdefequal{#4}{\@gls@default@value}%
14817   {%
14818     \@@gls@expand@field{#2}{#3}{#1}%
14819   }%
14820   {%
14821     \@@gls@expand@field{#2}{#3}{#4}%
14822   }%
14823 }%
14824 }
14825 \def\@gls@expandonce{\expandonce}
14826 \def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
14827   \def\@gls@tmp{#1}%
14828   \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
14829 }
14830 \let\gls@assign@field\@gls@expand@fields
14831 \newcommand*\@gls@expandfields{%
14832   \let\gls@assign@field\@gls@expand@fields
14833 }
14834 \newcommand*\@gls@noexpandfields{%
14835   \let\gls@assign@field\@gls@noexpand@fields
14836 }
14837 \newrobustcmd{\newglossaryentry}[2]{%
14838   \glsdoifnoexists{#1}%
14839   {%
14840     \gls@defglossaryentry{#1}{#2}%
14841   }%
14842 }
14843 \newcommand*\@gls@defdocnewglossaryentry{%
14844   \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
14845   \let\newglossaryentry\newglossaryentry
14846 }
14847 \newrobustcmd{\provideglossaryentry}[2]{%

```

```

14848 \ifglentryexists{#1}%
14849 {}%
14850 {%
14851   \gls@defglossaryentry{#1}{#2}%
14852 }%
14853 }
14854 \onlypreamble{\provideglossaryentry}
14855 \newrobustcmd{\new@glossaryentry}[2]{%
14856   \ifundef\@gls@deffile
14857   {%
14858     \global\newwrite\@gls@deffile
14859     \immediate\openout\@gls@deffile=\jobname.glsdefs
14860   }%
14861   {}%
14862   \ifglentryexists{#1}{}%
14863   {%
14864     \gls@defglossaryentry{#1}{#2}%
14865   }%
14866   \@gls@writedef{#1}%
14867 }
14868 \AtBeginDocument{\gls@begindocdefs}
14869 \AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
14870 \newcommand*\@gls@begindocdefs{%
14871   \@gls@enablesavenonumberlist
14872   \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
14873   \makeatletter
14874   \InputIfFileExists{\jobname.glsdefs}{\@gls@restreat}{\@gls@restreat}
14875   \undef\@gls@restreat
14876   \undef\@gls@restreat
14877   \gls@defdocnewglossaryentry
14878 }
14879 \newcommand*\@gls@writedef[1]{%
14880   \immediate\write\@gls@deffile
14881   {%
14882     \string\ifglentryexists{#1}{}\glspercentchar^^J%
14883     \expandafter\@gobble\string{\glspercentchar^^J%
14884       \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
14885       \expandafter\@gobble\string{\glspercentchar%
14886     }%
14887     \@for\@gls@map:=\@gls@keymap\do
14888     {%
14889       \letcs\glo@value{glo\@glsdetoklabel{#1}\@expandafter\@secondoftwo\@gls@map}%
14890       \ifdef\glo@value
14891       {%
14892         \@onelevel@sanitize\glo@value
14893         \immediate\write\@gls@deffile
14894         {%
14895           \expandafter\@firstoftwo\@gls@map
14896           =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%

```

```

14897      \glspercentchar
14898    }%
14899  }%
14900  {}%
14901 }%
14902 \glswritedefhook
14903 \immediate\write\@gls@deffile
14904 {%
14905     \glspercentchar^^J%
14906     \expandafter\@gobble\string\}\glspercentchar^^J%
14907     \expandafter\@gobble\string\}\glspercentchar%
14908 }%
14909 }
14910 \newcommand*{\@gls@keymap}{%
14911   {name}{name},%
14912   {sort}{sortvalue},% unescaped sort value
14913   {type}{type},%
14914   {first}{first},%
14915   {firstplural}{firstpl},%
14916   {text}{text},%
14917   {plural}{plural},%
14918   {description}{desc},%
14919   {descriptionplural}{descplural},%
14920   {symbol}{symbol},%
14921   {symbolplural}{symbolplural},%
14922   {user1}{useri},%
14923   {user2}{userii},%
14924   {user3}{useriii},%
14925   {user4}{useriv},%
14926   {user5}{userv},%
14927   {user6}{uservi},%
14928   {long}{long},%
14929   {longplural}{longpl},%
14930   {short}{short},%
14931   {shortplural}{shortpl},%
14932   {counter}{counter},%
14933   {parent}{parent}%
14934 }
14935 \newcommand*{\@gls@fetchfield}[2]{%
14936   \edef\@gls@thisval{#2}%
14937   \@for\@gls@map:=\@gls@keymap\do{%
14938     \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
14939     \ifdefequal{\@this@key}{\@gls@thisval}%
14940     {%
14941       \edef#1{\expandafter\@secondoftwo\@gls@map}%
14942       \@endfortrue
14943     }%
14944   }%
14945 }%

```

```

14946 }
14947 \newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
14948 \newcommand*{\@sglsaddstoragekey}[1]{%
14949   \key@ifundefined{glossentry}{#1}%
14950   {%
14951     \expandafter\newcommand\expandafter*\expandafter
14952       {\csname gls@assign@#1@field\endcsname}[2]{%
14953         \@gls@expand@field{##1}{#1}{##2}%
14954       }%
14955   }%
14956   {}%
14957   \@glsaddstoragekey{#1}%
14958 }
14959 \newcommand*{\@glsaddstoragekey}[3]{%
14960   \key@ifundefined{glossentry}{#1}%
14961   {%
14962     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
14963     \appto\@gls@keymap{,{#1}{#1}}%
14964     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
14965     \appto\@newglossaryentryposthook{%
14966       \letcs{\@glo@tmp}{@glo@#1}%
14967       \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
14968     }%
14969     \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
14970   }%
14971   {}%
14972   \PackageError{glossaries}{Key ‘#1’ already exists}{}%
14973   }%
14974 }
14975 \newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
14976 \newcommand*{\@sglsaddkey}[1]{%
14977   \key@ifundefined{glossentry}{#1}%
14978   {%
14979     \expandafter\newcommand\expandafter*\expandafter
14980       {\csname gls@assign@#1@field\endcsname}[2]{%
14981         \@gls@expand@field{##1}{#1}{##2}%
14982       }%
14983     }%
14984     {}%
14985     \@glsaddkey{#1}%
14986 }
14987 \newcommand*{\@glsaddkey}[7]{%
14988   \key@ifundefined{glossentry}{#1}%
14989   {%
14990     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
14991     \appto\@gls@keymap{,{#1}{#1}}%
14992     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
14993     \appto\@newglossaryentryposthook{%
14994       \letcs{\@glo@tmp}{@glo@#1}%

```

```

14995     \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
14996 }%
14997 \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
14998 \newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
14999 \ifcsdef{@gls@user@#1@}%
15000 {%
15001     \PackageError{glossaries}%
15002     {Can't define '\string#5' as helper command
15003     '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
15004     }%
15005 }%
15006 {%
15007     \expandafter\newcommand\expandafter*\expandafter
15008     {\csname @gls@user@#1\endcsname}[2][\%
15009         \new@ifnextchar[%
15010             {\csuse{@gls@user@#1@}{##1}{##2}}%
15011             {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
15012     \csdef{@gls@user@#1@}##1##2[##3]{%
15013         \@gls@field@link{##1}{##2}{#3{##2}##3}%
15014     }%
15015     \newrobustcmd*{#5}{%
15016         \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
15017     }%
15018     \ifcsdef{@Gls@user@#1@}%
15019     {%
15020         \PackageError{glossaries}%
15021         {Can't define '\string#6' as helper command
15022         '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
15023         }%
15024     }%
15025     {%
15026         \expandafter\newcommand\expandafter*\expandafter
15027         {\csname @Gls@user@#1\endcsname}[2][\%
15028             \new@ifnextchar[%
15029                 {\csuse{@Gls@user@#1@}{##1}{##2}}%
15030                 {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
15031         \csdef{@Gls@user@#1@}##1##2[##3]{%
15032             \@gls@field@link{##1}{##2}{#4{##2}##3}%
15033         }%
15034         \newrobustcmd*{#6}{%
15035             \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
15036         }%
15037         \ifcsdef{@GLS@user@#1@}%
15038         {%
15039             \PackageError{glossaries}%
15040             {Can't define '\string#7' as helper command
15041             '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
15042             }%
15043         }%

```

```

15044   {%
15045       \expandafter\newcommand\expandafter*\expandafter
15046       {\csname @GLS@user@#1\endcsname}[2] [] {%
15047           \new@ifnextchar[%
15048               {\csuse{@GLS@user@#1@}{##1}{##2}}}%
15049               {\csuse{@GLS@user@#1@}{##1}{##2} []}}}%
15050       \csdef{@GLS@user@#1@}##1##2[##3]{%
15051           \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}}%
15052       }%
15053       \newrobustcmd*{#7}{%
15054           \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
15055       }%
15056   }%
15057   {%
15058       \PackageError{glossaries}{Key ‘#1’ already exists}{}%
15059   }%
15060 }
15061 \newcommand{\glsfieldxdef}[3]{%
15062   \glsdoifexists{#1}%
15063   {%
15064       \edef\@glo@label{\glsdetoklabel{#1}}%
15065       \ifcsdef{glo@\@glo@label @#2}%
15066       {%
15067           \protected@csxdef{glo@\@glo@label @#2}{#3}%
15068       }%
15069       {%
15070           \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15071       }%
15072   }%
15073 }
15074 \newcommand{\glsfielddedef}[3]{%
15075   \glsdoifexists{#1}%
15076   {%
15077       \edef\@glo@label{\glsdetoklabel{#1}}%
15078       \ifcsdef{glo@\@glo@label @#2}%
15079       {%
15080           \protected@csedef{glo@\@glo@label @#2}{#3}%
15081       }%
15082       {%
15083           \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15084       }%
15085   }%
15086 }
15087 \newcommand{\glsfielddgdef}[3]{%
15088   \glsdoifexists{#1}%
15089   {%
15090       \edef\@glo@label{\glsdetoklabel{#1}}%
15091       \ifcsdef{glo@\@glo@label @#2}%
15092       {%

```

```

15093     \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
15094 }%
15095 {%
15096     \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15097 }%
15098 }%
15099 }
15100 \newcommand{\glsfielddef}[3]{%
15101     \glsdoifexists{#1}%
15102     {%
15103         \edef\@glo@label{\glsdetoklabel{#1}}%
15104         \ifcsdef{glo@\@glo@label @#2}%
15105             {%
15106                 \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
15107             }%
15108             {%
15109                 \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15110             }%
15111         }%
15112     }
15113 \newcommand{\glsfieldfetch}[3]{%
15114     \glsdoifexists{#1}%
15115     {%
15116         \edef\@glo@label{\glsdetoklabel{#1}}%
15117         \ifcsdef{glo@\@glo@label @#2}%
15118             {%
15119                 \letcs#3{glo@\@glo@label @#2}%
15120             }%
15121             {%
15122                 \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15123             }%
15124         }%
15125     }
15126 \newcommand{\ifglsfieldeq}[5]{%
15127     \glsdoifexists{#1}%
15128     {%
15129         \edef\@glo@label{\glsdetoklabel{#1}}%
15130         \ifcsdef{glo@\@glo@label @#2}%
15131             {%
15132                 \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
15133             }%
15134             {%
15135                 \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15136             }%
15137         }%
15138     }
15139 \newcommand{\ifglsfielddefeq}[5]{%
15140     \glsdoifexists{#1}%
15141     {%

```

```

15142 \edef\@glo@label{\glsdetoklabel{#1}}%
15143 \ifcsdef{glo@\@glo@label @#2}%
15144 {%
15145     \expandafter\ifdefstrequal
15146     \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
15147 }%
15148 {%
15149     \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15150 }%
15151 }%
15152 }
15153 \newcommand{\ifglsfieldcseq}[5]{%
15154 \glsdoifexists{#1}%
15155 {%
15156     \edef\@glo@label{\glsdetoklabel{#1}}%
15157     \ifcsdef{glo@\@glo@label @#2}%
15158     {%
15159         \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
15160     }%
15161     {%
15162         \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
15163     }%
15164 }%
15165 }
15166 \newcommand*\@gls@writedefhook{}
15167 \newcommand*\@gls@assign@desc}[1]{%
15168 \gls@assign@field{#1}{desc}{\@glo@desc}%
15169 \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
15170 }
15171 \newcommand{\longnewglossaryentry}[3]{%
15172 \glsdoifnoexists{#1}%
15173 {%
15174     \bgroup
15175     \let\@org@newglossaryentryprehook\@newglossaryentryprehook
15176     \long\def\@newglossaryentryprehook{%
15177         \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
15178         \@org@newglossaryentryprehook
15179     }%
15180     \renewcommand*\@gls@assign@desc}[1]{%
15181         \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
15182         \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
15183     }
15184     \gls@defglossaryentry{#1}{#2}%
15185     \egroup
15186 }
15187 }
15188 \@onlypreamble{\longnewglossaryentry}
15189 \newcommand{\longprovideglossaryentry}[3]{%
15190 \ifglsentryexists{#1}{}%

```



```

15191 {\longnewglossaryentry{#1}{#2}{#3}}%
15192 }
15193 \@onlypreamble{\longprovideglossaryentry}
15194 \newcommand{\gls@defglossaryentry}[2]{%
15195   \let\GlsSetQuote\gls@nosetquote
15196   \edef\@glo@label{\glsdetoklabel{#1}}%
15197   \let\glslabel\@glo@label
15198   \let\@glo@name\@glsnname
15199   \let\@glo@desc\@glsndesc
15200   \let\@glo@descplural\@gls@default@value
15201   \let\@glo@type\@gls@default@value
15202   \let\@glo@symbol\@gls@default@value
15203   \let\@glo@symbolplural\@gls@default@value
15204   \let\@glo@text\@gls@default@value
15205   \let\@glo@plural\@gls@default@value
15206   \let\@glo@first\@gls@default@value
15207   \let\@glo@firstplural\@gls@default@value
15208   \let\@glo@sort\@gls@default@value
15209   \let\@glo@counter\@gls@default@value
15210   \def\@glo@see{}%
15211   \def\@glo@parent{}%
15212   \def\@glo@prefix{}%
15213   \@gls@initnonnumberlist
15214   \def\@glo@useri{}%
15215   \def\@glo@userii{}%
15216   \def\@glo@useriii{}%
15217   \def\@glo@useriv{}%
15218   \def\@glo@userv{}%
15219   \def\@glo@uservi{}%
15220   \def\@glo@short{}%
15221   \def\@glo@shortpl{}%
15222   \def\@glo@long{}%
15223   \def\@glo@longpl{}%
15224   \@newglossaryentryprehook
15225   \setkeys{glossentry}{#2}%
15226   \ifundef\glsdefaulttype
15227   {%
15228     \PackageError{glossaries}%
15229     {No default glossary type (have you used ‘nomain’ by mistake?)}%
15230     {If you use package option ‘nomain’ you must define
15231      a new glossary before you can define entries}%
15232   }%
15233   {%
15234     \gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
15235     \edef\@glo@type{\glsentrytype{\@glo@label}}%
15236     \ifcsundef{glolist@\@glo@type}%
15237     {%
15238       \PackageError{glossaries}%
15239       {Glossary type ‘\@glo@type’ has not been defined}%

```

```

15240      {You need to define a new glossary type, before making entries
15241      in it}%
15242  }%
15243  {%
15244      \ifignoredglossary\@glo@type
15245      {%
15246          \ifx\@glo@desc\@glsnodesc
15247              \let\@glo@desc\@empty
15248          \fi
15249      }%
15250      {%
15251      }%
15252      \protected@edef\@glo@list@\csname glolist@\@glo@type\endcsname}%
15253      \expandafter\xdef\csname glolist@\@glo@type\endcsname{%
15254          \@glo@list@\@glo@label},}%
15255  }%
15256  \gls@level=0\relax
15257  \ifx\@glo@parent\@empty
15258      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15259  \else
15260      \ifdefequal\@glo@label\@glo@parent%
15261      {%
15262          \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
15263          \def\@glo@parent{}%
15264          \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15265      }%
15266      {%
15267          \ifglsentryexists{\@glo@parent}%
15268          {%
15269              \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
15270                  \@glo@parent}%
15271              \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
15272              \advance\gls@level by 1\relax
15273              \ifx\@glo@name\@glsnoname
15274                  \expandafter\let\expandafter\@glo@name
15275                      \csname glo@\@glo@parent @name\endcsname
15276                  \ifx\@glo@plural\@gls@default@value
15277                      \expandafter\let\expandafter\@glo@plural
15278                          \csname glo@\@glo@parent @plural\endcsname
15279                  \fi
15280              \fi
15281          }%
15282      {%
15283          \PackageError{glossaries}%
15284          {%
15285              Invalid parent ‘\@glo@parent’
15286              for entry ‘\@glo@label’ - parent doesn’t exist%
15287          }%
15288      {%

```

```

15289         Parent entries must be defined before their children%
15290     }%
15291     \def\@glo@parent{}%
15292     \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
15293 }%
15294 }%
15295 \fi
15296 \expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
15297 \gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
15298 \letcs\@glo@sort{glo@\@glo@label @sortvalue}%
15299 \gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
15300 \expandafter\gls@assign@field\expandafter
15301     {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
15302     {\@glo@label}{plural}{\@glo@plural}%
15303 \expandafter\gls@assign@field\expandafter
15304     {\csname glo@\@glo@label @text\endcsname}%
15305     {\@glo@label}{first}{\@glo@first}%
15306 \ifx\@glo@first\@gls@default@value
15307     \expandafter\gls@assign@field\expandafter
15308         {\csname glo@\@glo@label @plural\endcsname}%
15309         {\@glo@label}{firstpl}{\@glo@firstplural}%
15310 \else
15311     \expandafter\gls@assign@field\expandafter
15312         {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
15313         {\@glo@label}{firstpl}{\@glo@firstplural}%
15314 \fi
15315 \ifcsundef{@glo@type@\@glo@type @counter}%
15316 {%
15317     \def\@glo@defaultcounter{\glscounter}%
15318 }%
15319 {%
15320     \letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
15321 }%
15322 \gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
15323 \gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
15324 \gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
15325 \gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
15326 \gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
15327 \gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
15328 \gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
15329 \gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
15330 \gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
15331 \gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
15332 \gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
15333 \ifx\@glo@name\@gls@name
15334     \gls@name
15335     \let\@glo@name\@gls@default@value
15336 \fi
15337 \gls@assign@field{}{\@glo@label}{name}{\@glo@name}%

```

```

15338 \ifcsundef{glo@\@glo@label @numberlist}%
15339 {%
15340   \csxdef{glo@\@glo@label @numberlist}{%
15341     \noexpand\@gls@missingnumberlist{\@glo@label}}%
15342   }%
15343   {%
15344     \@gls@storenonumberlist{\@glo@label}%
15345     \def\@glo@@desc{\@glo@first}%
15346     \ifx\@glo@desc\@glo@@desc
15347       \let\@glo@desc\@glo@first
15348     \fi
15349     \ifx\@glo@desc\@glsnodesc
15350       \@glsnodesc
15351       \let\@glo@desc\@gls@default@value
15352     \fi
15353     \gls@assign@desc{\@glo@label}%
15354     \@gls@defsort{\@glo@type}{\@glo@label}%
15355     \def\@glo@@symbol{\@glo@text}%
15356     \ifx\@glo@symbol\@glo@@symbol
15357       \let\@glo@symbol\@glo@text
15358     \fi
15359     \gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
15360     \expandafter
15361       \gls@assign@field\expandafter
15362         {\csname glo@\@glo@label @symbol\endcsname}
15363         {\@glo@label}{symbolplural}{\@glo@symbolplural}%
15364     \expandafter\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
15365       \noexpand\global
15366       \noexpand\let\expandafter\noexpand
15367       \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
15368     }%
15369     \expandafter\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
15370       \noexpand\global
15371       \noexpand\let\expandafter\noexpand
15372       \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
15373     }%
15374     \csname glo@\@glo@label @flagfalse\endcsname
15375     \@glo@autosee
15376     \ifignoredglossary\@glo@type
15377     {%
15378       \csdef{glo@\@glo@label @index}{}%
15379     }
15380     {%
15381       \do@glo@storeentry{\@glo@label}%
15382     }%
15383     \@newglossaryentry@defcounters
15384     \@newglossaryentryposthook
15385   }
15386 \newcommand*{\@glo@autosee}{%

```

```

15387 \ifdefvoid\@glo@see{}%
15388 {%
15389     \protected@edef\@do@glsee{%
15390         \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
15391         \noexpand\expandafter\noexpand\@glsee\noexpand\@glo@list{\@glo@label}}%
15392     \@do@glsee
15393 }%
15394 \@glo@autoseehook
15395 }%
15396 \newcommand*{\@glo@autoseehook}{}
15397 \newcommand*{\@newglossaryentryprehook}{}
15398 \newcommand*{\@newglossaryentryposthook}{}
15399 \newcommand*{\@newglossaryentry@defcounters}{}
15400 \newcommand*{\glsmoveentry}[2]{%
15401     \edef\@glo@thislabel{\glsdetoklabel{#1}}%
15402     \edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
15403     \def\glo@list{,%}%
15404     \forglseentries[\glo@type]{\glo@label}%
15405     {%
15406         \ifdefequal\@glo@thislabel\glo@label
15407             {\eappto\glo@list{\glo@label,}}%
15408         }%
15409     \cslet{glolist@\glo@type}{\glo@list}%
15410     \csdef{glo@\@glo@thislabel @type}{#2}%
15411 }
15412 \ifglxindy
15413     \newcommand*{\@glossaryentryfield}{\string\glossentry}
15414 \else
15415     \newcommand*{\@glossaryentryfield}{\string\glossentry}
15416 \fi
15417 \ifglxindy
15418     \newcommand*{\@glossarysubentryfield}{%
15419         \string\subglossentry}
15420 \else
15421     \newcommand*{\@glossarysubentryfield}{%
15422         \string\subglossentry}
15423 \fi
15424 \newcommand{\@glo@storeentry}[1]{%
15425     \edef\@glo@esclabel{#1}%
15426     \@gls@checkmkidxchars\@glo@esclabel
15427     \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
15428     \@gls@checkmkidxchars\@glo@sort
15429     \@gls@checkmkidxchars\@glo@prefix
15430     \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
15431     \ifglxindy
15432         \ifx\@glo@parent\@empty
15433             \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15434                 (\string"\@glo@sort\string" %
15435                 \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %

```

```

15436     }%
15437   \else
15438     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15439       \csname glo@\@glo@parent @index\endcsname
15440       (\string"\@glo@sort\string" %
15441       \string"\@glo@prefix\@glossarysubentryfield
15442       {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
15443     }%
15444   \fi
15445 \else
15446   \ifx\@glo@parent\@empty
15447     \@onelevel@sanitize\@glo@prefix
15448     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15449       \@glo@sort\@gls@actualchar\@glo@prefix
15450       \@glossaryentryfield{\@glo@esclabel}%
15451     }%
15452   \else
15453     \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
15454       \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
15455       \@glo@sort\@gls@actualchar\@glo@prefix
15456       \@glossarysubentryfield
15457       {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
15458     }%
15459   \fi
15460 \fi
15461 }
15462 \AtBeginDocument{%
15463   \@ifpackageloaded{amsmath}%
15464   {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
15465   {}%
15466 }
15467 \newcommand*{\@gls@ifnotmeasuring}[1]{%
15468   \ifmeasuring@
15469   \else
15470     #1%
15471   \fi
15472 }
15473 \newcommand*\gls@ifnotmeasuring[1]{#1}
15474 \def\@gls@patchtabularx#1\hbox#2#3!!{%
15475   \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
15476 }
15477 \newcommand*\glspatchtabularx{%
15478   \ifdef\TX@trial
15479   {%
15480     \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
15481     \let\glspatchtabularx\relax
15482   }%
15483   {}%
15484 }

```

```

15485 \newcommand*{\glsreset}[1]{%
15486   \gls@ifnotmeasuring
15487   {%
15488     \glsdoifexists{#1}%
15489     {%
15490       \@glsreset{#1}%
15491     }%
15492   }%
15493 }
15494 \newcommand*{\glslocalreset}[1]{%
15495   \gls@ifnotmeasuring
15496   {%
15497     \glsdoifexists{#1}%
15498     {%
15499       \@glslocalreset{#1}%
15500     }%
15501   }%
15502 }
15503 \newcommand*{\glsunset}[1]{%
15504   \gls@ifnotmeasuring
15505   {%
15506     \glsdoifexists{#1}%
15507     {%
15508       \@glsunset{#1}%
15509     }%
15510   }%
15511 }
15512 \newcommand*{\glslocalunset}[1]{%
15513   \gls@ifnotmeasuring
15514   {%
15515     \glsdoifexists{#1}%
15516     {%
15517       \@glslocalunset{#1}%
15518     }%
15519   }%
15520 }
15521 \newcommand*{\@glslocalunset}{\@@glslocalunset}
15522 \newcommand*{\@@glslocalunset}[1]{%
15523   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
15524 }
15525 \newcommand*{\@glsunset}{\@@glsunset}
15526 \newcommand*{\@@glsunset}[1]{%
15527   \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
15528 }
15529 \newcommand*{\@glslocalreset}{\@@glslocalreset}
15530 \newcommand*{\@@glslocalreset}[1]{%
15531   \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
15532 }
15533 \newcommand*{\@glsreset}{\@@glsreset}

```

```

15534 \newcommand*{\@glsreset}[1]{%
15535   \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
15536 }
15537 \newcommand*{\glsresetall}[1][\@glo@types]{%
15538   \forallglsentries[#1]{\@glsentry}%
15539   {%
15540     \glsreset{\@glsentry}%
15541   }%
15542 }
15543 \newcommand*{\glslocalresetall}[1][\@glo@types]{%
15544   \forallglsentries[#1]{\@glsentry}%
15545   {%
15546     \glslocalreset{\@glsentry}%
15547   }%
15548 }
15549 \newcommand*{\glsunsetall}[1][\@glo@types]{%
15550   \forallglsentries[#1]{\@glsentry}%
15551   {%
15552     \glsunset{\@glsentry}%
15553   }%
15554 }
15555 \newcommand*{\glslocalunsetall}[1][\@glo@types]{%
15556   \forallglsentries[#1]{\@glsentry}%
15557   {%
15558     \glslocalunset{\@glsentry}%
15559   }%
15560 }
15561 \newcommand*{\@@newglossaryentry@defcounters}{%
15562   \csdef{glo@\@glo@label @currcount}{0}%
15563   \csdef{glo@\@glo@label @prevcount}{0}%
15564 }
15565 \newcommand*{\glsenableentrycount}{%
15566   \let\@newglossaryentry@defcounters\@@newglossaryentry@defcounters
15567   \renewcommand*{\gls@defdocnewglossaryentry}{%
15568     \renewcommand*\newglossaryentry[2]{%
15569       \PackageError{glossaries}{\string\newglossaryentry\space
15570         may only be used in the preamble when entry counting has
15571         been activated}{If you use \string\glsenableentrycount\space
15572         you must place all entry definitions in the preamble not in
15573         the document environment}%
15574     }%
15575   }%
15576   \newcommand*{\glsentrycurrcount}[1]{%
15577     \ifcsundef{glo@glsdetoklabel{##1}@currcount}%
15578     {0}{\@gls@entry@field{##1}{currcount}}%
15579   }%
15580   \newcommand*{\glsentryprevcount}[1]{%
15581     \ifcsundef{glo@glsdetoklabel{##1}@prevcount}%
15582     {0}{\@gls@entry@field{##1}{prevcount}}%

```



```

15583 }%
15584 \renewcommand*{\@glsunset}[1]{%
15585   \@@glsunset{##1}%
15586   \@gls@increment@currcount{##1}%
15587 }%
15588 \renewcommand*{\@glslocalunset}[1]{%
15589   \@@glslocalunset{##1}%
15590   \@gls@local@increment@currcount{##1}%
15591 }%
15592 \renewcommand*{\@glsreset}[1]{%
15593   \@@glsreset{##1}%
15594   \csgdef{glo@glsdetoklabel{##1}@currcount}{0}%
15595 }%
15596 \renewcommand*{\@glslocalreset}[1]{%
15597   \@@glslocalreset{##1}%
15598   \csdef{glo@glsdetoklabel{##1}@currcount}{0}%
15599 }%
15600 \def\@cgl@s@##1##2[##3]{%
15601   \ifnum\glsentryprevcount{##2}=1\relax
15602     \cgl@sformat{##2}{##3}%
15603     \glsunset{##2}%
15604   \else
15605     \@gls@{##1}{##2}[##3]%
15606   \fi
15607 }%
15608 \def\@cgl spl@##1##2[##3]{%
15609   \ifnum\glsentryprevcount{##2}=1\relax
15610     \cgl splformat{##2}{##3}%
15611     \glsunset{##2}%
15612   \else
15613     \@glspl@{##1}{##2}[##3]%
15614   \fi
15615 }%
15616 \def\@cGls@##1##2[##3]{%
15617   \ifnum\glsentryprevcount{##2}=1\relax
15618     \cGlsformat{##2}{##3}%
15619     \glsunset{##2}%
15620   \else
15621     \@Gls@{##1}{##2}[##3]%
15622   \fi
15623 }%
15624 \def\@cGlspl@##1##2[##3]{%
15625   \ifnum\glsentryprevcount{##2}=1\relax
15626     \cGlsplformat{##2}{##3}%
15627     \glsunset{##2}%
15628   \else
15629     \@Glspl@{##1}{##2}[##3]%
15630   \fi
15631 }%

```

```

15632 \AtEndDocument{\@gls@write@entrycounts}%
15633 \renewcommand*{\@gls@entry@count}[2]{%
15634   \csgdef{glo@glstdetoklabel{##1}@prevcount}{##2}%
15635 }%
15636 \let\glsenableentrycount\relax
15637 }
15638 \@onlypreamble\glsenableentrycount
15639 \newcommand*{\@gls@increment@currcount}[1]{%
15640   \csxdef{glo@glstdetoklabel{##1}@currcount}{%
15641     \number\numexpr\glsentrycurrcount{##1}+1}%
15642 }
15643 \newcommand*{\@gls@local@increment@currcount}[1]{%
15644   \csedef{glo@glstdetoklabel{##1}@currcount}{%
15645     \number\numexpr\glsentrycurrcount{##1}+1}%
15646 }
15647 \newcommand*{\@gls@write@entrycounts}{%
15648   \immediate\write\@auxout
15649     {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
15650   \forallglsentries{\@glsentry}{%
15651     \ifglsused{\@glsentry}%
15652     {\immediate\write\@auxout
15653       {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
15654     {}%
15655   }%
15656 }
15657 \newcommand*{\@gls@entry@count}[2]{%
15658 \newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
15659 \newcommand*{\@cgl}[2][{}]{%
15660   \new@ifnextchar[{\@cgl@{##1}{##2}}{\@cgl@{##1}{##2}[]}%
15661 }
15662 \def\@cgl@#1#2[#3]{%
15663   \GlossariesWarning{\string\cgl\space is defaulting to
15664     \string\gls\space since you haven't enabled entry counting}%
15665   \@gls@{##1}{##2}[##3]}
15666 }
15667 \newcommand*{\cglformat}[2]{%
15668   \ifglshaslong{##1}{\glsentrylong{##1}}{\glsentryfirst{##1}}#2%
15669 }
15670 \newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
15671 \newcommand*{\@cGls}[2][{}]{%
15672   \new@ifnextchar[{\@cGls@{##1}{##2}}{\@cGls@{##1}{##2}[]}%
15673 }
15674 \def\@cGls@#1#2[#3]{%
15675   \GlossariesWarning{\string\cGls\space is defaulting to
15676     \string\Gls\space since you haven't enabled entry counting}%
15677   \@Gls@{##1}{##2}[##3]}
15678 }
15679 \newcommand*{\cGlsformat}[2]{%
15680   \ifglshaslong{##1}{\Glsentrylong{##1}}{\Glsentryfirst{##1}}#2%

```

```

15681 }
15682 \newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
15683 \newcommand*{\@cglspl}[2][{}]{%
15684   \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}[]}]%
15685 }
15686 \def\@cglspl@#1#2[#3]{%
15687   \GlossariesWarning{\string\cglspl\space is defaulting to
15688     \string\glspl\space since you haven't enabled entry counting}%
15689   \@glspl@{#1}{#2}[#3]%
15690 }
15691 \newcommand*{\cglsplformat}[2]{%
15692   \ifglshaslong{#1}{\glentrylongpl{#1}}{\glentryfirstplural{#1}}#2%
15693 }
15694 \newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
15695 \newcommand*{\@cGlspl}[2][{}]{%
15696   \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[]}]%
15697 }
15698 \def\@cGlspl@#1#2[#3]{%
15699   \GlossariesWarning{\string\cGlspl\space is defaulting to
15700     \string\Glspl\space since you haven't enabled entry counting}%
15701   \@Glspl@{#1}{#2}[#3]%
15702 }
15703 \newcommand*{\cGlsplformat}[2]{%
15704   \ifglshaslong{#1}{\Glentrylongpl{#1}}{\Glentryfirstplural{#1}}#2%
15705 }
15706 \newcommand*{\loadglsentries}[2][\@gls@default]{%
15707   \let\@gls@default\glsdefaulttype
15708   \def\glsdefaulttype{#1}\input{#2}%
15709   \let\glsdefaulttype\@gls@default
15710 }
15711 \@onlypreamble{\loadglsentries}
15712 \newcommand*{\glstextformat}[1]{#1}
15713 \newcommand*{\glentryfmt}{%
15714   \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
15715 }
15716 \newcommand*{\@@gls@default@entryfmt}[2]{%
15717   \ifdefempty\glscustomtext
15718     {%
15719       \glsifplural
15720       {%
15721         \glscapscase
15722         {%
15723           \ifglused\glslabel
15724             {%
15725               #2{\glentryplural{\glslabel}}%
15726               {\glentrydescplural{\glslabel}}%
15727               {\glentrysymbolplural{\glslabel}}{\glsinsert}%
15728             }%
15729             {%

```

```

15730      #1{\glsentryfirstplural{\glslabel}}%
15731      {\glsentrydescplural{\glslabel}}%
15732      {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15733  }%
15734 }%
15735 {%
15736   \ifglused\glslabel
15737   {%
15738     \ifbool{glscompatible-3.07}%
15739     {%
15740       \protected@edef\@glo@etext{%
15741         #2{\glsentryplural{\glslabel}}%
15742         {\glsentrydescplural{\glslabel}}%
15743         {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15744       \xmakefirstuc\@glo@etext
15745     }%
15746     {%
15747       #2{\Glsentryplural{\glslabel}}%
15748       {\glsentrydescplural{\glslabel}}%
15749       {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15750     }%
15751   }%
15752   {%
15753     \ifbool{glscompatible-3.07}%
15754     {%
15755       \protected@edef\@glo@etext{%
15756         #1{\glsentryfirstplural{\glslabel}}%
15757         {\glsentrydescplural{\glslabel}}%
15758         {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15759       \xmakefirstuc\@glo@etext
15760     }%
15761     {%
15762       #1{\Glsentryfirstplural{\glslabel}}%
15763       {\glsentrydescplural{\glslabel}}%
15764       {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
15765     }%
15766   }%
15767 }%
15768 {%
15769   \ifglused\glslabel
15770   {%
15771     \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
15772     {\glsentrydescplural{\glslabel}}%
15773     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
15774   }%
15775   {%
15776     \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
15777     {\glsentrydescplural{\glslabel}}%
15778     {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%

```

```

15779     }%
15780 }%
15781 }%
15782 {%
15783   \glscapscase
15784   {%
15785     \ifglused\glslabel
15786     {%
15787       #2{\glsentrytext{\glslabel}}%
15788       {\glsentrydesc{\glslabel}}%
15789       {\glsentrysymbol{\glslabel}}{\glsinsert}%
15790     }%
15791     {%
15792       #1{\glsentryfirst{\glslabel}}%
15793       {\glsentrydesc{\glslabel}}%
15794       {\glsentrysymbol{\glslabel}}{\glsinsert}%
15795     }%
15796   }%
15797   {%
15798     \ifglused\glslabel
15799     {%
15800       \ifbool{glscompatible-3.07}%
15801       {%
15802         \protected@edef\@glo@etext{%
15803           #2{\glsentrytext{\glslabel}}%
15804           {\glsentrydesc{\glslabel}}%
15805           {\glsentrysymbol{\glslabel}}{\glsinsert}}%
15806         \xmakefirstuc\@glo@etext
15807       }%
15808       {%
15809         #2{\Glsentrytext{\glslabel}}%
15810         {\glsentrydesc{\glslabel}}%
15811         {\glsentrysymbol{\glslabel}}{\glsinsert}%
15812       }%
15813     }%
15814     {%
15815       \ifbool{glscompatible-3.07}%
15816       {%
15817         \protected@edef\@glo@etext{%
15818           #1{\glsentryfirst{\glslabel}}%
15819           {\glsentrydesc{\glslabel}}%
15820           {\glsentrysymbol{\glslabel}}{\glsinsert}}%
15821         \xmakefirstuc\@glo@etext
15822       }%
15823       {%
15824         #1{\Glsentryfirst{\glslabel}}%
15825         {\glsentrydesc{\glslabel}}%
15826         {\glsentrysymbol{\glslabel}}{\glsinsert}%
15827       }%

```

```

15828     }%
15829 }%
15830 {%
15831     \ifglused\glslabel
15832     {%
15833         \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}}%
15834         {\glsentrydesc{\glslabel}}}%
15835         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
15836     }%
15837     {%
15838         \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}}%
15839         {\glsentrydesc{\glslabel}}}%
15840         {\glsentrysymbol{\glslabel}}{\glsinsert}}}%
15841     }%
15842 }%
15843 }%
15844 }%
15845 {%
15846     \ifglused{\glslabel}%
15847     {%
15848         #2{\glscustomtext}%
15849         {\glsentrydesc{\glslabel}}}%
15850         {\glsentrysymbol{\glslabel}}{}%
15851     }%
15852     {%
15853         #1{\glscustomtext}%
15854         {\glsentrydesc{\glslabel}}}%
15855         {\glsentrysymbol{\glslabel}}{}%
15856     }%
15857 }%
15858 }
15859 \newcommand*{\glsgenentryfmt}{%
15860     \ifdefempty\glscustomtext
15861     {%
15862         \glsifplural
15863         {%
15864             \glscapscase
15865             {%
15866                 \ifglused\glslabel
15867                 {%
15868                     \glsentryplural{\glslabel}\glsinsert
15869                 }%
15870                 {%
15871                     \glsentryfirstplural{\glslabel}\glsinsert
15872                 }%
15873             }%
15874         }%
15875         \ifglused\glslabel
15876         {%

```

```

15877         \Glsentryplural{\glslabel}\glsinsert
15878     }%
15879     {%
15880         \Glsentryfirstplural{\glslabel}\glsinsert
15881     }%
15882 }%
15883 {%
15884     \ifglsused\glslabel
15885     {%
15886         \mfirstucMakeUppercase
15887         {\Glsentryplural{\glslabel}\glsinsert}%
15888     }%
15889     {%
15890         \mfirstucMakeUppercase
15891         {\Glsentryfirstplural{\glslabel}\glsinsert}%
15892     }%
15893 }%
15894 }%
15895 {%
15896     \glscapscase
15897     {%
15898         \ifglsused\glslabel
15899         {%
15900             \Glsentrytext{\glslabel}\glsinsert
15901         }%
15902         {%
15903             \Glsentryfirst{\glslabel}\glsinsert
15904         }%
15905     }%
15906     {%
15907         \ifglsused\glslabel
15908         {%
15909             \Glsentrytext{\glslabel}\glsinsert
15910         }%
15911         {%
15912             \Glsentryfirst{\glslabel}\glsinsert
15913         }%
15914     }%
15915     {%
15916         \ifglsused\glslabel
15917         {%
15918             \mfirstucMakeUppercase{\Glsentrytext{\glslabel}\glsinsert}%
15919         }%
15920         {%
15921             \mfirstucMakeUppercase{\Glsentryfirst{\glslabel}\glsinsert}%
15922         }%
15923     }%
15924 }%
15925 }%

```

```

15926 {%
15927   \glscustomtext\glsinsert
15928 }%
15929 }
15930 \newcommand*{\glsgenacfmt}{%
15931   \ifdefempty\glscustomtext
15932   {%
15933     \ifglssused\glslabel
15934     {%
15935       \glsifplural
15936       {%
15937         \glscapscase
15938         {%
15939           \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
15940         }%
15941         {%
15942           \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
15943         }%
15944         {%
15945           \mfirstucMakeUppercase
15946             {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
15947         }%
15948       }%
15949     }%
15950     \glscapscase
15951     {%
15952       \acronymfont{\glsentryshort{\glslabel}}\glsinsert
15953     }%
15954     {%
15955       \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
15956     }%
15957     {%
15958       \mfirstucMakeUppercase
15959         {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
15960     }%
15961   }%
15962 }%
15963 {%
15964   \glsifplural
15965   {%
15966     \glscapscase
15967     {%
15968       \genplacrfullformat{\glslabel}{\glsinsert}%
15969     }%
15970     {%
15971       \Genplacrfullformat{\glslabel}{\glsinsert}%
15972     }%
15973     {%
15974       \mfirstucMakeUppercase

```



```

15975         {\genplacrfullformat{\glslabel}{\glsinsert}}}%
15976     }%
15977 }%
15978 {%
15979     \glscapscase
15980     {%
15981         \genacrfullformat{\glslabel}{\glsinsert}%
15982     }%
15983     {%
15984         \Genacrfullformat{\glslabel}{\glsinsert}%
15985     }%
15986     {%
15987         \mfirstucMakeUppercase
15988         {\genacrfullformat{\glslabel}{\glsinsert}}}%
15989     }%
15990 }%
15991 }%
15992 }%
15993 {%
15994     \glscustomtext
15995 }%
15996 }
15997 \newcommand*{\genacrfullformat}[2]{%
15998     \glsentrylong{#1}#2\space
15999     (\protect\firstacronymfont{\glsentryshort{#1}})%
16000 }
16001 \newcommand*{\Genacrfullformat}[2]{%
16002     \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
16003     \xmakefirstuc\gls@text
16004 }
16005 \newcommand*{\genplacrfullformat}[2]{%
16006     \glsentrylongpl{#1}#2\space
16007     (\protect\firstacronymfont{\glsentryshortpl{#1}})%
16008 }
16009 \newcommand*{\Genplacrfullformat}[2]{%
16010     \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
16011     \xmakefirstuc\gls@text
16012 }
16013 \newcommand*{\glsdisplayfirst}[4]{#1#4}
16014 \newcommand*{\glsdisplay}[4]{#1#4}
16015 \newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
16016     \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
16017     Use \string\defglsentryfmt\space instead}%
16018     \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
16019     \edef\@gls@doentrydef{%
16020         \noexpand\defglsentryfmt[#1]{%
16021             \noexpand\ifcsdef{gls@#1@displayfirst}%
16022             {%
16023                 \noexpand\@gls@default@entryfmt

```

```

16024         {\noexpand\csuse{gls@#1@displayfirst}}%
16025         {\noexpand\csuse{gls@#1@display}}%
16026     }%
16027     {%
16028         \noexpand\@gls@default@entryfmt
16029         {\noexpand\glsdisplayfirst}%
16030         {\noexpand\csuse{gls@#1@display}}%
16031     }%
16032 }%
16033 }%
16034 \@gls@doentrydef
16035 }
16036 \newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
16037   \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
16038   Use \string\defglsentryfmt\space instead}%
16039   \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
16040   \edef\@gls@doentrydef{%
16041     \noexpand\defglsentryfmt[#1]{%
16042       \noexpand\ifcsdef{gls@#1@display}%
16043       {%
16044         \noexpand\@gls@default@entryfmt
16045         {\noexpand\csuse{gls@#1@displayfirst}}%
16046         {\noexpand\csuse{gls@#1@display}}%
16047       }%
16048       {%
16049         \noexpand\@gls@default@entryfmt
16050         {\noexpand\csuse{gls@#1@displayfirst}}%
16051         {\noexpand\glsdisplay}%
16052       }%
16053     }%
16054   }%
16055   \@gls@doentrydef
16056 }
16057 \define@key{glslink}{counter}{%
16058   \ifcsundef{c@#1}%
16059   {%
16060     \PackageError{glossaries}%
16061     {There is no counter called ‘#1’}%
16062     {%
16063       The counter key should have the name of a valid counter
16064       as its value%
16065     }%
16066   }%
16067   {%
16068     \def\@gls@counter{#1}%
16069   }%
16070 }
16071 \define@key{glslink}{format}{%
16072   \def\@glsnumberformat{#1}}

```

```

16073 \define@boolkey{glslink}{hyper}[true]{}
16074 \ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
16075 \define@boolkey{glslink}{local}[true]{}
16076 \newcommand*{\glslinkvar}[3]{#1}
16077 \newcommand*{\glsifhyper}[2]{%
16078   \glslinkvar{#1}{#2}{#1}%
16079   \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
16080     you mean \string\glsifhyperon\space or \string\glslinkvar?}%
16081 }
16082 \newcommand*{\@gls@hyp@opt}[1]{%
16083   \let\glslinkvar\@firstofthree
16084   \let\@gls@hyp@opt@cs#1\relax
16085   \@ifstar{\s@gls@hyp@opt}%
16086   {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
16087 }
16088 \newcommand*{\s@gls@hyp@opt}[1][]{%
16089   \let\glslinkvar\@secondofthree
16090   \@gls@hyp@opt@cs[hyper=false,#1]}
16091 \newcommand*{\p@gls@hyp@opt}[1][]{%
16092   \let\glslinkvar\@thirdofthree
16093   \@gls@hyp@opt@cs[hyper=true,#1]}
16094 \newrobustcmd*{\glslink}{%
16095   \@gls@hyp@opt\@gls@@link
16096 }
16097 \newcommand*{\@gls@@link}[3][]{%
16098   \glsdoifexistsordo{#2}%
16099   {%
16100     \let\do@gls@link@checkfirsthyper\relax
16101     \@gls@link[#1]{#2}{#3}%
16102   }{%
16103     \glstextformat{#3}%
16104   }%
16105   \glspostlinkhook
16106 }
16107 \newcommand*{\glspostlinkhook}{}
16108 \newcommand*{\@gls@link@checkfirsthyper}{%
16109   \ifglssused{\glslabel}%
16110   {%
16111   }%
16112   {%
16113     \gls@checkisacronymlist\glstype
16114     \ifglshyperfirst
16115       \if@glsisacronymlist
16116         \ifglssacrfootnote
16117         \KV@glslink@hyperfalse
16118         \fi
16119       \fi
16120     \else
16121       \KV@glslink@hyperfalse

```

```

16122 \fi
16123 }%
16124 \glslinkcheckfirsthyperhook
16125 }
16126 \newcommand*{\glslinkcheckfirsthyperhook}{}
16127 \newcommand*{\glslinkpostsetkeys}{}
16128 \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
16129 \newcommand*{\do@glsglssdisablehyperinlist}{}%
16130 \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
16131 {\KV@glslink@hyperfalse}{}%
16132 }
16133 \newcommand*{\@gls@setdefault@glslink@opts}{}
16134 \def\@gls@link[#1]#2#3{%
16135 \leavevmode
16136 \edef\glslabel{\glsdetoklabel{#2}}%
16137 \def\@gls@link@opts{#1}%
16138 \let\@gls@link@label\glslabel
16139 \def\@glsnumberformat{glsnumberformat}%
16140 \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
16141 \edef\glstype{\csname glo@\glslabel @type\endcsname}%
16142 \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
16143 \@gls@setdefault@glslink@opts
16144 \do@glsglssdisablehyperinlist
16145 \do@gls@link@checkfirsthyper
16146 \setkeys{glslink}{#1}%
16147 \glslinkpostsetkeys
16148 \@gls@saveentrycounter
16149 \@gls@setsort{\glslabel}%
16150 \@do@wrglossary{#2}%
16151 \ifKV@glslink@hyper
16152 \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
16153 \else
16154 \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
16155 \fi
16156 \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
16157 }
16158 \newcommand*{\glolinkprefix}{glo:}
16159 \def\glsentrycounter{\glscounter}%
16160 \newcommand*{\@gls@saveentrycounter}{}%
16161 \def\@gls@Hcounter{}%
16162 \ifthenelse{\equal{\@gls@counter}{equation}}%
16163 {
16164 \ifcsundef{xatlevel@}%
16165 {%
16166 \edef\theglsentrycounter{\expandafter\noexpand
16167 \csname the\@gls@counter\endcsname}%
16168 }%
16169 {%
16170 \ifx\xatlevel@\@empty

```

```

16171 \edef\theglsentrycounter{\expandafter\noexpand
16172 \csname the\@gls@counter\endcsname}%
16173 \else
16174 \savecounters@
16175 \advance\c@equation by 1\relax
16176 \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
16177 \ifcsundef{theH\@gls@counter}%
16178 {%
16179 \def\@gls@Hcounter{\theglsentrycounter}%
16180 }%
16181 {%
16182 \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
16183 }%
16184 \protected@edef\theHglentrycounter{\@gls@Hcounter}%
16185 \restorecounters@
16186 \fi
16187 }%
16188 }%
16189 {%
16190 \edef\theglsentrycounter{\expandafter\noexpand
16191 \csname the\@gls@counter\endcsname}%
16192 }%
16193 \ifx\@gls@Hcounter\@empty
16194 \ifcsundef{theH\@gls@counter}%
16195 {%
16196 \def\theHglentrycounter{\theglsentrycounter}%
16197 }%
16198 {%
16199 \protected@edef\theHglentrycounter{\expandafter\noexpand
16200 \csname theH\@gls@counter\endcsname}%
16201 }%
16202 \fi
16203 }
16204 \def\@set@glo@numformat#1#2#3#4{%
16205 \expandafter\@glo@check@mkidxrangear#3\@nil
16206 \protected@edef#1{%
16207 \@glo@prefix setentrycounter[#4]{#2}%
16208 \expandafter\string\csname\@glo@suffix\endcsname
16209 }%
16210 \@gls@checkmkidxchars#1%
16211 }
16212 \def\@glo@check@mkidxrangear#1#2\@nil{%
16213 \if#1(\relax
16214 \def\@glo@prefix{(%
16215 \if\relax#2\relax
16216 \def\@glo@suffix{glsnumberformat}%
16217 \else
16218 \def\@glo@suffix{#2}%
16219 \fi

```

```

16220 \else
16221   \if#1)\relax
16222     \def\@glo@prefix{}}}%
16223     \if\relax#2\relax
16224       \def\@glo@suffix{glsnumberformat}%
16225     \else
16226       \def\@glo@suffix{#2}%
16227   \fi
16228 \else
16229   \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
16230 \fi
16231 \fi}
16232 \newcommand*{\@gls@escbsdq}[1]{%
16233   \def\@gls@checkedmkidx{%
16234     \let\gls@xdystring=#1\relax
16235     \@onelevel@sanitize\gls@xdystring
16236     \edef\do@gls@xdycheckbackslash{%
16237       \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
16238       \@backslashchar\@backslashchar\noexpand\null}%
16239     \do@gls@xdycheckbackslash
16240     \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
16241     \def\@gls@checkedmkidx{%
16242       \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
16243       \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
16244       \@for\@gls@tmp:=\gls@protected@pagefmts\do
16245       {%
16246         \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
16247         \@onelevel@sanitize\@gls@sanitized@tmp
16248         \edef\gls@dostubst{%
16249           \noexpand\DTLsubstituteall\noexpand\gls@xdystring
16250           {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
16251         }%
16252         \gls@dostubst
16253       }%
16254     \let#1=\gls@xdystring
16255 }
16256 \newcommand{\@gls@checkmkidxchars}[1]{%
16257   \ifglxsindy
16258     \@gls@escbsdq{#1}%
16259   \else
16260     \def\@gls@checkedmkidx{%
16261       \expandafter\@gls@checkquote#1\@nil""\null
16262       \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16263     \def\@gls@checkedmkidx{%
16264       \expandafter\@gls@checkescquote#1\@nil\""\null
16265       \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16266     \def\@gls@checkedmkidx{%
16267       \expandafter\@gls@checkescactual#1\@nil\?\?\null
16268     \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%

```

```

16269 \def\@gls@checkedmkidx{%
16270 \expandafter\@gls@checkactual#1\@nil??\null
16271 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16272 \def\@gls@checkedmkidx{%
16273 \expandafter\@gls@checkbar#1\@nil||\null
16274 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16275 \def\@gls@checkedmkidx{%
16276 \expandafter\@gls@checkescbar#1\@nil\\|\null
16277 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16278 \def\@gls@checkedmkidx{%
16279 \expandafter\@gls@checklevel#1\@nil!!\null
16280 \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
16281 \fi
16282 }
16283 \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
16284 \newtoks\@gls@tmpb
16285 \def\@gls@checkquote#1"#2"#3\null{%
16286 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16287 \toks@={#1}%
16288 \ifx\null#2\null
16289 \ifx\null#3\null
16290 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16291 \def\@gls@checkquote{\relax}%
16292 \else
16293 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16294 \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
16295 \def\@gls@checkquote{\@gls@checkquote#3\null}%
16296 \fi
16297 \else
16298 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16299 \@gls@quotechar\@gls@quotechar}%
16300 \ifx\null#3\null
16301 \def\@gls@checkquote{\@gls@checkquote#2""\null}%
16302 \else
16303 \def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
16304 \fi
16305 \fi
16306 \@gls@checkquote
16307 }
16308 \def\@gls@checkescquote#1\ "#2\ "#3\null{%
16309 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16310 \toks@={#1}%
16311 \ifx\null#2\null
16312 \ifx\null#3\null
16313 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16314 \def\@gls@checkescquote{\relax}%
16315 \else
16316 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16317 \@gls@quotechar\string"\@gls@quotechar

```

```

16318     \@gls@quotechar\string\" \@gls@quotechar}%
16319     \def\@gls@checkescquote{\@gls@checkescquote#3\null}%
16320     \fi
16321   \else
16322     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16323       \@gls@quotechar\string\" \@gls@quotechar}%
16324     \ifx\null#3\null
16325       \def\@gls@checkescquote{\@gls@checkescquote#2\" \"\null}%
16326     \else
16327       \def\@gls@checkescquote{\@gls@checkescquote#2\"#3\null}%
16328     \fi
16329   \fi
16330 \@@gls@checkescquote
16331 }
16332 \def\@gls@checkescactual#1\?#2\?#3\null{%
16333   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16334   \toks@={#1}%
16335   \ifx\null#2\null
16336     \ifx\null#3\null
16337       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16338       \def\@gls@checkescactual{\relax}%
16339     \else
16340       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16341         \@gls@quotechar\string\" \@gls@actualchar
16342         \@gls@quotechar\string\" \@gls@actualchar}%
16343       \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
16344     \fi
16345   \else
16346     \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16347       \@gls@quotechar\string\" \@gls@actualchar}%
16348     \ifx\null#3\null
16349       \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
16350     \else
16351       \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
16352     \fi
16353   \fi
16354 \@@gls@checkescactual
16355 }
16356 \def\@gls@checkescbar#1\|#2\|#3\null{%
16357   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16358   \toks@={#1}%
16359   \ifx\null#2\null
16360     \ifx\null#3\null
16361       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16362       \def\@gls@checkescbar{\relax}%
16363     \else
16364       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16365         \@gls@quotechar\string\" \@gls@encapchar
16366         \@gls@quotechar\string\" \@gls@encapchar}%

```



```

16367 \def\@gls@checkesbar{\@gls@checkesbar#3\null}%
16368 \fi
16369 \else
16370 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16371 \@gls@quotechar\string"\@gls@encapchar}%
16372 \ifx\null#3\null
16373 \def\@gls@checkesbar{\@gls@checkesbar#2\|\|\null}%
16374 \else
16375 \def\@gls@checkesbar{\@gls@checkesbar#2\|#3\null}%
16376 \fi
16377 \fi
16378 \@gls@checkesbar
16379 }
16380 \def\@gls@checkesclevel#1\!#2\!#3\null{%
16381 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16382 \toks@={#1}%
16383 \ifx\null#2\null
16384 \ifx\null#3\null
16385 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16386 \def\@gls@checkesclevel{\relax}%
16387 \else
16388 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16389 \@gls@quotechar\string"\@gls@levelchar
16390 \@gls@quotechar\string"\@gls@levelchar}%
16391 \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
16392 \fi
16393 \else
16394 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16395 \@gls@quotechar\string"\@gls@levelchar}%
16396 \ifx\null#3\null
16397 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
16398 \else
16399 \def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
16400 \fi
16401 \fi
16402 \@gls@checkesclevel
16403 }
16404 \def\@gls@checkbar#1|#2|#3\null{%
16405 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16406 \toks@={#1}%
16407 \ifx\null#2\null
16408 \ifx\null#3\null
16409 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16410 \def\@gls@checkbar{\relax}%
16411 \else
16412 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16413 \@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
16414 \def\@gls@checkbar{\@gls@checkbar#3\null}%
16415 \fi

```

```

16416 \else
16417 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16418 \@gls@quotechar\@gls@encapchar}%
16419 \ifx\null#3\null
16420 \def\@gls@checkbar{\@gls@checkbar#2||\null}%
16421 \else
16422 \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
16423 \fi
16424 \fi
16425 \@gls@checkbar
16426 }
16427 \def\@gls@checklevel#1!#2!#3\null{%
16428 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16429 \toks@={#1}%
16430 \ifx\null#2\null
16431 \ifx\null#3\null
16432 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16433 \def\@gls@checklevel{\relax}%
16434 \else
16435 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16436 \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
16437 \def\@gls@checklevel{\@gls@checklevel#3\null}%
16438 \fi
16439 \else
16440 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16441 \@gls@quotechar\@gls@levelchar}%
16442 \ifx\null#3\null
16443 \def\@gls@checklevel{\@gls@checklevel#2!!\null}%
16444 \else
16445 \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
16446 \fi
16447 \fi
16448 \@gls@checklevel
16449 }
16450 \def\@gls@checkactual#1?#2?#3\null{%
16451 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16452 \toks@={#1}%
16453 \ifx\null#2\null
16454 \ifx\null#3\null
16455 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16456 \def\@gls@checkactual{\relax}%
16457 \else
16458 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16459 \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
16460 \def\@gls@checkactual{\@gls@checkactual#3\null}%
16461 \fi
16462 \else
16463 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16464 \@gls@quotechar\@gls@actualchar}%

```

```

16465 \ifx\null#3\null
16466 \def\@gls@checkactual{\@gls@checkactual#2??\null}%
16467 \else
16468 \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
16469 \fi
16470 \fi
16471 \@gls@checkactual
16472 }
16473 \def\@gls@xdycheckquote#1"#2"#3\null{%
16474 \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
16475 \toks@={#1}%
16476 \ifx\null#2\null
16477 \ifx\null#3\null
16478 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
16479 \def\@gls@xdycheckquote{\relax}%
16480 \else
16481 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16482 \string"\string"}%
16483 \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
16484 \fi
16485 \else
16486 \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
16487 \string"}%
16488 \ifx\null#3\null
16489 \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
16490 \else
16491 \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
16492 \fi
16493 \fi
16494 \@gls@xdycheckquote
16495 }
16496 \edef\def@gls@xdycheckbackslash{%
16497 \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
16498 ##2\@backslashchar##3\noexpand\null{%
16499 \noexpand\@gls@tmpb=\noexpand\expandafter
16500 {\noexpand\@gls@checkedmkidx}%
16501 \noexpand\toks@={##1}%
16502 \noexpand\ifx\noexpand\null##2\noexpand\null
16503 \noexpand\ifx\noexpand\null##3\noexpand\null
16504 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16505 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
16506 \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
16507 \noexpand\else
16508 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16509 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
16510 \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
16511 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16512 \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
16513 \noexpand\fi

```

```

16514 \noexpand\else
16515 \noexpand\edef\noexpand\@gls@checkedmkidx{%
16516 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
16517 \@backslashchar\@backslashchar}%
16518 \noexpand\ifx\noexpand\null##3\noexpand\null
16519 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16520 \noexpand\@gls@xdycheckbackslash##2\@backslashchar
16521 \@backslashchar\noexpand\null}%
16522 \noexpand\else
16523 \noexpand\def\noexpand\@gls@xdycheckbackslash{%
16524 \noexpand\@gls@xdycheckbackslash##2\@backslashchar
16525 ##3\noexpand\null}%
16526 \noexpand\fi
16527 \noexpand\fi
16528 \noexpand\@gls@xdycheckbackslash
16529 }%
16530 }
16531 \def@gls@xdycheckbackslash
16532
16533 \newlength\gls@tmplen
16534 \newcommand*\glsdohypertarget}[2]{%
16535 \glsshowtarget{#1}%
16536 \settoheight{\gls@tmplen}{#2}%
16537 \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
16538 }
16539 \newcommand*\glsdohyperlink}[2]{%
16540 \glsshowtarget{#1}%
16541 \hyperlink{#1}{#2}%
16542 }
16543 \newcommand*\glsdonohyperlink}[2]{#2}
16544 \ifcsundef{hyperlink}%
16545 {%
16546 \let\@glslink\glsdonohyperlink
16547 }%
16548 {%
16549 \let\@glslink\glsdohyperlink
16550 }
16551 \ifcsundef{hypertarget}%
16552 {%
16553 \let\@glstarget\@secondoftwo
16554 }%
16555 {%
16556 \let\@glstarget\glsdohypertarget
16557 }
16558 \newcommand*\glsdisablehyper}{%
16559 \KV@glslink@hyperfalse
16560 \let\@glslink\glsdonohyperlink
16561 \let\@glstarget\@secondoftwo
16562 }

```

```

16563 \newcommand{\glsenablehyper}{%
16564   \KV@glslink@hypertrue
16565   \let\@glslink\glsdohyperlink
16566   \let\@glstarget\glsdohypertarget
16567 }
16568 \providecommand{\@firstofthree}[3]{#1}
16569 \providecommand{\@secondofthree}[3]{#2}
16570 \newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
16571 \newcommand*{\@gls}[2][{}]{%
16572   \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
16573 }
16574 \def\@gls@#1#2[#3]{%
16575   \glstoifexists{#2}%
16576   {%
16577     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16578     \let\glsifplural\@secondoftwo
16579     \let\glscapscase\@firstofthree
16580     \let\glscustomtext\@empty
16581     \def\glsinsert{#3}%
16582     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16583     \@gls@link[#1]{#2}{\@glo@text}%
16584     \ifKV@glslink@local
16585       \glslocalunset{#2}%
16586     \else
16587       \glsunset{#2}%
16588     \fi
16589   }%
16590   \glspostlinkhook
16591 }
16592 \newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
16593 \newcommand*{\@Gls}[2][{}]{%
16594   \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
16595 }
16596 \def\@Gls@#1#2[#3]{%
16597   \glstoifexists{#2}%
16598   {%
16599     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16600     \let\glsifplural\@secondoftwo
16601     \let\glscapscase\@secondofthree
16602     \let\glscustomtext\@empty
16603     \def\glsinsert{#3}%
16604     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16605     \@gls@link[#1]{#2}{\@glo@text}%
16606     \ifKV@glslink@local
16607       \glslocalunset{#2}%
16608     \else
16609       \glsunset{#2}%
16610     \fi
16611   }%

```

```

16612 \glspostlinkhook
16613 }
16614 \newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
16615 \newcommand*{\@GLS}[2] [] {%
16616   \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}]%
16617 }
16618 \def\@GLS@#1#2[#3] {%
16619   \glsdoifexists{#2}%
16620   {%
16621     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16622     \let\glsifplural\@secondoftwo
16623     \let\glscapscase\@thirdofthree
16624     \let\glscustomtext\@empty
16625     \def\glsinsert{#3}%
16626     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16627     \@gls@link[#1]{#2}{\@glo@text}%
16628     \ifKV@glslink@local
16629       \glslocalunset{#2}%
16630     \else
16631       \glsunset{#2}%
16632     \fi
16633   }%
16634   \glspostlinkhook
16635 }
16636 \newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
16637 \newcommand*{\@glspl}[2] [] {%
16638   \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}]%
16639 }
16640 \def\@glspl@#1#2[#3] {%
16641   \glsdoifexists{#2}%
16642   {%
16643     \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
16644     \let\glsifplural\@firstoftwo
16645     \let\glscapscase\@firstofthree
16646     \let\glscustomtext\@empty
16647     \def\glsinsert{#3}%
16648     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16649     \@gls@link[#1]{#2}{\@glo@text}%
16650     \ifKV@glslink@local
16651       \glslocalunset{#2}%
16652     \else
16653       \glsunset{#2}%
16654     \fi
16655   }%
16656   \glspostlinkhook
16657 }
16658 \newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
16659 \newcommand*{\@Glspl}[2] [] {%
16660   \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}]%

```

```

16661 }
16662 \def\@GLSp1@#1#2[#3]{%
16663   \glsdoifexists{#2}%
16664   {%
16665     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16666     \let\glsifplural\@firstoftwo
16667     \let\glscapscase\@secondofthree
16668     \let\glscustomtext\@empty
16669     \def\glsinsert{#3}%
16670     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16671     \@gl@link[#1]{#2}{\@glo@text}%
16672     \ifKV@gl@link@local
16673       \glslocalunset{#2}%
16674     \else
16675       \glsunset{#2}%
16676     \fi
16677   }%
16678   \glspostlinkhook
16679 }
16680 \newrobustcmd*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
16681 \newcommand*{\@GLSp1}[2][{}]{%
16682   \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
16683 }
16684 \def\@GLSp1@#1#2[#3]{%
16685   \glsdoifexists{#2}%
16686   {%
16687     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16688     \let\glsifplural\@firstoftwo
16689     \let\glscapscase\@thirdofthree
16690     \let\glscustomtext\@empty
16691     \def\glsinsert{#3}%
16692     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16693     \@gl@link[#1]{#2}{\@glo@text}%
16694     \ifKV@gl@link@local
16695       \glslocalunset{#2}%
16696     \else
16697       \glsunset{#2}%
16698     \fi
16699   }%
16700   \glspostlinkhook
16701 }
16702 \newrobustcmd*{\glsdisp}{\@gl@hyp@opt\@glsdisp}
16703 \newcommand*{\@glsdisp}[3][{}]{%
16704   \glsdoifexists{#2}{%
16705     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
16706     \let\glsifplural\@secondoftwo
16707     \let\glscapscase\@firstofthree
16708     \def\glscustomtext{#3}%
16709     \def\glsinsert{}%

```

```

16710 \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
16711 \@gls@link[#1]{#2}{\@glo@text}%
16712 \ifKV@gls@link@local
16713 \glslocalunset{#2}%
16714 \else
16715 \glsunset{#2}%
16716 \fi
16717 }%
16718 \gls@postlinkhook
16719 }
16720 \newcommand*{\@gls@link@nocheckfirsthyper}{%
16721 \newcommand{\@gls@field@link}[3]{%
16722 \glsdoifexists{#2}%
16723 {%
16724 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
16725 \@gls@link[#1]{#2}{#3}%
16726 }%
16727 \gls@postlinkhook
16728 }
16729 \newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
16730 \newcommand*{\@glstext}[2][]{%
16731 \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
16732 \def\@glstext@#1#2[#3]{%
16733 \@gls@field@link{#1}{#2}{\glstentrytext{#2}#3}%
16734 }
16735 \newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
16736 \newcommand*{\@GLStext}[2][]{%
16737 \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
16738 \def\@GLStext@#1#2[#3]{%
16739 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrytext{#2}#3}%
16740 }
16741 \newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
16742 \newcommand*{\@Glstext}[2][]{%
16743 \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
16744 \def\@Glstext@#1#2[#3]{%
16745 \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
16746 }
16747 \newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
16748 \newcommand*{\@glsfirst}[2][]{%
16749 \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
16750 \def\@glsfirst@#1#2[#3]{%
16751 \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
16752 }
16753 \newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
16754 \newcommand*{\@Glsfirst}[2][]{%
16755 \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}}
16756 \def\@Glsfirst@#1#2[#3]{%
16757 \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
16758 }

```



```

16759 \newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
16760 \newcommand*{\@GLSfirst}[2] [] {%
16761   \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
16762 \def\@GLSfirst@#1#2[#3] {%
16763   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}{#3}}}%
16764 }
16765 \newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
16766 \newcommand*{\@glsplural}[2] [] {%
16767   \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
16768 \def\@glsplural@#1#2[#3] {%
16769   \@gls@field@link{#1}{#2}{\glsentryplural{#2}{#3}}%
16770 }
16771 \newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
16772 \newcommand*{\@Glsplural}[2] [] {%
16773   \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
16774 \def\@Glsplural@#1#2[#3] {%
16775   \@gls@field@link{#1}{#2}{\Glsentryplural{#2}{#3}}%
16776 }
16777 \newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
16778 \newcommand*{\@GLSplural}[2] [] {%
16779   \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
16780 \def\@GLSplural@#1#2[#3] {%
16781   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}{#3}}}%
16782 }
16783 \newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
16784 \newcommand*{\@glsfirstplural}[2] [] {%
16785   \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
16786 \def\@glsfirstplural@#1#2[#3] {%
16787   \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}{#3}}%
16788 }
16789 \newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
16790 \newcommand*{\@Glsfirstplural}[2] [] {%
16791   \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
16792 \def\@Glsfirstplural@#1#2[#3] {%
16793   \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}{#3}}%
16794 }
16795 \newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
16796 \newcommand*{\@GLSfirstplural}[2] [] {%
16797   \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
16798 \def\@GLSfirstplural@#1#2[#3] {%
16799   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}{#3}}}%
16800 }
16801 \newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
16802 \newcommand*{\@glsname}[2] [] {%
16803   \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
16804 \def\@glsname@#1#2[#3] {%
16805   \@gls@field@link{#1}{#2}{\glsentryname{#2}{#3}}%
16806 }
16807 \newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}

```

```

16808 \newcommand*{\@Glsname}[2] [] {%
16809   \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
16810 \def\@Glsname@#1#2[#3] {%
16811   \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
16812 }
16813 \newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
16814 \newcommand*{\@Glsname}[2] [] {%
16815   \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
16816 \def\@Glsname@#1#2[#3] {%
16817   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryname{#2}#3}}%
16818 }
16819 \newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
16820 \newcommand*{\@glsdesc}[2] [] {%
16821   \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
16822 \def\@glsdesc@#1#2[#3] {%
16823   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
16824 }
16825 \newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
16826 \newcommand*{\@Glsdesc}[2] [] {%
16827   \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
16828 \def\@Glsdesc@#1#2[#3] {%
16829   \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
16830 }
16831 \newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
16832 \newcommand*{\@Glsdesc}[2] [] {%
16833   \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
16834 \def\@Glsdesc@#1#2[#3] {%
16835   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydesc{#2}#3}}%
16836 }
16837 \newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
16838 \newcommand*{\@glsdescplural}[2] [] {%
16839   \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
16840 \def\@glsdescplural@#1#2[#3] {%
16841   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
16842 }
16843 \newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
16844 \newcommand*{\@Glsdescplural}[2] [] {%
16845   \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
16846 \def\@Glsdescplural@#1#2[#3] {%
16847   \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
16848 }
16849 \newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
16850 \newcommand*{\@Glsdescplural}[2] [] {%
16851   \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
16852 \def\@Glsdescplural@#1#2[#3] {%
16853   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydescplural{#2}#3}}%
16854 }
16855 \newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
16856 \newcommand*{\@glssymbol}[2] [] {%

```

```

16857 \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
16858 \def\@glssymbol@#1#2[#3]{%
16859 \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
16860 }
16861 \newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
16862 \newcommand*{\@Glsymbol}[2] [] {%
16863 \new@ifnextchar[{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
16864 \def\@Glsymbol@#1#2[#3]{%
16865 \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
16866 }
16867 \newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
16868 \newcommand*{\@GLSsymbol}[2] [] {%
16869 \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
16870 \def\@GLSsymbol@#1#2[#3]{%
16871 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrysymbol{#2}#3}}%
16872 }
16873 \newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
16874 \newcommand*{\@glssymbolplural}[2] [] {%
16875 \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}]
16876 \def\@glssymbolplural@#1#2[#3]{%
16877 \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
16878 }
16879 \newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
16880 \newcommand*{\@Glsymbolplural}[2] [] {%
16881 \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
16882 \def\@Glsymbolplural@#1#2[#3]{%
16883 \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
16884 }
16885 \newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
16886 \newcommand*{\@GLSsymbolplural}[2] [] {%
16887 \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
16888 \def\@GLSsymbolplural@#1#2[#3]{%
16889 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrysymbolplural{#2}#3}}%
16890 }
16891 \newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
16892 \newcommand*{\@glsuseri}[2] [] {%
16893 \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
16894 \def\@glsuseri@#1#2[#3]{%
16895 \@gls@field@link{#1}{#2}{\glstentryuseri{#2}#3}%
16896 }
16897 \newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
16898 \newcommand*{\@Glsuseri}[2] [] {%
16899 \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
16900 \def\@Glsuseri@#1#2[#3]{%
16901 \@gls@field@link{#1}{#2}{\glstentryuseri{#2}#3}%
16902 }
16903 \newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
16904 \newcommand*{\@GLSuseri}[2] [] {%
16905 \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]

```

```

16906 \def\@GLSuseri@#1#2[#3]{%
16907   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}%
16908 }
16909 \newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
16910 \newcommand*{\@glsuserii}[2][{}]{%
16911   \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
16912 \def\@glsuserii@#1#2[#3]{%
16913   \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
16914 }
16915 \newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
16916 \newcommand*{\@Glsuserii}[2][{}]{%
16917   \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
16918 \def\@Glsuserii@#1#2[#3]{%
16919   \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
16920 }
16921 \newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
16922 \newcommand*{\@GLSuserii}[2][{}]{%
16923   \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}}
16924 \def\@GLSuserii@#1#2[#3]{%
16925   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}%
16926 }
16927 \newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
16928 \newcommand*{\@glsuseriii}[2][{}]{%
16929   \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}}
16930 \def\@glsuseriii@#1#2[#3]{%
16931   \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
16932 }
16933 \newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
16934 \newcommand*{\@Glsuseriii}[2][{}]{%
16935   \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
16936 \def\@Glsuseriii@#1#2[#3]{%
16937   \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
16938 }
16939 \newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
16940 \newcommand*{\@GLSuseriii}[2][{}]{%
16941   \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}}
16942 \def\@GLSuseriii@#1#2[#3]{%
16943   \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriii{#2}#3}}%
16944 }
16945 \newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
16946 \newcommand*{\@glsuseriv}[2][{}]{%
16947   \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2}[]}}
16948 \def\@glsuseriv@#1#2[#3]{%
16949   \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
16950 }
16951 \newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
16952 \newcommand*{\@Glsuseriv}[2][{}]{%
16953   \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}}
16954 \def\@Glsuseriv@#1#2[#3]{%

```

```

16955 \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
16956 }
16957 \newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
16958 \newcommand*{\@GLSuseriv}[2] [] {%
16959 \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
16960 \def\@GLSuseriv@#1#2[#3]{%
16961 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuseriv{#2}#3}}%
16962 }
16963 \newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
16964 \newcommand*{\@glsuserv}[2] [] {%
16965 \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
16966 \def\@glsuserv@#1#2[#3]{%
16967 \@gls@field@link{#1}{#2}{\glentryuserv{#2}#3}%
16968 }
16969 \newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
16970 \newcommand*{\@GLSuserv}[2] [] {%
16971 \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
16972 \def\@GLSuserv@#1#2[#3]{%
16973 \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
16974 }
16975 \newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
16976 \newcommand*{\@GLSuservi}[2] [] {%
16977 \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
16978 \def\@GLSuservi@#1#2[#3]{%
16979 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuservi{#2}#3}}%
16980 }
16981 \newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
16982 \newcommand*{\@glsuservi}[2] [] {%
16983 \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
16984 \def\@glsuservi@#1#2[#3]{%
16985 \@gls@field@link{#1}{#2}{\glentryuservi{#2}#3}%
16986 }
16987 \newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
16988 \newcommand*{\@GLSuservi}[2] [] {%
16989 \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
16990 \def\@GLSuservi@#1#2[#3]{%
16991 \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
16992 }
16993 \newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
16994 \newcommand*{\@GLSuservi}[2] [] {%
16995 \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
16996 \def\@GLSuservi@#1#2[#3]{%
16997 \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glentryuservi{#2}#3}}%
16998 }
16999 \newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
17000 \newcommand*{\@ns@acrshort}[2] [] {%
17001 \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
17002 }
17003 \def\@acrshort#1#2[#3]{%

```

```

17004 \glsdoifexists{#2}%
17005 {%
17006   \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17007   \let\glsifplural\@secondoftwo
17008   \let\glscapscase\@firstofthree
17009   \let\glsinsert\@empty
17010   \def\glscustomtext{%
17011     \acronymfont{\glsentryshort{#2}}#3%
17012   }%
17013   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17014 }%
17015 \glspostlinkhook
17016 }
17017 \newrobustcmd*{\Acrshort}{\@gl@hyp@opt\ns@Acrshort}
17018 \newcommand*{\ns@Acrshort}[2][]{%
17019   \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2}[]}%
17020 }
17021 \def\@Acrshort#1#2[#3]{%
17022   \glsdoifexists{#2}%
17023   {%
17024     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17025     \def\glslabel{#2}%
17026     \let\glsifplural\@secondoftwo
17027     \let\glscapscase\@secondofthree
17028     \let\glsinsert\@empty
17029     \def\glscustomtext{%
17030       \acronymfont{\Glsentryshort{#2}}#3%
17031     }%
17032     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17033   }%
17034   \glspostlinkhook
17035 }
17036 \newrobustcmd*{\ACRshort}{\@gl@hyp@opt\ns@ACRshort}
17037 \newcommand*{\ns@ACRshort}[2][]{%
17038   \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2}[]}%
17039 }
17040 \def\@ACRshort#1#2[#3]{%
17041   \glsdoifexists{#2}%
17042   {%
17043     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17044     \def\glslabel{#2}%
17045     \let\glsifplural\@secondoftwo
17046     \let\glscapscase\@thirdofthree
17047     \let\glsinsert\@empty
17048     \def\glscustomtext{%
17049       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
17050     }%
17051     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17052   }%

```

```

17053 \glspostlinkhook
17054 }
17055 \newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
17056 \newcommand*{\ns@acrshortpl}[2] [] {%
17057   \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
17058 }
17059 \def\@acrshortpl#1#2[#3]{%
17060   \glsdoifexists{#2}%
17061   {%
17062     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17063     \def\glslabel{#2}%
17064     \let\glsifplural\@firstoftwo
17065     \let\glsapscase\@firstofthree
17066     \let\glsinsert\@empty
17067     \def\glscustomtext{%
17068       \acronymfont{\glsentryshortpl{#2}}#3%
17069     }%
17070     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17071   }%
17072   \glspostlinkhook
17073 }
17074 \newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
17075 \newcommand*{\ns@Acrshortpl}[2] [] {%
17076   \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
17077 }
17078 \def\@Acrshortpl#1#2[#3]{%
17079   \glsdoifexists{#2}%
17080   {%
17081     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17082     \def\glslabel{#2}%
17083     \let\glsifplural\@firstoftwo
17084     \let\glsapscase\@secondofthree
17085     \let\glsinsert\@empty
17086     \def\glscustomtext{%
17087       \acronymfont{\Glsentryshortpl{#2}}#3%
17088     }%
17089     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17090   }%
17091   \glspostlinkhook
17092 }
17093 \newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
17094 \newcommand*{\ns@ACRshortpl}[2] [] {%
17095   \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
17096 }
17097 \def\@ACRshortpl#1#2[#3]{%
17098   \glsdoifexists{#2}%
17099   {%
17100     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17101     \def\glslabel{#2}%

```

```

17102 \let\glsifplural\@firstoftwo
17103 \let\glschapscase\@thirdofthree
17104 \let\glsinsert\@empty
17105 \def\glscustomtext{%
17106     \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
17107 }%
17108 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17109 }%
17110 \glspostlinkhook
17111 }
17112 \newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}
17113 \newcommand*{\ns@acrlong}[2] [] {%
17114     \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
17115 }
17116 \def\@acrlong#1#2[#3] {%
17117     \glsdoifexists{#2}%
17118     {%
17119         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17120         \def\glslabel{#2}%
17121         \let\glsifplural\@secondoftwo
17122         \let\glschapscase\@firstofthree
17123         \let\glsinsert\@empty
17124         \def\glscustomtext{%
17125             \glsentrylong{#2}#3%
17126         }%
17127         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17128     }%
17129     \glspostlinkhook
17130 }
17131 \newrobustcmd*{\Acrlong}{\@gls@hyp@opt\@ns@Acrlong}
17132 \newcommand*{\ns@Acrlong}[2] [] {%
17133     \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2} []}%
17134 }
17135 \def\@Acrlong#1#2[#3] {%
17136     \glsdoifexists{#2}%
17137     {%
17138         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17139         \def\glslabel{#2}%
17140         \let\glsifplural\@secondoftwo
17141         \let\glschapscase\@secondofthree
17142         \let\glsinsert\@empty
17143         \def\glscustomtext{%
17144             \Glsentrylong{#2}#3%
17145         }%
17146         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17147     }%
17148     \glspostlinkhook
17149 }
17150 \newrobustcmd*{\ACRlong}{\@gls@hyp@opt\@ns@ACRlong}

```



```

17151 \newcommand*{\ns@ACRlong}[2][{}]{%
17152   \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}]%
17153 }
17154 \def\@ACRlong#1#2[#3]{%
17155   \glsdoifexists{#2}%
17156   {%
17157     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17158     \def\glslabel{#2}%
17159     \let\glsifplural\@secondoftwo
17160     \let\glscapscase\@thirdofthree
17161     \let\glsinsert\@empty
17162     \def\glscustomtext{%
17163       \mfirstucMakeUppercase{\glstrylong{#2}#3}%
17164     }%
17165     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17166   }%
17167   \glspostlinkhook
17168 }
17169 \newrobustcmd*{\acrlongpl}{\@gl@hyp@opt\ns@acrlongpl}
17170 \newcommand*{\ns@acrlongpl}[2][{}]{%
17171   \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}]%
17172 }
17173 \def\@acrlongpl#1#2[#3]{%
17174   \glsdoifexists{#2}%
17175   {%
17176     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17177     \def\glslabel{#2}%
17178     \let\glsifplural\@firstoftwo
17179     \let\glscapscase\@firstofthree
17180     \let\glsinsert\@empty
17181     \def\glscustomtext{%
17182       \glstrylongpl{#2}#3%
17183     }%
17184     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17185   }%
17186   \glspostlinkhook
17187 }
17188 \newrobustcmd*{\Acrlongpl}{\@gl@hyp@opt\ns@Acrlongpl}
17189 \newcommand*{\ns@Acrlongpl}[2][{}]{%
17190   \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}]%
17191 }
17192 \def\@Acrlongpl#1#2[#3]{%
17193   \glsdoifexists{#2}%
17194   {%
17195     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
17196     \def\glslabel{#2}%
17197     \let\glsifplural\@firstoftwo
17198     \let\glscapscase\@secondofthree
17199     \let\glsinsert\@empty

```

```

17200 \def\glscustomtext{%
17201 \Glsentrylongpl{#2}#3%
17202 }%
17203 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17204 }%
17205 \glspostlinkhook
17206 }
17207 \newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
17208 \newcommand*{\ns@ACRlongpl}[2][{}]{%
17209 \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
17210 }
17211 \def\@ACRlongpl#1#2[#3]{%
17212 \glsdoifexists{#2}%
17213 {%
17214 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
17215 \def\glslabel{#2}%
17216 \let\glsifplural\@firstoftwo
17217 \let\glscapscase\@thirdofthree
17218 \let\glsinsert\@empty
17219 \def\glscustomtext{%
17220 \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
17221 }%
17222 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
17223 }%
17224 \glspostlinkhook
17225 }
17226 \newcommand*{\@gls@entry@field}[2]{%
17227 \csname glo@\glsdetoklabel{#1}@#2\endcsname
17228 }
17229 \newcommand*{\glsletentryfield}[3]{%
17230 \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
17231 }
17232 \newcommand*{\@Gls@entry@field}[2]{%
17233 \glsdoifexistsordo{#1}%
17234 {%
17235 \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
17236 \ifdef\@glo@text
17237 {%
17238 \xmakefirstuc{\@glo@text}%
17239 }%
17240 {%
17241 ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
17242 entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
17243 label and the field name}%
17244 }%
17245 }%
17246 {%
17247 ???%
17248 }%

```

```

17249 }
17250 \newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
17251 \newrobustcmd*{\Glsentryname}[1]{%
17252   \@Gls@entryname{#1}%
17253 }
17254 \newcommand*{\@Gls@entryname}[1]{%
17255   \@Gls@entry@field{#1}{name}%
17256 }
17257 \newcommand*{\@Gls@acentryname}[1]{%
17258   \ifglshaslong{#1}%
17259   {%
17260     \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
17261     \expandafter\@gls@getbody\@glo@text{}\@nil
17262     \expandafter\ifx\@gls@body\glsentrylong\relax
17263       \expandafter\Glsentrylong\@gls@rest
17264     \else
17265       \expandafter\ifx\@gls@body\glsentryshort\relax
17266         \expandafter\Glsentryshort\@gls@rest
17267       \else
17268         \expandafter\ifx\@gls@body\acronymfont\relax
17269         {%
17270           \let\glsentryshort\Glsentryshort
17271           \@glo@text
17272         }%
17273       \else
17274         \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
17275         {%
17276           \let\glsentryshort\Glsentryshort
17277           \@glo@text
17278         }%
17279       \else
17280         \xmakefirstuc{\@glo@text}%
17281       \fi
17282     \fi
17283   \fi
17284   \fi
17285   }%
17286   {%
17287     \@Gls@entry@field{#1}{name}%
17288   }%
17289 }
17290 \newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
17291 \newrobustcmd*{\Glsentrydesc}[1]{%
17292   \@Gls@entry@field{#1}{desc}%
17293 }
17294 \newcommand*{\glsentrydescplural}[1]{%
17295   \@gls@entry@field{#1}{descplural}%
17296 }
17297 \newrobustcmd*{\Glsentrydescplural}[1]{%

```

```

17298 \@Gls@entry@field{#1}{descplural}%
17299 }
17300 \newcommand*{\glentrytext}[1]{\@Gls@entry@field{#1}{text}}
17301 \newrobustcmd*{\Glentrytext}[1]{%
17302 \@Gls@entry@field{#1}{text}%
17303 }
17304 \newcommand*{\glentryplural}[1]{%
17305 \@Gls@entry@field{#1}{plural}%
17306 }
17307 \newrobustcmd*{\Glentryplural}[1]{%
17308 \@Gls@entry@field{#1}{plural}%
17309 }
17310 \newcommand*{\glentrysymbol}[1]{%
17311 \@Gls@entry@field{#1}{symbol}%
17312 }
17313 \newrobustcmd*{\Glentrysymbol}[1]{%
17314 \@Gls@entry@field{#1}{symbol}%
17315 }
17316 \newcommand*{\glentrysymbolplural}[1]{%
17317 \@Gls@entry@field{#1}{symbolplural}%
17318 }
17319 \newrobustcmd*{\Glentrysymbolplural}[1]{%
17320 \@Gls@entry@field{#1}{symbolplural}%
17321 }
17322 \newcommand*{\glentryfirst}[1]{%
17323 \@Gls@entry@field{#1}{first}%
17324 }
17325 \newrobustcmd*{\Glentryfirst}[1]{%
17326 \@Gls@entry@field{#1}{first}%
17327 }
17328 \newcommand*{\glentryfirstplural}[1]{%
17329 \@Gls@entry@field{#1}{firstpl}%
17330 }
17331 \newrobustcmd*{\Glentryfirstplural}[1]{%
17332 \@Gls@entry@field{#1}{firstpl}%
17333 }
17334 \newrobustcmd*{\@glentrytitlecase}[2]{%
17335 \glstoifexists{#1}%
17336 {%
17337 \glstfieldfetch{#1}{#2}{\@Gls@value}%
17338 \xcapitalisewords{\@Gls@value}%
17339 }%
17340 }
17341 \ifdef\texorpdfstring
17342 {
17343 \newcommand*{\glentrytitlecase}[2]{%
17344 \texorpdfstring
17345 {\@glentrytitlecase{#1}{#2}}%
17346 {\@Gls@entry@field{#1}{#2}}%

```

```

17347 }
17348 }
17349 {
17350   \newcommand*{\glentrytitlecase}[2]{\@glentrytitlecase{#1}{#2}}
17351 }
17352 \newcommand*{\glentrytype}[1]{\@gls@entry@field{#1}{type}}
17353 \newcommand*{\glentrysort}[1]{%
17354   \@gls@entry@field{#1}{sort}%
17355 }
17356 \newcommand*{\glentryparent}[1]{%
17357   \@gls@entry@field{#1}{parent}%
17358 }
17359 \newcommand*{\glentryuseri}[1]{%
17360   \@gls@entry@field{#1}{useri}%
17361 }
17362 \newrobustcmd*{\Glsentryuseri}[1]{%
17363   \@Gls@entry@field{#1}{useri}%
17364 }
17365 \newcommand*{\glentryuserii}[1]{%
17366   \@gls@entry@field{#1}{userii}%
17367 }
17368 \newrobustcmd*{\Glsentryuserii}[1]{%
17369   \@Gls@entry@field{#1}{userii}%
17370 }
17371 \newcommand*{\glentryuseriii}[1]{%
17372   \@gls@entry@field{#1}{useriii}%
17373 }
17374 \newrobustcmd*{\Glsentryuseriii}[1]{%
17375   \@Gls@entry@field{#1}{useriii}%
17376 }
17377 \newcommand*{\glentryuseriv}[1]{%
17378   \@gls@entry@field{#1}{useriv}%
17379 }
17380 \newrobustcmd*{\Glsentryuseriv}[1]{%
17381   \@Gls@entry@field{#1}{useriv}%
17382 }
17383 \newcommand*{\glentryuserv}[1]{%
17384   \@gls@entry@field{#1}{userv}%
17385 }
17386 \newrobustcmd*{\Glsentryuserv}[1]{%
17387   \@Gls@entry@field{#1}{userv}%
17388 }
17389 \newcommand*{\glentryuservi}[1]{%
17390   \@gls@entry@field{#1}{uservi}%
17391 }
17392 \newrobustcmd*{\Glsentryuservi}[1]{%
17393   \@Gls@entry@field{#1}{uservi}%
17394 }
17395 \newcommand*{\glentryshort}[1]{\@gls@entry@field{#1}{short}}

```

```

17396 \newrobustcmd*{\Glsentryshort}[1]{%
17397   \@Gls@entry@field{#1}{short}%
17398 }
17399 \newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
17400 \newrobustcmd*{\Glsentryshortpl}[1]{%
17401   \@Gls@entry@field{#1}{shortpl}%
17402 }
17403 \newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
17404 \newrobustcmd*{\Glsentrylong}[1]{%
17405   \@Gls@entry@field{#1}{long}%
17406 }
17407 \newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
17408 \newrobustcmd*{\Glsentrylongpl}[1]{%
17409   \@Gls@entry@field{#1}{longpl}%
17410 }
17411 \newcommand*{\glsentryfull}[1]{%
17412   \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
17413 }
17414 \newrobustcmd*{\Glsentryfull}[1]{%
17415   \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
17416 }
17417 \newcommand*{\glsentryfullpl}[1]{%
17418   \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
17419 }
17420 \newrobustcmd*{\Glsentryfullpl}[1]{%
17421   \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
17422 }
17423 \newcommand*{\glsentrynumberlist}[1]{%
17424   \glsdoifexists{#1}%
17425   {%
17426     \@gls@entry@field{#1}{numberlist}%
17427   }%
17428 }
17429 \@ifpackageloaded{hyperref} {%
17430   \newcommand*{\glsdisplaynumberlist}[1]{%
17431     \GlossariesWarning
17432     {%
17433       \string\glsdisplaynumberlist\space
17434       doesn't work with hyperref.^^JUsing
17435       \string\glsentrynumberlist\space instead%
17436     }%
17437     \glsentrynumberlist{#1}%
17438   }%
17439 }%
17440 {%
17441   \newcommand*{\glsdisplaynumberlist}[1]{%
17442     \glsdoifexists{#1}%
17443     {%
17444       \bgroup

```



```

17494 }
17495 \newcommand*{\@gls@adjustmode}{%
17496 \AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
17497 \newrobustcmd*{\glsaddall}[1][1]{%
17498   \edef\@glo@type{\@glo@types}%
17499   \setkeys{glossadd}{#1}%
17500   \forallglsentries[\@glo@type]{\@glo@entry}{%
17501     \glsadd[#1]{\@glo@entry}%
17502   }%
17503 }
17504 \newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
17505   \forallglsentries[#1]{\@glo@entry}%
17506 }%
17507   \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
17508 }%
17509 }
17510 \newcommand*{\glsignore}[1]{%
17511   \edef\glsopenbrace{\expandafter\@gobble\string\{ }
17512   \edef\glsclosebrace{\expandafter\@gobble\string\} }
17513   \edef\glsbackslash{\expandafter\@gobble\string\ }
17514   \edef\glsquote#1{\string"#1\string"}
17515   \edef\glspercentchar{\expandafter\@gobble\string\% }
17516   \edef\glstildechar{\string~ }
17517   \ifglxsindy
17518     \newcommand*{\@glsfirstletter}{A}
17519   \fi
17520 \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
17521   \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
17522 \ifglxsindy
17523   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
17524     \renewcommand*{\@glsfirstletter}{#1}}
17525   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
17526     \renewcommand*{\@glsfirstletter}{#1}%
17527     \@onelevel@sanitize\@glsfirstletter
17528   }
17529 \else
17530   \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
17531     \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
17532   \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
17533     \@GlsSetXdyFirstLetterAfterDigits
17534   }
17535 \fi
17536 \ifglxsindy
17537   \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
17538 \fi
17539 \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
17540   \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
17541 }
17542 \ifglxsindy

```



```

17543 \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
17544   \renewcommand*{\@xdynumbergrouporder}{#1}%
17545 }
17546 \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
17547   \renewcommand*{\@xdynumbergrouporder}{#1}%
17548   \@onelevel@sanitize\@xdynumbergrouporder
17549 }
17550 \else
17551   \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
17552     \glsnoindywarning\GlsSetXdyNumberGroupOrder}
17553   \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
17554     \@GlsSetXdyNumberGroupOrder}
17555 \fi
17556 \newcommand*{\@glsminrange}{2}
17557 \ifglxindy
17558   \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
17559     \renewcommand*{\@glsminrange}{#1}}
17560 \else
17561   \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
17562     \glsnoindywarning\GlsSetXdyMinRangeLength}
17563 \fi
17564
17565 \ifglxindy
17566   \def\writeist{%
17567     \ifundef{\glswrite}{\newwrite\glswrite}{}%
17568     \@gls@addpredefinedattributes
17569     \openout\glswrite=\istfilename
17570     \write\glswrite{;; xindy style file created by the glossaries
17571       package}%
17572     \write\glswrite{;; for document '\jobname' on
17573       \the\year-\the\month-\the\day}%
17574     \write\glswrite{^^J; required styles^^J}
17575     \@for\@xdystyle:=\@xdyrequiredstyles\do{%
17576       \ifx\@xdystyle\@empty
17577       \else
17578         \protected@write\glswrite{}{(require
17579           \string"\@xdystyle.xdy\string")}%
17580       \fi
17581     }%
17582     \write\glswrite{^^J%
17583       ; list of allowed attributes (number formats)^^J}%
17584     \write\glswrite{(define-attributes ((\@xdyattributes)))}%
17585     \write\glswrite{^^J; user defined alphabets^^J}%
17586     \write\glswrite{\@xdyuseralphabets}%
17587     \write\glswrite{^^J; location class definitions^^J}%
17588     \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
17589       \protected@write\glswrite{}{(define-location-class
17590         \string"\@gls@classI\string"^^J\space\space\space
17591         (

```

```

17592         :sep "{}{"
17593         \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
17594         :sep "}"
17595     )
17596     ^^J\space\space\space
17597     :min-range-length \@glsminrange^^J%
17598 )
17599 }%
17600 {%
17601     \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
17602         \protected@write\glswrite{}\{(define-location-class
17603             \string"\@gls@classII-\@gls@classI\string"
17604             ^^J\space\space\space
17605             (
17606                 :sep "{"
17607                 \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
17608                 :sep "{}{"
17609                 \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
17610                 :sep "}"
17611             )
17612             ^^J\space\space\space
17613             :min-range-length \@glsminrange^^J%
17614         )
17615     }%
17616 }%
17617 }%
17618 }%
17619 \write\glswrite{^^J; user defined location classes}%
17620 \write\glswrite{\@xdyuserlocationdefs}%
17621 \write\glswrite{^^J; define cross-reference class^^J}%
17622 \write\glswrite{(define-crossref-class \string"see\string"
17623     :unverified )}%
17624 \write\glswrite{(markup-crossref-list
17625     :class \string"see\string"^^J\space\space\space
17626     :open \string"\string\glsseeformat\string"
17627     :close \string"{}\string")}%
17628 \@xdycrossrefhook
17629 \write\glswrite{^^J; define the order of the location classes}%
17630 \write\glswrite{(define-location-class-order
17631     (\@xdylocationclassorder))}%
17632 \write\glswrite{^^J; define the glossary markup^^J}%
17633 \write\glswrite{(markup-index^^J\space\space\space
17634     :open \string"\string
17635     \glossarysection[\string\glossarytoctitle]{\string
17636     \glossarytitle}\string\glossarypreamble}%
17637 \@for\@this@ctr:=\@xdycounters\do{%
17638     {%
17639         \@for\@this@attr:=\@xdyattributelist\do{%
17640             \protected@write\glswrite{}\{\string\providecommand*%

```

```

17641         \expandafter\string
17642         \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
17643         {%
17644             \string\setentrycounter
17645             [\expandafter\@gobble\string\#1]{\@this@ctr}%
17646             \expandafter\string
17647             \csname\@this@attr\endcsname
17648             {\expandafter\@gobble\string\#2}%
17649         }%
17650     }%
17651 }%
17652 }%
17653 }%
17654 \write\glswrite{%
17655     \string\begin
17656     {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
17657     \space\space:close \string"\glspercentchar\glstildechar n\string
17658     \end{theglossary}\string\glossarypostamble
17659     \glstildechar n\string" ^^J\space\space\space
17660     :tree)}}%
17661 \write\glswrite{(markup-letter-group-list
17662     :sep \string"\string\glsgroupskip\glstildechar n\string"))%
17663 \write\glswrite{(markup-indexentry
17664     :open \string"\string\relax \string\glsresetentrylist
17665     \glstildechar n\string"))%
17666 \write\glswrite{(markup-locclass-list :open
17667     \string"\glsopenbrace\string\glossaryentrynumbers
17668     \glsopenbrace\string\relax\space \string"^^J\space\space\space
17669     :sep \string", \string"
17670     :close \string"\glsclosebrace\glsclosebrace\string"))%
17671 \write\glswrite{(markup-locref-list
17672     :sep \string"\string\delimN\space\string"))%
17673 \write\glswrite{(markup-range
17674     :sep \string"\string\delimR\space\string"))%
17675 \@onelevel@sanitize\gls@suffixF
17676 \@onelevel@sanitize\gls@suffixFF
17677 \ifx\gls@suffixF\@empty
17678 \else
17679     \write\glswrite{(markup-range
17680         :close "\gls@suffixF" :length 1 :ignore-end)}}%
17681 \fi
17682 \ifx\gls@suffixFF\@empty
17683 \else
17684     \write\glswrite{(markup-range
17685         :close "\gls@suffixFF" :length 2 :ignore-end)}}%
17686 \fi
17687 \write\glswrite{^^J; define format to use for locations^^J}%
17688 \write\glswrite{\@xdylocref}%
17689 \write\glswrite{^^J; define letter group list format^^J}%

```

```

17690 \write\glswrite{(markup-letter-group-list
17691 :sep \string"\string\glsgroupskip\glstildechar n\string")}%
17692 \write\glswrite{^^J; letter group headings^^J}%
17693 \write\glswrite{(markup-letter-group
17694 :open-head \string"\string\glsgroupheading
17695 \glsoopenbrace\string"^^J\space\space\space
17696 :close-head \string"\glsclosebrace\string")}%
17697 \write\glswrite{^^J; additional letter groups^^J}%
17698 \write\glswrite{\@xdylettergroups}%
17699 \write\glswrite{^^J; additional sort rules^^J}
17700 \write\glswrite{\@xdysortrules}%
17701 \@gls@writeisthook
17702 \closeout\glswrite
17703 \let\writeist\relax
17704 }
17705 \else
17706 \edef\@gls@actualchar{\string?}
17707 \edef\@gls@encapchar{\string|}
17708 \edef\@gls@levelchar{\string!}
17709 \edef\@gls@quotechar{\string"}%
17710 \let\GlsSetQuote\gls@nosetquote
17711 \def\writeist{\relax
17712 \ifundef{\glswrite}{\newwrite\glswrite}{\relax
17713 \openout\glswrite=\istfilename
17714 \write\glswrite{\glspersentchar\space makeindex style file
17715 created by the glossaries package}
17716 \write\glswrite{\glspersentchar\space for document
17717 '\jobname' on \the\year-\the\month-\the\day}
17718 \write\glswrite{actual '\@gls@actualchar'}
17719 \write\glswrite{encap '\@gls@encapchar'}
17720 \write\glswrite{level '\@gls@levelchar'}
17721 \write\glswrite{quote '\@gls@quotechar'}
17722 \write\glswrite{keyword \string"\string\glossaryentry\string"}
17723 \write\glswrite{preamble \string"\string\glossarysection[\string
17724 \glossarytoctitle]{\string\glossarytitle}\string
17725 \glossarypreamble\string\n\string\begin{theglossary}\string
17726 \glossaryheader\string\n\string"}
17727 \write\glswrite{postamble \string"\string%\string\n\string
17728 \end{theglossary}\string\glossarypostamble\string\n
17729 \string"}
17730 \write\glswrite{group_skip \string"\string\glsgroupskip\string\n
17731 \string"}
17732 \write\glswrite{item_0 \string"\string%\string\n\string"}
17733 \write\glswrite{item_1 \string"\string%\string\n\string"}
17734 \write\glswrite{item_2 \string"\string%\string\n\string"}
17735 \write\glswrite{item_01 \string"\string%\string\n\string"}
17736 \write\glswrite{item_x1
17737 \string"\string\relax \string\glsresetentrylist\string\n
17738 \string"}

```

```

17739 \write\glswrite{item_12 \string"\string%\string\n\string"}
17740 \write\glswrite{item_x2
17741 \string"\string\relax \string\glresetentrylist\string\n
17742 \string"}
17743 \write\glswrite{delim_0 \string"\string\{\string
17744 \glossaryentrynumbers\string\{\string\relax \string"}
17745 \write\glswrite{delim_1 \string"\string\{\string
17746 \glossaryentrynumbers\string\{\string\relax \string"}
17747 \write\glswrite{delim_2 \string"\string\{\string
17748 \glossaryentrynumbers\string\{\string\relax \string"}
17749 \write\glswrite{delim_t \string"\string\}\string\}\string"}
17750 \write\glswrite{delim_n \string"\string\delimN \string"}
17751 \write\glswrite{delim_r \string"\string\delimR \string"}
17752 \write\glswrite{headings_flag 1}
17753 \write\glswrite{heading_prefix
17754 \string"\string\glsgroupheading\string\{\string"}
17755 \write\glswrite{heading_suffix
17756 \string"\string\}\string\relax
17757 \string\glresetentrylist \string"}
17758 \write\glswrite{symhead_positive \string"glssymbols\string"}
17759 \write\glswrite{numhead_positive \string"glnumbers\string"}
17760 \write\glswrite{page_compositor \string"glscpositor\string"}
17761 \@gls@escbsdq\gls@suffixF
17762 \@gls@escbsdq\gls@suffixFF
17763 \ifx\gls@suffixF\@empty
17764 \else
17765 \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
17766 \fi
17767 \ifx\gls@suffixFF\@empty
17768 \else
17769 \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
17770 \fi
17771 \@gls@writeisthook
17772 \closeout\glswrite
17773 \let\writeist\relax
17774 }
17775 \fi
17776 \newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
17777 \@onlypremakeg\GlsSetWriteIstHook
17778 \newcommand*\@gls@writeisthook}{
17779 \ifglxindy
17780 \newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
17781 \newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
17782 \else
17783 \newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
17784 \@ifpackageloaded{tracklang}%
17785 {%
17786 \IfTrackedLanguage{german}%
17787 {%

```

```

17788     \def\@gls@extramakeindexopts{-g}%
17789 }%
17790 {}%
17791 }%
17792 {}%
17793 \edef\@gls@docheckquotedef{%
17794     \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
17795         \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
17796         \noexpand\toks@={####1}%
17797         \noexpand\ifx\noexpand\null####2\noexpand\null
17798         \noexpand\ifx\noexpand\null####3\noexpand\null
17799         \noexpand\edef\noexpand\@gls@checkedmkidx{%
17800             \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
17801         \noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
17802     \noexpand\else
17803         \noexpand\edef\noexpand\@gls@checkedmkidx{%
17804             \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17805             \noexpand\@gls@quotechar\noexpand\@gls@quotechar
17806             \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
17807         \noexpand\def\noexpand\@gls@checkquote{%
17808             \noexpand\@gls@checkquote####3\noexpand\null}%
17809     \noexpand\fi
17810 \noexpand\else
17811     \noexpand\edef\noexpand\@gls@checkedmkidx{%
17812         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17813         \noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
17814     \noexpand\ifx\noexpand\null####3\noexpand\null
17815     \noexpand\def\noexpand\@gls@checkquote{%
17816         \noexpand\@gls@checkquote####2#1#1\noexpand\null}%
17817     \noexpand\else
17818     \noexpand\def\noexpand\@gls@checkquote{%
17819         \noexpand\@gls@checkquote####2#1####3\noexpand\null}%
17820     \noexpand\fi
17821     \noexpand\fi
17822     \noexpand\@gls@checkquote
17823 }%
17824 }%
17825 \@gls@docheckquotedef
17826 \edef\@gls@docheckquotedef{%
17827     \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
17828         \noexpand\def\noexpand\@gls@checkedmkidx{%
17829             \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
17830             #1#1\noexpand\null
17831             \noexpand\expandafter\noexpand\@gls@updatechecked
17832             \noexpand\@gls@checkedmkidx{####1}%
17833             \noexpand\def\noexpand\@gls@checkedmkidx{%
17834                 \noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
17835                 \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
17836                 \noexpand\null

```

```

17837 \noexpand\expandafter\noexpand\@gls@updatechecked
17838 \noexpand\@gls@checkedmkidx{####1}%
17839 \noexpand\def\noexpand\@gls@checkedmkidx{%
17840 \noexpand\expandafter\noexpand\@gls@checkscactual####1\noexpand\@nil
17841 \noexpand\?\noexpand\?\noexpand\null
17842 \noexpand\expandafter\noexpand\@gls@updatechecked
17843 \noexpand\@gls@checkedmkidx{####1}%
17844 \noexpand\def\noexpand\@gls@checkedmkidx{%
17845 \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
17846 \noexpand?\noexpand?\noexpand\null
17847 \noexpand\expandafter\noexpand\@gls@updatechecked
17848 \noexpand\@gls@checkedmkidx{####1}%
17849 \noexpand\def\noexpand\@gls@checkedmkidx{%
17850 \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
17851 \noexpand|\noexpand|\noexpand\null
17852 \noexpand\expandafter\noexpand\@gls@updatechecked
17853 \noexpand\@gls@checkedmkidx{####1}%
17854 \noexpand\def\noexpand\@gls@checkedmkidx{%
17855 \noexpand\expandafter\noexpand\@gls@checkscbar####1\noexpand\@nil
17856 \noexpand||\noexpand||\noexpand\null
17857 \noexpand\expandafter\noexpand\@gls@updatechecked
17858 \noexpand\@gls@checkedmkidx{####1}%
17859 \noexpand\def\noexpand\@gls@checkedmkidx{%
17860 \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
17861 \noexpand!\noexpand!\noexpand\null
17862 \noexpand\expandafter\noexpand\@gls@updatechecked
17863 \noexpand\@gls@checkedmkidx{####1}%
17864 }%
17865 }%
17866 \@gls@docheckquotedef
17867 \edef\@gls@docheckquotedef{%
17868 \noexpand\def\noexpand\@gls@checkescquote####1%
17869 \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
17870 ####3\noexpand\null{%
17871 \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
17872 \noexpand\toks@={####1}%
17873 \noexpand\ifx\noexpand\null####2\noexpand\null
17874 \noexpand\ifx\noexpand\null####3\noexpand\null
17875 \noexpand\edef\noexpand\@gls@checkedmkidx{%
17876 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
17877 \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
17878 \noexpand\else
17879 \noexpand\edef\noexpand\@gls@checkedmkidx{%
17880 \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17881 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
17882 \csname#1\endcsname}\noexpand\@gls@quotechar
17883 \noexpand\@gls@quotechar\noexpand\string\expandonce{%
17884 \csname#1\endcsname}\noexpand\@gls@quotechar}%
17885 \noexpand\def\noexpand\@gls@checkescquote{%

```

```

17886         \noexpand\@gls@checkescquote####3\noexpand\null}%
17887     \noexpand\fi
17888     \noexpand\else
17889     \noexpand\edef\noexpand\@gls@checkedmkidx{%
17890         \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
17891         \noexpand\@gls@quotearch\noexpand\string
17892         \expandonce{\csname#1\endcsname}\noexpand\@gls@quotearch}%
17893     \noexpand\ifx\noexpand\null####3\noexpand\null
17894     \noexpand\def\noexpand\@gls@checkescquote{%
17895         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
17896         \expandonce{\csname#1\endcsname}\noexpand\null}%
17897     \noexpand\else
17898     \noexpand\def\noexpand\@gls@checkescquote{%
17899         \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
17900         ####3\noexpand\null}%
17901     \noexpand\fi
17902     \noexpand\fi
17903     \noexpand\@gls@checkescquote
17904 }%
17905 }%
17906 \@gls@docheckquotedef
17907 }
17908 \newcommand*{\@gls@nosetquote}[1]{\PackageError{glossaries}%
17909     {\string\GlsSetQuote\space not permitted here}%
17910     {\Move \string\GlsSetQuote\space earlier in the preamble, as
17911     soon as possible after glossaries.sty has been loaded}}
17912 \fi
17913 \newcommand*{\@gls@extramakeindexopts}[1]{%
17914 \newcommand{\noist}{%
17915     \@gls@addpredefinedattributes
17916     \let\writeist\relax
17917 }
17918 \newcommand*{\@makeglossary}[1]{%
17919     \ifglossaryexists{#1}%
17920     {%
17921         \ifglssavewrites
17922         \expandafter\newtoks\csname glo@#1@filetok\endcsname
17923         \else
17924         \expandafter\newwrite\csname glo@#1@file\endcsname
17925         \expandafter\@gls@openfile\csname glo@#1@file\endcsname{#1}%
17926         \fi
17927         \@gls@renewglossary
17928         \writeist
17929     }%
17930     {%
17931         \PackageError{glossaries}%
17932         {Glossary type ‘#1’ not defined}%
17933         {New glossaries must be defined before using \string\makeglossaries}%
17934     }%

```



```

17935 }
17936 \newcommand*{\@glsopenfile}[2]{%
17937   \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
17938   \PackageInfo{glossaries}{Writing glossary file
17939     \jobname.\csname @glotype@#2@out\endcsname}%
17940 }
17941 \newcommand*{\@closegls}[1]{%
17942   \closeout\csname glo@#1@file\endcsname
17943 }
17944 \ifglxindy
17945   \newcommand*{\@gls@automake}[1]{%
17946     \ifglossaryexists{#1}
17947     {%
17948       \@closegls{#1}%
17949       \ifdefstring{\glsorder}{letter}%
17950       {\def\@gls@order{-M ord/letorder }}%
17951       {\let\@gls@order\@empty}%
17952       \ifcsundef{\xdy@#1@language}%
17953       {\let\@gls@langmod\@xdy@main@language}%
17954       {\letcs\@gls@langmod{\xdy@#1@language}}%
17955       \edef\@gls@dothiswrite{\noexpand\write18{xindy
17956         -I xindy
17957         \@gls@order
17958         -L \@gls@langmod\space
17959         -M \@gls@istfilebase\space
17960         -C \@gls@codepage\space
17961         -t \jobname.\csuse{@glotype@#1@log}
17962         -o \jobname.\csuse{@glotype@#1@in}
17963         \jobname.\csuse{@glotype@#1@out}}}%
17964       }%
17965       \@gls@dothiswrite
17966     }%
17967     {%
17968       \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17969     }%
17970   }
17971 \else
17972   \newcommand*{\@gls@automake}[1]{%
17973     \ifglossaryexists{#1}
17974     {%
17975       \@closegls{#1}%
17976       \ifdefstring{\glsorder}{letter}%
17977       {\def\@gls@order{-l }}%
17978       {\let\@gls@order\@empty}%
17979       \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
17980         -s \istfilename\space
17981         -t \jobname.\csuse{@glotype@#1@log}
17982         -o \jobname.\csuse{@glotype@#1@in}
17983         \jobname.\csuse{@glotype@#1@out}}}%

```

```

17984     }%
17985     \@gls@dothiswrite
17986 }%
17987 {%
17988     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
17989 }%
17990 }
17991 \fi
17992 \ifglsxindy
17993 \newcommand*{\@gls@automake@immediate}[1]{%
17994     \ifglossaryexists{#1}
17995     {%
17996         \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
17997             {%
17998                 \ifdefstring{\glsorder}{letter}%
17999                 {\def\@gls@order{-M ord/letorder }}%
18000                 {\let\@gls@order\empty}%
18001                 \ifcsundef{xdy@#1@language}%
18002                 {\let\@gls@langmod\@xdy@main@language}%
18003                 {\letcs\@gls@langmod{\@xdy@#1@language}}%
18004                 \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
18005                     -I xindy
18006                     \@gls@order
18007                     -L \@gls@langmod\space
18008                     -M \gls@istfilebase\space
18009                     -C \gls@codepage\space
18010                     -t \jobname.\csuse{@glotype@#1@log}
18011                     -o \jobname.\csuse{@glotype@#1@in}
18012                     \jobname.\csuse{@glotype@#1@out}}}%
18013             }%
18014             \@gls@dothiswrite
18015         }%
18016         {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
18017             doesn't exist. Rerun may be required}}%
18018     }%
18019 }%
18020     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
18021 }%
18022 }
18023 \else
18024 \newcommand*{\@gls@automake@immediate}[1]{%
18025     \ifglossaryexists{#1}
18026     {%
18027         \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
18028             {%
18029                 \ifdefstring{\glsorder}{letter}%
18030                 {\def\@gls@order{-l }}%
18031                 {\let\@gls@order\empty}%
18032                 \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order

```

```

18033         -s \istfilename\space
18034         -t \jobname.\csuse{@glotype@#1@log}
18035         -o \jobname.\csuse{@glotype@#1@in}
18036         \jobname.\csuse{@glotype@#1@out}}}%
18037     }%
18038     \@gls@dothiswrite
18039 }%
18040 {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
18041     doesn't exist. Rerun may be required}}}%
18042 }%
18043 {%
18044     \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
18045 }%
18046 }
18047 \fi
18048 \newcommand*{\@warn@nomakeglossaries}{}
18049 \newcommand*{\@warn@nomakeglossaries}{\@warn@nomakeglossaries}
18050 \newcommand{\@gls@@automake@immediate}{%
18051     \ifnum\gls@automake@nr=2\relax
18052         \@for\@gls@type:=\@glo@types\do{%
18053             \ifdefempty{\@gls@type}{}%
18054             {\@gls@automake@immediate{\@gls@type}}%
18055         }%
18056         \glsautomakefalse
18057         \renewcommand*{\@gls@doautomake}{}%
18058     \fi
18059 }
18060 \newcommand*{\makeglossaries}{%
18061     \@domakeglossaries
18062 {%
18063     \@gls@@automake@immediate
18064     \ifundef{\glswrite}{\newwrite\glswrite}{}%
18065     \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
18066     \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
18067     \ifundef\@gls@extramakeindexopts
18068     {}%
18069     {%
18070         \protected@write\@auxout{}{\string\providecommand
18071             \string\@gls@extramakeindexopts[1]{} }
18072         \protected@write\@auxout{}{\string\@gls@extramakeindexopts
18073             {\@gls@extramakeindexopts}}%
18074     }%
18075     \protected@write\@auxout{}{\string\@istfilename{\istfilename}}}%
18076     \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
18077     \@for\@glo@type:=\@glo@types\do{%
18078         \ifthenelse{\equal{\@glo@type}{} }{}{%
18079             \@makeglossary{\@glo@type}}%
18080     }%
18081     \renewcommand*{\newglossary[4] []}{%

```

```

18082 \PackageError{glossaries}{New glossaries
18083 must be created before \string\makeglossaries}{You need
18084 to move \string\makeglossaries\space after all your
18085 \string\newglossary\space commands}}%
18086 \let\@makeglossary\@gobble
18087 \let\makeglossaries\relax
18088 \@disable@onlypremakeg
18089 \let\gls@checkseeallowed\relax
18090 \let\warn@nomakeglossaries\relax
18091 \def\warn@noprintglossary{%
18092   \ifdefstring{\@glo@types}{,}%
18093   {%
18094     \GlossariesWarningNoLine{No glossaries have been defined}%
18095   }%
18096   {%
18097     \GlossariesWarningNoLine{No \string\printglossary\space
18098       or \string\printglossaries\space
18099       found. ^^J(Remove \string\makeglossaries\space if you
18100       don't want any glossaries.) ^^JThis document will not
18101       have a glossary}%
18102   }%
18103 }%
18104 \ifglssavenumberlist
18105   \edef\@gls@dodolistparser{\noexpand\DeclareListParser
18106     {\noexpand\glsnumlistparser}{\delimN}}%
18107   \@gls@dodolistparser
18108 \fi
18109 \let\makenoidxglossaries\@no@makeglossaries
18110 \renewcommand*{\@printgloss@setsort}{%
18111   \let\@glo@assign@sortkey\@glo@no@assign@sortkey
18112 }%
18113 \ifglssautomake
18114   \renewcommand*{\@gls@doautomake}{%
18115     \@for\@gls@type:=\@glo@types\do{%
18116       \ifdefempty{\@gls@type}{}%
18117       {\@gls@automake{\@gls@type}}%
18118     }%
18119   }%
18120 \fi
18121 \@glo@check@sortallowed\makeglossaries
18122 }%
18123 }
18124 \@onlypreamble{\makeglossaries}
18125 \AtEndDocument{%
18126   \warn@nomakeglossaries
18127   \warn@noprintglossary
18128 }
18129 \newcommand*{\makenoidxglossaries}{%
18130   \@domakeglossaries

```

```

18131 {%
18132 \renewcommand{\@gls@noref@warn}[1]{%
18133 \GlossariesWarning{Empty glossary for
18134 \string\printnoidxglossary[type={##1}].
18135 Rerun may be required (or you may have forgotten to use
18136 commands like \string\gls)}}%
18137 }%
18138 \let\@gls@checkmkidxchars\@gobble
18139 \glsclocationsfalse
18140 \let\@do@@wrglossary\gls@noidxglossary
18141 \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
18142 \let\gls@checkseeallowed\relax
18143 \renewcommand{\@do@seeglossary}[2]{%
18144 \edef\@gls@label{\glsdetoklabel{##1}}%
18145 \protected@write\@auxout{}{%
18146 \string\@gls@reference
18147 {\csname glo@\@gls@label @type\endcsname}%
18148 {\@gls@label}%
18149 {%
18150 \string\glsseeformat##2}%
18151 }%
18152 }%
18153 }%
18154 \AtBeginDocument
18155 {%
18156 \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
18157 }%
18158 \def\warn@noprintglossary{%
18159 \GlossariesWarningNoLine{No \string\printnoidxglossary\space
18160 or \string\printnoidxglossaries ^^J
18161 found. (Remove \string\makenoidxglossaries\space if you
18162 don't want any glossaries.)^^JThis document will not have a glossary}%
18163 }%
18164 \let\warn@nomakeglossaries\relax
18165 \let\makeglossaries\@no@makeglossaries
18166 \renewcommand*\@printgloss@setsort{%
18167 \let\@glo@assign@sortkey\@glo@assign@sortkey
18168 \def\@glo@sorttype{\@glo@default@sorttype}%
18169 }%
18170 \renewcommand*\new@glossaryentry[2]{%
18171 \PackageError{glossaries}{Glossary entries must be
18172 defined in the preamble^^Jwhen you use
18173 \string\makenoidxglossaries}%
18174 {Either move your definitions to the preamble or use
18175 \string\makeglossaries}%
18176 }%
18177 \renewcommand*\@glsentrynumberlist}[1]{%
18178 \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18179 \ifdef\@gls@loclist

```

```

18180 {%
18181     \glsnoidxloclist{\@gls@loclist}%
18182 }%
18183 {%
18184     ??\glsdoifexists{##1}%
18185     {%
18186         \GlossariesWarning{Missing location list for ‘##1’. Either
18187             a rerun is required or you haven’t referenced the entry}%
18188     }%
18189 }%
18190 }%
18191 \renewcommand*{\glsdisplaynumberlist}[1]{%
18192     \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18193     \ifdef\@gls@loclist
18194     {%
18195         \def\@gls@noidxloclist@sep{%
18196             \def\@gls@noidxloclist@sep{%
18197                 \def\@gls@noidxloclist@sep{%
18198                     \glsnumlistsep
18199                 }%
18200                 \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
18201             }%
18202         }%
18203         \def\@gls@noidxloclist@finalsep{}%
18204         \def\@gls@noidxloclist@prev{}%
18205         \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
18206             \@gls@noidxloclist@finalsep
18207             \@gls@noidxloclist@prev
18208     }%
18209     {%
18210         ??\glsdoifexists{##1}%
18211         {%
18212             \GlossariesWarning{Missing location list for ‘##1’. Either
18213                 a rerun is required or you haven’t referenced the entry}%
18214         }%
18215     }%
18216 }%
18217 \renewcommand*{\glsnumberlistloop}[3]{%
18218     \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
18219     \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
18220     \let\@gls@org@glsseeformat\glsseeformat
18221     \let\glsnoidxdisplayloc##2\relax
18222     \let\glsseeformat##3\relax
18223     \ifdef\@gls@loclist
18224     {%
18225         \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
18226     }%
18227     {%
18228         ??\glsdoifexists{##1}%

```

```

18229      {%
18230      \GlossariesWarning{Missing location list for ‘##1’. Either
18231      a rerun is required or you haven’t referenced the entry}%
18232      }%
18233      }%
18234      \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
18235      \let\glsseeformat\@gls@org@glsseeformat
18236      }%
18237      \let\@gls@sanitizesort\@gls@noidx@sanitizesort
18238      \let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
18239      \@gls@noidx@setsanitizesort
18240      \@glo@check@sortallowed\makenoidxglossaries
18241      }%
18242  }
18243  \onlypreamble{\makenoidxglossaries}
18244  \newcommand*{\glsnumberlistloop}[2]{%
18245      \PackageError{glossaries}{\string\glsnumberlistloop\space
18246      only works with \string\makenoidxglossaries}{}%
18247  }
18248  \newcommand*{\glsnoidxnumberlistloop}[1]{%
18249      #1%
18250  }
18251  \newcommand*{\@no@makeglossaries}{%
18252      \PackageError{glossaries}{You can’t use both
18253      \string\makeglossaries\space and \string\makenoidxglossaries}%
18254      {Either use one or other (or none) of those commands but not both
18255      together.}%
18256  }
18257  \newcommand{\@gls@noref@warn}[1]{%
18258      \GlossariesWarning{\string\makenoidxglossaries\space
18259      is required to make \string\printnoidxglossary[type={#1}] work}%
18260  }
18261  \newcommand*{\gls@noidxglossary}{%
18262      \protected@write\@auxout{}{%
18263          \string\@gls@reference
18264          {\csname glo@\@gls@label @type\endcsname}%
18265          {\@gls@label}%
18266          {\string\glsnoidxdisplayloc
18267          {\@glo@counterprefix}%
18268          {\@gls@counter}%
18269          {\@glsnumberformat}%
18270          {\@glslocref}%
18271          }%
18272      }%
18273  }
18274  \providecommand\istfile{\glswrite}
18275  \AtEndDocument{%
18276      \glswritefiles
18277  }

```

```

18278 \newcommand*{\@glswritefiles}{%
18279   \forallglossaries{\@glo@type}{%
18280     \ifcsundef{glo@\@glo@type @filetok}%
18281     {%
18282       \def\gls@tmp{}%
18283     }%
18284     {%
18285       \edef\gls@tmp{\expandafter\the
18286         \csname glo@\@glo@type @filetok\endcsname}%
18287     }%
18288     \ifx\gls@tmp\@empty
18289       \ifx\@glo@type\glsdefaulttype
18290         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
18291           entries.^^JRemember to use package option 'nomain' if
18292 you
18293           don't want to^^Juse the main glossary}%
18294       \else
18295         \GlossariesWarningNoLine{Glossary '@@glo@type' has no
18296           entries}%
18297       \fi
18298     \else
18299       \@glsoverwrite{\glswrite}{\@glo@type}%
18300       \immediate\write\glswrite{%
18301         \expandafter\the
18302         \csname glo@\@glo@type @filetok\endcsname}%
18303       \immediate\closeout\glswrite
18304     \fi
18305   }%
18306 }
18307 \newcommand*{\gls@glossary}[1]{%
18308   \@gls@glossary{#1}%
18309 }
18310 \newcommand*{\@gls@glossary}[2]{%
18311   \if@gls@debug
18312     \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
18313   \fi
18314 }
18315 \newcommand{\@gls@renewglossary}{%
18316   \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
18317   \let\@gls@renewglossary\@empty
18318 }
18319 \newcommand*{\gls@wrglossary}[2]{%
18320   \ifglssavewrites
18321     \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
18322     \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
18323     \expandafter{\@gls@tmp^^J}%
18324   \else
18325     \ifcsdef{glo@#1@file}%
18326     {%

```



```

18327 \expandafter\protected@write\csname glo@#1@file\endcsname{%
18328 \gls@disablepagerefexpansion}{#2}%
18329 }%
18330 {%
18331 \ifignoredglossary{#1}{}%
18332 {%
18333 \GlossariesWarning{No file defined for glossary '#1'}%
18334 }%
18335 }%
18336 \fi
18337 \endgroup\@esphack
18338 }
18339 \newcommand*{\@do@wrglossary}[1]{%
18340 \glswriteentry{#1}{\@do@wrglossary{#1}}%
18341 }
18342 \newcommand*{\glswriteentry}[2]{%
18343 \ifglsindexonlyfirst
18344 \ifglsused{#1}{#2}%
18345 \else
18346 #2%
18347 \fi
18348 }
18349 \newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
18350 \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
18351 \newcommand*{\gls@disablepagerefexpansion}{%
18352 \@for\@gls@this:=\gls@protected@pagefmts\do
18353 {%
18354 \expandafter\let\@gls@this\relax
18355 }%
18356 }
18357 \newcommand*{\gls@alphpage}{\@alph\c@page}
18358 \newcommand*{\gls@Alphpage}{\@Alph\c@page}
18359 \newcommand*{\gls@numberpage}{\number\c@page}
18360 \newcommand*{\gls@arabicpage}{\@arabic\c@page}
18361 \newcommand*{\gls@romanpage}{\romannumeral\c@page}
18362 \newcommand*{\gls@Romanpage}{\@Roman\c@page}
18363 \newcommand*{\glsaddprotectedpagefmt}[1]{%
18364 \eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
18365 \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
18366 \eappto\@wrglossarynumberhook{%
18367 \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
18368 \expandonce{\csname#1\endcsname}%
18369 \noexpand\def\expandonce{\csname#1\endcsname}{%
18370 \noexpand\@wrglossary@pageformat
18371 \expandonce{\csname gls#1page\endcsname}%
18372 \expandonce{\csname org@gls#1\endcsname}%
18373 }%
18374 }%
18375 }

```

```

18376 \newcommand*\@wrglossarynumberhook{}
18377 \newcommand{\@wrglossary@pageformat}[3]{%
18378   \ifx#3\c@page #1\else #2#3\fi
18379 }
18380 \newcommand*{\@do@wrglossary}[1]{%
18381   \ifglswrallowlocations
18382     \@do@esc@wrglossary{#1}%
18383   \else
18384     \@do@noesc@wrglossary{#1}%
18385   \fi
18386 }
18387 \newcommand*{\@do@noesc@wrglossary}[1]{%
18388   \expandafter\def\expandafter\@glslocref\expandafter{\theglsentrycounter}%
18389   \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
18390   \ifx\@glsHlocref\@glslocref
18391     \def\@glo@counterprefix{}%
18392   \else
18393     \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
18394       {\@glslocref}{\@glsHlocref}}%
18395     }%
18396     \@do@gls@getcounterprefix
18397   \fi
18398   \edef\@gls@label{\glsdetoklabel{#1}}%
18399   \@do@@wrglossary
18400 }
18401 \newif\ifglswrallowprimitivemods
18402 \glswrallowprimitivemodstrue
18403 \newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
18404   \begingroup
18405     \let\gls@orgthe\the
18406     \let\gls@orgnumber\number
18407     \let\gls@orgarabic\@arabic
18408     \let\gls@orgromannumeral\romannumeral
18409     \let\gls@orgalph\@alph
18410     \let\gls@orgAlph\@Alph
18411     \let\gls@orgRoman\@Roman
18412     \ifglswrallowprimitivemods
18413       \def\gls@the##1{%
18414         \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
18415       \def\the{\expandafter\gls@the}%
18416       \def\gls@number##1{%
18417         \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
18418       \def\number{\expandafter\gls@number}%
18419     \fi
18420     \def\@arabic##1{%
18421       \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
18422     \def\romannumeral##1{%
18423       \ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
18424     \def\@Roman##1{%

```

```

18425     \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
18426     \def\@alph##1{%
18427         \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
18428     \def\@Alph##1{%
18429         \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
18430     \@wrglossarynumberhook
18431     \gls@disablepagerefexpansion
18432     \protected@xdef\@glslocref{\theglsentrycounter}%
18433     \endgroup
18434     \@gls@checkmkidxchars\@glslocref
18435     \expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
18436         \def\@glo@counterprefix{}%
18437     \else
18438         \protected@edef\@glsHlocref{\theglsentrycounter}%
18439         \@gls@checkmkidxchars\@glsHlocref
18440         \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
18441             {\@glslocref}{\@glsHlocref}}%
18442         }%
18443         \@do@gls@getcounterprefix
18444     \fi
18445     \edef\@gls@label{\glsdetoklabel{#1}}%
18446     \@@do@@wrglossary
18447 }
18448 \newcommand*{\@@do@@wrglossary}{%
18449     \ifglxindy
18450         \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
18451         \def\@glo@range{}%
18452         \expandafter\if\@glo@prefix(\relax
18453             \def\@glo@range{:open-range}%
18454         \else
18455             \expandafter\if\@glo@prefix)\relax
18456             \def\@glo@range{:close-range}%
18457         \fi
18458         \fi
18459         \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
18460             (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
18461                 :locref \string{\@glo@counterprefix}{\@glslocref}\string" %
18462                 :attr \string"\@gls@counter\@glo@suffix\string"
18463                 \@glo@range
18464             )
18465             }%
18466         \else
18467             \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
18468             {\@glo@counterprefix}%
18469             \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
18470                 \string\glossaryentry{\csname glo@\@gls@label @index\endcsname
18471                     \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
18472             \fi
18473 }

```

```

18474 \newcommand*\@gls@getcounterprefix[2]{%
18475   \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
18476   \ifx\@gls@thisloc\@gls@thisHloc
18477     \def\@glo@counterprefix{}%
18478   \else
18479     \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
18480       \def\@glo@tmp{##2}%
18481       \ifx\@glo@tmp\@empty
18482         \def\@glo@counterprefix{}%
18483       \else
18484         \def\@glo@counterprefix{##1}%
18485       \fi
18486     }%
18487     \@gls@get@counterprefix#2.#1\end@getprefix
18488     \ifx\@glo@counterprefix\@empty
18489       \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
18490         prefixing^^Jlocation ‘#1’. You need to modify the
18491         definition of \string\theH\@gls@counter^^Jotherwise you
18492         will get the warning: “‘name{\@gls@counter.#1}’ has been^^J
18493         referenced but does not exist”}%
18494     \fi
18495   \fi
18496 }
18497 \newcommand{\@do@seeglossary}[2]{%
18498   \def\@gls@xref{#2}%
18499   \@onelevel@sanitize\@gls@xref
18500   \@gls@checkmkidxchars\@gls@xref
18501   \ifglxsindy
18502     \gls@glossary{\csname glo@#1@type\endcsname}{%
18503       (indexentry
18504         :tkey (\csname glo@#1@index\endcsname)
18505         :xref (\string"\@gls@xref\string")
18506         :attr \string"see\string"
18507       )
18508     }%
18509   \else
18510     \gls@glossary{\csname glo@#1@type\endcsname}{%
18511       \string@glossaryentry{\csname glo@#1@index\endcsname
18512         \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
18513   \fi
18514 }
18515 \def\@gls@fixbraces#1#2#3\@nil{%
18516   \ifx#2[\relax
18517     \@gls@fixbraces#1#2#3\@end@fixbraces
18518   \else
18519     \def#1{{#2#3}}%
18520   \fi
18521 }
18522 \def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%

```

```

18523 \def#1{[#2]{#3}}%
18524 }
18525 \newrobustcmd*{\glssee}[3][\seename]{%
18526 \do@seeglossary{#2}{[#1]{#3}}}
18527 \newcommand*{\@glssee}[3][\seename]{%
18528 \glssee[#1]{#3}{#2}}
18529 \newrobustcmd*{\glsseeformat}[3][\seename]{%
18530 \emph{#1} \glsseelist{#2}}
18531 \newrobustcmd*{\glsseelist}[1]{%
18532 \let\@gls@dolast\relax
18533 \let\@gls@donext\relax
18534 \@for\@gls@thislabel:=#1\do{%
18535 \ifx\@xfor@nextelement\@nnil
18536 \@gls@dolast
18537 \else
18538 \@gls@donext
18539 \fi
18540 \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
18541 \let\@gls@dolast\glsseelastsep
18542 \let\@gls@donext\glsseesep
18543 }%
18544 }
18545 \newcommand*{\glsseelastsep}{\space\andname\space}
18546 \newcommand*{\glsseesep}{, }
18547 \newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
18548 \newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
18549 \newcommand*{\gls@save@numberlist}[1]{%
18550 \ifglssavenumberlist
18551 \toks@{#1}%
18552 \edef\@do@writeaux@info{%
18553 \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
18554 }%
18555 \@onelevel@sanitize\@do@writeaux@info
18556 \protected@write\@auxout{}{\@do@writeaux@info}%
18557 \fi
18558 }
18559 \newcommand*{\warn@noprintglossary}{}%
18560 \ifcsundef{printglossary}{}%
18561 {%
18562 \@gls@warnonglossdefined
18563 \undef\printglossary
18564 }
18565 \newcommand*{\@printgloss@checkexists@noignored}[2]{%
18566 \@ifglossaryexists{#1}%
18567 {#2}%
18568 {%
18569 \ifignoredglossary{#1}%
18570 {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
18571 {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%

```

```

18572 }%
18573 }
18574 \newcommand*{\@printgloss@checkexists@allowignored}[2]{%
18575   \s@ifglossaryexists{#1}%
18576   {#2}%
18577   {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
18578 }
18579 \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18580 \newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
18581   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18582   \@printglossary{#1}{\@print@glossary}%
18583 }
18584 \newcommand*{\printglossaries}{%
18585   \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
18586 }
18587 \newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
18588   \let\@printgloss@checkexists\@printgloss@checkexists@noignored
18589   \@printglossary{#1}{\@print@noidx@glossary}%
18590 }
18591 \newcommand*{\printnoidxglossaries}{%
18592   \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
18593 }
18594 \newcommand*{\@printgloss@setsort}{}
18595 \newcommand*{\@gls@preglossaryhook}{}
18596 \newcommand{\@printglossary}[2]{%
18597   \def\@glo@type{\glsdefaulttype}%
18598   \def\glossarytitle{\csname @glotype@\@glo@type @title\endcsname}%
18599   \def\glossarytoctitle{\glossarytitle}%
18600   \let\org@glossarytitle\glossarytitle
18601   \def\@glossarystyle{%
18602     \ifx\@glossary@default@style\relax
18603       \GlossariesWarning{No default glossary style provided \MessageBreak
18604         for the glossary ‘\@glo@type’. \MessageBreak
18605         Using deprecated fallback. \MessageBreak
18606         To fix this set the style with \MessageBreak
18607         \string\setglossarystyle\space or use the \MessageBreak
18608         style key=value option}%
18609     \fi
18610   }%
18611   \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
18612   \let\@org@glossaryentrynumbers\glossaryentrynumbers
18613   \bgroup
18614   \@printgloss@setsort
18615   \setkeys{printgloss}{#1}%
18616   \@printgloss@checkexists{\@glo@type}%
18617   {%
18618     \ifx\glossarytitle\org@glossarytitle
18619     \else
18620       \expandafter\let\csname @glotype@\@glo@type @title\endcsname

```

```

18621             \glossarytitle
18622     \fi
18623     \let\currentglossary\@glo@type
18624     \let\org@glossaryentrynumbers\glossaryentrynumbers
18625     \let\glsnonextpages\@glsnonextpages
18626     \let\glsnextpages\@glsnextpages
18627     \let\nopostdesc\@nopostdesc
18628     \gls@dotoc@title
18629     \@glossarystyle
18630     \let\gls@org@glossaryentryfield\glossentry
18631     \let\gls@org@glossarysubentryfield\subglossentry
18632     \renewcommand{\glossentry}[1]{%
18633         \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
18634         \gls@org@glossaryentryfield{##1}%
18635     }%
18636     \renewcommand{\subglossentry}[2]{%
18637         \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
18638         \gls@org@glossarysubentryfield{##1}{##2}%
18639     }%
18640     \@gls@preglossaryhook
18641     #2%
18642 }%
18643 \egroup
18644 \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
18645 \global\let\warn@noprintglossary\relax
18646 }
18647 \newcommand{\@print@glossary}{%
18648     \makeatletter
18649     \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
18650     \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
18651     {}%
18652     {\null}%
18653     \ifglxindy
18654         \ifcsundef{\xdy@\@glo@type @language}%
18655         {%
18656             \edef\@do@auxoutstuff{%
18657                 \noexpand\AtEndDocument{%
18658                     \noexpand\immediate\noexpand\write\@auxout{%
18659                         \string\providecommand\string\@xdylanguage[2]{}}%
18660                     \noexpand\immediate\noexpand\write\@auxout{%
18661                         \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
18662                 }%
18663             }%
18664         }%
18665     {%
18666         \edef\@do@auxoutstuff{%
18667             \noexpand\AtEndDocument{%
18668                 \noexpand\immediate\noexpand\write\@auxout{%
18669                     \string\providecommand\string\@xdylanguage[2]{}}%

```

```

18670         \noexpand\immediate\noexpand\write\@auxout{%
18671         \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
18672         @language\endcsname}}}%
18673     }%
18674 }%
18675 }%
18676 \do@auxoutstuff
18677 \edef\do@auxoutstuff{%
18678     \noexpand\AtEndDocument{%
18679         \noexpand\immediate\noexpand\write\@auxout{%
18680         \string\providecommand\string\@gls@codepage[2]{}}%
18681         \noexpand\immediate\noexpand\write\@auxout{%
18682         \string\@gls@codepage{\@glo@type}{\@gls@codepage}}}%
18683     }%
18684 }%
18685 \do@auxoutstuff
18686 \fi
18687 \renewcommand*{\@warn@nomakeglossaries}{%
18688     \GlossariesWarningNoLine{\string\makeglossaries\space
18689     hasn't been used,^^Jthe glossaries will not be updated}%
18690 }%
18691 }
18692 \newcommand*{\@glo@sortentries}[2]{%
18693     \glosortentrieswarning
18694     \def\@glo@sortinglist{}%
18695     \def\@glo@sortinghandler{#1}%
18696     \edef\@glo@type{#2}%
18697     \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
18698     \csdef{\@glsref@#2}{}%
18699     \@for\@this@label:=\@glo@sortinglist\do{%
18700         \xifinlistcs{\@this@label}{\@glsref@#2}%
18701         {}%
18702         {%
18703             \listcsxadd{\@glsref@#2}{\@this@label}%
18704             }%
18705         \ifcsdef{\@glo@sortingchildren@\@this@label}%
18706             {%
18707                 \@glo@addchildren{#2}{\@this@label}%
18708             }%
18709         {}%
18710     }%
18711 }
18712 \newcommand*{\@glo@addchildren}[2]{%
18713     \bgroup
18714     \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
18715     \@for\@this@childlabel:=\@glo@childlist\do
18716     {%
18717         \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
18718         {}%

```



```

18719      {%
18720      \listcsxadd{@glsref{#1}}{\@this@childlabel}%
18721      }%
18722      \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
18723      {%
18724      \@glo@addchildren{#1}{\@this@childlabel}%
18725      }%
18726      {%
18727      }%
18728      }%
18729      \egroup
18730 }
18731 \newcommand*{\@glo@do@sortentries}[1]{%
18732   \ifglshasparent{#1}%
18733   {%
18734     \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
18735     \ifcsundef{@glo@sortingchildren@\@glo@parent}%
18736     {%
18737       \csdef{@glo@sortingchildren@\@glo@parent}{}%
18738     }%
18739     {}%
18740     \expandafter\@glo@sortedinsert
18741     \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
18742     \xifinlistcs{@glo@parent}{@glsref{@glo@type}}%
18743     {%
18744     }%
18745     {%
18746       \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
18747     }%
18748     }%
18749     {%
18750       \@glo@sortedinsert{\@glo@sortinglist}{#1}%
18751     }%
18752   }
18753 \newcommand*{\@glo@sortedinsert}[2]{%
18754   \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
18755 }%
18756 \newcommand*{\@glo@sorthandler@word}[2]{%
18757   \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
18758   \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
18759   \edef\@glo@do@compare{%
18760     \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
18761     {\expandonce\@gls@sort@B}%
18762     {\expandonce\@gls@sort@A}%
18763   }%
18764   \@glo@do@compare
18765 }
18766 \newcommand*{\@glo@sorthandler@letter}[2]{%
18767   \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%

```

```

18768 \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18769 \edef\glo@do@compare{%
18770   \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
18771   {\expandonce\@gls@sort@B}%
18772   {\expandonce\@gls@sort@A}%
18773 }%
18774 \glo@do@compare
18775 }
18776 \newcommand*{\@glo@sorthandler@case}[2]{%
18777   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
18778   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18779   \edef\glo@do@compare{%
18780     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
18781     {\expandonce\@gls@sort@B}%
18782     {\expandonce\@gls@sort@A}%
18783   }%
18784   \glo@do@compare
18785 }
18786 \newcommand*{\@glo@sorthandler@nocase}[2]{%
18787   \letcs\@gls@sort@A{glo@glstetoklabel{#1}@sort}%
18788   \letcs\@gls@sort@B{glo@glstetoklabel{#2}@sort}%
18789   \edef\glo@do@compare{%
18790     \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
18791     {\expandonce\@gls@sort@B}%
18792     {\expandonce\@gls@sort@A}%
18793   }%
18794   \glo@do@compare
18795 }
18796 \newcommand*{\@glo@sortmacro@word}[1]{%
18797   \ifdefstring{\@glo@default@sorttype}{standard}%
18798   {%
18799     \@glo@sortentries{\@glo@sorthandler@word}{#1}%
18800   }%
18801   {%
18802     \PackageError{glossaries}{Conflicting sort options:^^J
18803       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18804       \string\printnoidxglossary[sort=word]}{}%
18805   }%
18806 }
18807 \newcommand*{\@glo@sortmacro@letter}[1]{%
18808   \ifdefstring{\@glo@default@sorttype}{standard}%
18809   {%
18810     \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
18811   }%
18812   {%
18813     \PackageError{glossaries}{Conflicting sort options:^^J
18814       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18815       \string\printnoidxglossary[sort=letter]}{}%
18816   }%

```

```

18817 }
18818 \newcommand*{\@glo@sortmacro@standard}[1]{%
18819   \ifdefstring{\@glo@default@sorttype}{standard}%
18820   {%
18821     \ifcsdef{\@glo@sorthandler@\glsorder}%
18822     {%
18823       \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
18824     }%
18825     {%
18826       \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}%
18827     }%
18828   }%
18829   {%
18830     \PackageError{glossaries}{Conflicting sort options:^^J
18831       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18832       \string\printnoidxglossary[sort=standard]}{}%
18833   }%
18834 }
18835 \newcommand*{\@glo@sortmacro@case}[1]{%
18836   \ifdefstring{\@glo@default@sorttype}{standard}%
18837   {%
18838     \@glo@sortentries{\@glo@sorthandler@case}{#1}%
18839   }%
18840   {%
18841     \PackageError{glossaries}{Conflicting sort options:^^J
18842       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18843       \string\printnoidxglossary[sort=case]}{}%
18844   }%
18845 }
18846 \newcommand*{\@glo@sortmacro@nocase}[1]{%
18847   \ifdefstring{\@glo@default@sorttype}{standard}%
18848   {%
18849     \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
18850   }%
18851   {%
18852     \PackageError{glossaries}{Conflicting sort options:^^J
18853       \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
18854       \string\printnoidxglossary[sort=nocase]}{}%
18855   }%
18856 }
18857 \newcommand*{\@glo@sortmacro@def}[1]{%
18858   \def\@glo@sortinglist{}%
18859   \for\glsentries[#1]{\@gls@thislabel}%
18860   {%
18861     \xifinlistcs{\@gls@thislabel}{\@gls@ref@#1}%
18862   }%
18863   \listadd{\@glo@sortinglist}{\@gls@thislabel}%
18864   }%
18865   {%

```

```

18866 }%
18867 }%
18868 \cslet{@glsref@#1}{\@glo@sortinglist}%
18869 }
18870 \newcommand*{\@glo@sortmacro@def@do}[1]{%
18871   \ifinlistcs{#1}{@glsref@\@glo@type}%
18872   {}%
18873   {%
18874     \listcsadd{@glsref@\@glo@type}{#1}%
18875   }%
18876   \ifcsdef{@glo@sortingchildren@#1}%
18877   {%
18878     \@glo@addchildren{\@glo@type}{#1}%
18879   }%
18880   {}%
18881 }
18882 \newcommand*{\@glo@sortmacro@use}[1]{%
18883 \newcommand*{\@print@noidx@glossary}{%
18884   \ifcsdef{@glsref@\@glo@type}%
18885   {%
18886     \ifcsdef{@glo@sortmacro@\@glo@sorttype}%
18887     {%
18888       \csuse{@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
18889     }%
18890     {%
18891       \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
18892     }%
18893     \glossarysection[\glossarytoctitle]{\glossarytitle}%
18894     \glossarypreamble
18895     \def\@gls@currentlettergroup{}%
18896     \begin{theglossary}%
18897     \glossaryheader
18898     \glsresetentrylist
18899     \forlistcsloop{\@gls@noidx@do}{@glsref@\@glo@type}%
18900     \end{theglossary}%
18901     \glossarypostamble
18902   }%
18903   {%
18904     \@gls@noref@warn{\@glo@type}%
18905   }%
18906 }
18907 \def\glo@grabfirst#1#2\@nil{%
18908   \def\@gls@firsttok{#1}%
18909   \ifdefempty\@gls@firsttok
18910   {%
18911     \def\@glo@thislettergrp{0}%
18912   }%
18913   {%
18914     \@onelevel@sanitize\@gls@firsttok

```

```

18915 \expandafter\@glo@grabfirst\@gls@firsttok{}\}\@nil
18916 }%
18917 }
18918 \def\@glo@grabfirst#1#2\@nil{%
18919 \ifdefempty\@glo@thislettergrp
18920 {%
18921 \def\@glo@thislettergrp{glssymbols}%
18922 }%
18923 {%
18924 \count@=\uccode'#1\relax
18925 \ifnum\count@=0\relax
18926 \def\@glo@thislettergrp{glssymbols}%
18927 \else
18928 \ifdefstring\@glo@sorttype{case}%
18929 {%
18930 \count@='#1\relax
18931 }%
18932 {%
18933 }%
18934 \edef\@glo@thislettergrp{\the\count@}%
18935 \fi
18936 }%
18937 }
18938 \newcommand{\@gls@noidx@do}[1]{%
18939 \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
18940 \ifglshasparent{#1}%
18941 {%
18942 \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
18943 \ifdefvoid{\@gls@loclist}
18944 {%
18945 \subglossentry{\gls@level}{#1}{}%
18946 }%
18947 {%
18948 \subglossentry{\gls@level}{#1}%
18949 }%
18950 \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
18951 }%
18952 }%
18953 }%
18954 {%
18955 \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
18956 \expandafter\glo@grabfirst\@gls@sort{}\}\@nil
18957 \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
18958 {}%
18959 {%
18960 \ifdefempty{\@gls@currentlettergroup}{}%
18961 {%
18962 \global\let\@glo@thislettergrp\@glo@thislettergrp
18963 \glsgroupskip

```

```

18964     }%
18965     \glsgroupheading{\@glo@thislettergrp}%
18966     }%
18967     \global\let\@gls@currentlettergroup\@glo@thislettergrp
18968     \ifdefvoid{\@gls@loclist}
18969     {%
18970         \glossentry{#1}{}%
18971     }%
18972     {%
18973         \glossentry{#1}%
18974     }%
18975     \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
18976     }%
18977     }%
18978 }%
18979 }
18980 \newcommand*\@glsnoidxloclist[1]{%
18981     \def\@gls@noidxloclist@sep{}%
18982     \def\@gls@noidxloclist@prev{}%
18983     \forlistloop{\glsnoidxloclisthandler}{#1}%
18984 }
18985 \newcommand*\@glsnoidxloclisthandler[1]{%
18986     \ifdefstring{\@gls@noidxloclist@prev}{#1}%
18987     {%
18988     }%
18989     {%
18990         \@gls@noidxloclist@sep
18991         #1%
18992         \def\@gls@noidxloclist@sep{\delimN}%
18993         \def\@gls@noidxloclist@prev{#1}%
18994     }%
18995 }
18996 \newcommand*\@glsnoidxdisplayloclisthandler[1]{%
18997     \ifdefstring{\@gls@noidxloclist@prev}{#1}%
18998     {%
18999     }%
19000     {%
19001         \@gls@noidxloclist@sep
19002         \@gls@noidxloclist@prev
19003         \def\@gls@noidxloclist@prev{#1}%
19004     }%
19005 }
19006 \newcommand*\@glsnoidxdisplayloc[4]{%
19007     \setentrycounter{#1}{#2}%
19008     \csuse{#3}{#4}%
19009 }
19010 \newcommand*\@gls@reference}[3]{%
19011     \glsdoifexistsorwarn{#2}%
19012     {%

```

```

19013 \ifcsundef{@glsref@#1}{\csgdef{@glsref@#1}{}}{}%
19014 \ifinlistcs{#2}{@glsref@#1}%
19015 {}%
19016 {\listcsgadd{@glsref@#1}{#2}}%
19017 \ifcsundef{glo@glstdetoklabel{#2}@loclist}%
19018 {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%
19019 {}%
19020 \listcsgadd{glo@glstdetoklabel{#2}@loclist}{#3}%
19021 }%
19022 }
19023 \define@key{printgloss}{type}{\def@glo@type{#1}}
19024 \define@key{printgloss}{title}{%
19025 \def@glossarytitle{#1}%
19026 \let@gls@dotoc@title\relax
19027 }
19028 \define@key{printgloss}{toctitle}{%
19029 \def@glossarytoctitle{#1}%
19030 \let@gls@dotoc@title\relax
19031 }
19032 \define@key{printgloss}{style}{%
19033 \ifcsundef{@glsstyle@#1}%
19034 {%
19035 \PackageError{glossaries}%
19036 {Glossary style ‘#1’ undefined}{}%
19037 }%
19038 {%
19039 \def@glossarystyle{\setglossentrycompatibility
19040 \csname @glsstyle@#1\endcsname}%
19041 }%
19042 }
19043 \define@choicekey{printgloss}{numberedsection}%
19044 [\gls@numberedsection@val\gls@numberedsection@nr]%
19045 {false,nolabel,autolabel,nameref}[nolabel]%
19046 {%
19047 \ifcase\gls@numberedsection@nr\relax
19048 \renewcommand*{\@@glossarysecstar}{*}%
19049 \renewcommand*{\@@glossaryseclabel}{}%
19050 \or
19051 \renewcommand*{\@@glossarysecstar}{}%
19052 \renewcommand*{\@@glossaryseclabel}{}%
19053 \or
19054 \renewcommand*{\@@glossarysecstar}{*}%
19055 \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix@glo@type}}%
19056 \or
19057 \renewcommand*{\@@glossarysecstar}{*}%
19058 \renewcommand*{\@@glossaryseclabel}{%
19059 \protected@edef\@currentlabelname{\glossarytoctitle}%
19060 \label{\glsautoprefix@glo@type}}%
19061 \fi

```

```

19062 }
19063 \define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
19064   \csuse{glsnogroupskip#1}%
19065 }
19066 \define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
19067   \csuse{glsnopostdot#1}%
19068 }
19069 \newcommand*{\GlsEntryCounterLabelPrefix}{glentry-}
19070 \define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
19071   \csuse{glentrycounter#1}%
19072   \@gls@define@glossaryentrycounter
19073 }
19074 \define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
19075   \csuse{glssubentrycounter#1}%
19076   \@gls@define@glossarysubentrycounter
19077 }
19078 \define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
19079   \ifglsnonumberlist
19080     \def\glossaryentrynumbers##1{%
19081       \else
19082         \def\glossaryentrynumbers##1{##1}%
19083       \fi}
19084 \define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
19085 \newcommand*{\@glo@no@assign@sortkey}[1]{%
19086   \PackageError{glossaries}{‘sort’ key not permitted with
19087     \string\printglossary}%
19088   {The ‘sort’ key may only be used with \string\printnoidxglossary}%
19089 }
19090 \newcommand*{\@glo@assign@sortkey}[1]{%
19091   \def\@glo@sorttype{#1}%
19092 }
19093 \newcommand*{\@glsnonextpages}{%
19094   \gdef\glossaryentrynumbers##1{%
19095     \glsresetentrylist
19096   }%
19097 }
19098 \newcommand*{\@glsnextpages}{%
19099   \gdef\glossaryentrynumbers##1{%
19100     ##1\glsresetentrylist}}
19101 \newcommand*{\glsresetentrylist}{%
19102   \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
19103 \newcommand*{\glsnonextpages}{}
19104 \newcommand*{\glsnextpages}{}
19105 \@gls@define@glossaryentrycounter
19106 \@gls@define@glossarysubentrycounter
19107 \newcommand*{\glsresetsubentrycounter}{%
19108   \ifglssubentrycounter
19109     \setcounter{glossarysubentry}{0}%
19110   \fi

```



```

19111 }
19112 \newcommand*{\glsresetentrycounter}{%
19113   \ifglssentrycounter
19114     \setcounter{glossaryentry}{0}%
19115   \fi
19116 }
19117 \newcommand*{\glsstepentry}[1]{%
19118   \ifglssentrycounter
19119     \refstepcounter{glossaryentry}%
19120     \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19121   \fi
19122 }
19123 \newcommand*{\glsstepsubentry}[1]{%
19124   \ifglsssubentrycounter
19125     \edef\currentglsssubentry{\glsdetoklabel{#1}}%
19126     \refstepcounter{glossarysubentry}%
19127     \label{\GlsEntryCounterLabelPrefix\currentglsssubentry}%
19128   \fi
19129 }
19130 \newcommand*{\glsrefentry}[1]{%
19131   \ifglssentrycounter
19132     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19133   \else
19134     \ifglsssubentrycounter
19135       \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
19136     \else
19137       \gls{#1}%
19138     \fi
19139   \fi
19140 }
19141 \newcommand*{\glsentrycounterlabel}{%
19142   \ifglssentrycounter
19143     \theglossaryentry.\space
19144   \fi
19145 }
19146 \newcommand*{\glsssubentrycounterlabel}{%
19147   \ifglsssubentrycounter
19148     \theglossarysubentry)\space
19149   \fi
19150 }
19151 \newcommand*{\glsentryitem}[1]{%
19152   \ifglssentrycounter
19153     \glsstepentry{#1}\glsentrycounterlabel
19154   \else
19155     \glsresetsubentrycounter
19156   \fi
19157 }
19158 \newcommand*{\glssubentryitem}[1]{%
19159   \ifglsssubentrycounter

```

```

19160 \glsstepsubentry{#1}\glssubentrycounterlabel
19161 \fi
19162 }
19163 \ifcsundef{theglossary}%
19164 {%
19165 \newenvironment{theglossary}{}{}%
19166 }%
19167 {%
19168 \@gls@warnontheglossdefined
19169 \renewenvironment{theglossary}{}{}%
19170 }
19171 \newcommand*{\glossaryheader}{}
19172 \newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
19173 \providecommand*{\compatibleglossentry}[2]{%
19174 \toks@{#2}%
19175 \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
19176 {\noexpand\glsnamefont
19177 {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
19178 {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
19179 {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
19180 {\the\toks@}%
19181 }%
19182 \@do@glossentry
19183 }
19184 \newcommand*{\glossentryname}[1]{%
19185 \glsdoifexistsorwarn{#1}%
19186 {%
19187 \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
19188 \expandafter\glsnamefont\expandafter{\glo@name}%
19189 }%
19190 }
19191 \newcommand*{\Glossentryname}[1]{%
19192 \glsdoifexistsorwarn{#1}%
19193 {%
19194 \glsnamefont{\Glsentryname{#1}}%
19195 }%
19196 }
19197 \newcommand*{\glossentrydesc}[1]{%
19198 \glsdoifexistsorwarn{#1}%
19199 {%
19200 \glsentrydesc{#1}%
19201 }%
19202 }
19203 \newcommand*{\Glossentrydesc}[1]{%
19204 \glsdoifexistsorwarn{#1}%
19205 {%
19206 \Glsentrydesc{#1}%
19207 }%
19208 }

```

```

19209 \newcommand*{\glossentrysymbol}[1]{%
19210   \glsdoifexistsorwarn{#1}%
19211   {%
19212     \glsentrysymbol{#1}%
19213   }%
19214 }
19215 \newcommand*{\Glossentrysymbol}[1]{%
19216   \glsdoifexistsorwarn{#1}%
19217   {%
19218     \Glsentrysymbol{#1}%
19219   }%
19220 }
19221 \providecommand*{\compatiblesubglossentry}[3]{%
19222   \toks@{#3}%
19223   \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
19224     {#2}%
19225     {\noexpand\glsnamefont
19226       {\expandafter\expandonce\csname glo@#2@name\endcsname}}}%
19227     {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
19228     {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
19229     {\the\toks@}%
19230   }%
19231   \@do@subglossentry
19232 }
19233 \newcommand*{\setglossentrycompatibility}{%
19234   \let\glossentry\compatibleglossentry
19235   \let\subglossentry\compatiblesubglossentry
19236 }
19237 \setglossentrycompatibility
19238 \newcommand{\glossaryentryfield}[5]{%
19239   \GlossariesWarning
19240   {Deprecated use of \string\glossaryentryfield.^^J
19241     I recommend you change to \string\glossentry.^^J
19242     If you've just upgraded, try removing your gls auxiliary
19243     files^^J and recompile}%
19244   \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
19245 \newcommand*{\glossarysubentryfield}[6]{%
19246   \GlossariesWarning
19247   {Deprecated use of \string\glossarysubentryfield.^^J
19248     I recommend you change to \string\subglossentry.^^J
19249     If you've just upgraded, try removing your gls auxiliary
19250     files^^J and recompile}%
19251   \glstarget{#2}{\strut}#4. #6\par}
19252 \newcommand*{\glsgroupskip}{}
19253 \newcommand*{\glsgroupheading}[1]{}
19254 \newcommand*{\glsgrouptitle}[1]{%
19255   \@gls@getgrouptitle{#1}{\@gls@grptitle}%
19256   \@gls@grptitle
19257 }

```

```

19258 \newcommand*{\@gls@getgrouptitle}[2]{%
19259   \dtl@ifsingle{#1}%
19260   {%
19261     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19262   }%
19263   {%
19264     \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
19265                 or test{\ifstrequal{#1}{glsnumbers}}}%
19266     {%
19267       \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19268     }%
19269     {%
19270       \def#2{#1}%
19271     }%
19272   }%
19273 }
19274 \newcommand*{\@gls@noidx@getgrouptitle}[2]{%
19275   \DTLifint{#1}%
19276   {\edef#2{\char#1\relax}}%
19277   {%
19278     \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
19279   }%
19280 }
19281 \newcommand*{\glsgetgrouplabel}[1]{%
19282   \ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
19283   \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
19284 \newcommand*{\setentrycounter}[2][ ]{%
19285   \def\@glo@counterprefix{#1}%
19286   \ifx\@glo@counterprefix\empty
19287     \def\@glo@counterprefix{.}%
19288   \else
19289     \def\@glo@counterprefix{.#1.}%
19290   \fi
19291   \def\glsentrycounter{#2}%
19292 }
19293 \newcommand*{\setglossarystyle}[1]{%
19294   \ifcsundef{@glsstyle@#1}%
19295   {%
19296     \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
19297   }%
19298   {%
19299     \csname @glsstyle@#1\endcsname
19300   }%
19301   \ifx\@glossary@default@style\relax
19302     \protected@edef\@glossary@default@style{#1}%
19303   \fi
19304 }
19305 \newcommand*{\glossarystyle}[1]{%
19306   \ifcsundef{@glsstyle@#1}%

```

```

19307 {%
19308   \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
19309 }%
19310 {%
19311   \GlossariesWarning
19312   {Deprecated command \string\glossarystyle.^~J
19313     I recommend you switch to \string\setglossarystyle\space unless
19314     you want to maintain backward compatibility}%
19315   \setglossentrycompatibility
19316   \csname @glsstyle@#1\endcsname
19317   \ifcsdef{@glscompstyle@#1}%
19318     {\setglossentrycompatibility\csuse{@glscompstyle@#1}}%
19319   }%
19320 }%
19321 \ifx\@glossary@default@style\relax
19322   \protected@edef\@glossary@default@style{#1}%
19323 \fi
19324 }
19325 \newcommand{\newglossarystyle}[2]{%
19326   \ifcsundef{@glsstyle@#1}%
19327   {%
19328     \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
19329   }%
19330   {%
19331     \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
19332   }%
19333 }
19334 \newcommand{\renewglossarystyle}[2]{%
19335   \ifcsundef{@glsstyle@#1}%
19336   {%
19337     \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
19338   }%
19339   {%
19340     \csdef{@glsstyle@#1}{#2}%
19341   }%
19342 }
19343 \newcommand*{\glsnamefont}[1]{#1}
19344 \ifcsundef{hyperlink}%
19345 {%
19346   \def\glshypernumber#1{#1}%
19347 }%
19348 {%
19349   \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
19350 }
19351 \def\@glshypernumber#1\nohyperpage#2#3\@nil{%
19352   \ifx\#1\%
19353     \else
19354       \@delimR#1\delimR\delimR\%
19355   \fi

```

```

19356 \ifx\#2\%
19357 \else
19358   #2%
19359 \fi
19360 \ifx\#3\%
19361 \else
19362   \@glshypernumber#3\@nil
19363 \fi
19364 }
19365 \def\@delimR#1\delimR #2\delimR #3\%{
19366 \ifx\#2\%
19367   \@delimN{#1}%
19368 \else
19369   \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
19370 \fi}
19371 \def\@delimN#1{\@delimN#1\delimN \delimN\}
19372 \def\@delimN#1\delimN #2\delimN#3\%{
19373 \ifx\#3\%
19374   \@gls@numberlink{#1}%
19375 \else
19376   \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
19377 \fi
19378 }
19379 \def\@gls@numberlink#1{%
19380 \begingroup
19381 \toks@={}%
19382 \@gls@removespaces#1 \@nil
19383 \endgroup}
19384 \def\@gls@removespaces#1 #2\@nil{%
19385 \toks@=\expandafter{\the\toks@#1}%
19386 \ifx\#2\%
19387   \edef\x{\the\toks@}%
19388   \ifx\x\empty
19389   \else
19390     \hyperlink{\glstrycounter\@glo@counterprefix\the\toks@}%
19391       {\the\toks@}%
19392   \fi
19393 \else
19394   \@gls@ReturnAfterFi{%
19395     \@gls@removespaces#2\@nil
19396   }%
19397 \fi
19398 }
19399 \long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
19400 \newcommand*{\hyperrm}[1]{\textrm{\glshypernumber{#1}}}
19401 \newcommand*{\hypersf}[1]{\textsf{\glshypernumber{#1}}}
19402 \newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}
19403 \newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
19404 \newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}

```

```

19405 \newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}
19406 \newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
19407 \newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
19408 \newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
19409 \newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}
19410 \newcommand{\oldacronym}[4][\gls@label]{%
19411   \def\gls@label{#2}%
19412   \newacronym[#4]{#1}{#2}{#3}%
19413   \ifcsundef{xspace}%
19414   {%
19415     \expandafter\edef\csname#1\endcsname{%
19416       \noexpand@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
19417   }%
19418 }%
19419 {%
19420   \expandafter\edef\csname#1\endcsname{%
19421     \noexpand@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
19422       \noexpand\gls{#1}\noexpand\xspace}%
19423   }%
19424 }%
19425 }
19426 \newcommand{\newacronym}[4][{}]{
19427 \newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
19428 \newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
19429 \newcommand*{\glsshortkey}{short}
19430 \newcommand*{\glsshortpluralkey}{shortplural}
19431 \newcommand*{\glslongkey}{long}
19432 \newcommand*{\glslongpluralkey}{longplural}
19433 \newrobustcmd*{\acrfull}{\@gls@hyp@opt\@ns@acrfull}
19434 \newcommand*\ns@acrfull[2][{}]{%
19435   \new@ifnextchar[{\@acrfull{#1}{#2}}{%
19436     {\@acrfull{#1}{#2}}[{}]}%
19437 }
19438 \def\@acrfull#1#2[#3]{%
19439   \acrfullfmt{#1}{#2}{#3}%
19440 }
19441 \newcommand*{\acrfullfmt}[3]{%
19442   \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
19443 }
19444 \newcommand{\acrlinkfullformat}[5]{%
19445   \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[{}]}%
19446 }
19447 \newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
19448 \newrobustcmd{\glsspace}{\space}
19449 \newrobustcmd*{\Acrfull}{\@gls@hyp@opt\@ns@Acrfull}
19450 \newcommand*\ns@Acrfull[2][{}]{%
19451   \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
19452     {\@Acrfull{#1}{#2}}[{}]}%
19453 }

```

```

19454 \def\@Acrfull#1#2[#3]{%
19455   \Acrfullfmt{#1}{#2}{#3}%
19456 }
19457 \newcommand*\@Acrfullfmt}[3]{%
19458   \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
19459 }
19460 \newrobustcmd*\@ACRfull{\@gls@hyp@opt\@ns@ACRfull}
19461 \newcommand*\@ns@ACRfull[2][{}]{%
19462   \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
19463     {\@ACRfull{#1}{#2}[]}%
19464 }
19465 \def\@ACRfull#1#2[#3]{%
19466   \ACRfullfmt{#1}{#2}{#3}%
19467 }
19468 \newcommand*\@ACRfullfmt}[3]{%
19469   \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
19470 }
19471 \newrobustcmd*\@acrfullpl{\@gls@hyp@opt\@ns@acrfullpl}
19472 \newcommand*\@ns@acrfullpl[2][{}]{%
19473   \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
19474     {\@acrfullpl{#1}{#2}[]}%
19475 }
19476 \def\@acrfullpl#1#2[#3]{%
19477   \acrfullplfmt{#1}{#2}{#3}%
19478 }
19479 \newcommand*\@acrfullplfmt}[3]{%
19480   \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
19481 }
19482 \newrobustcmd*\@Acrfullpl{\@gls@hyp@opt\@ns@Acrfullpl}
19483 \newcommand*\@ns@Acrfullpl[2][{}]{%
19484   \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
19485     {\@Acrfullpl{#1}{#2}[]}%
19486 }
19487 \def\@Acrfullpl#1#2[#3]{%
19488   \Acrfullplfmt{#1}{#2}{#3}%
19489 }
19490 \newcommand*\@Acrfullplfmt}[3]{%
19491   \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
19492 }
19493 \newrobustcmd*\@ACRfullpl{\@gls@hyp@opt\@ns@ACRfullpl}
19494 \newcommand*\@ns@ACRfullpl[2][{}]{%
19495   \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
19496     {\@ACRfullpl{#1}{#2}[]}%
19497 }
19498 \def\@ACRfullpl#1#2[#3]{%
19499   \ACRfullplfmt{#1}{#2}{#3}%
19500 }
19501 \newcommand*\@ACRfullplfmt}[3]{%
19502   \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%

```



```

19503 }
19504 \newcommand{\acronymfont}[1]{#1}
19505 \newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
19506 \newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
19507 \newtoks\glskeylisttok
19508 \newtoks\glslabeltok
19509 \newtoks\glsshorttok
19510 \newtoks\glslongtok
19511 \newcommand*{\newacronymhook}{}
19512 \newcommand*{\SetGenericNewAcronym}{%
19513   \let\@Gls@entryname\@Gls@acrenryname
19514   \renewcommand{\newacronym}[4][]{%
19515     \ifdefempty{\@glsacronymlists}%
19516       {%
19517         \def\@glo@type{\acronymtype}%
19518         \setkeys{glossentry}{##1}%
19519         \DeclareAcronymList{\@glo@type}%
19520       }%
19521     }%
19522     \glskeylisttok{##1}%
19523     \glslabeltok{##2}%
19524     \glsshorttok{##3}%
19525     \glslongtok{##4}%
19526     \newacronymhook
19527     \protected@edef\@do@newglossaryentry{%
19528       \noexpand\newglossaryentry{\the\glslabeltok}%
19529       {%
19530         type=\acronymtype,%
19531         name={\expandonce{\acronymentry{##2}}},%
19532         sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
19533         text={\the\glsshorttok},%
19534         short={\the\glsshorttok},%
19535         shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
19536         long={\the\glslongtok},%
19537         longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
19538         \GenericAcronymFields,%
19539         \the\glskeylisttok
19540       }%
19541     }%
19542     \@do@newglossaryentry
19543   }%
19544   \renewcommand*{\acrfullfmt}[3]{%
19545     \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
19546   \renewcommand*{\Acrfullfmt}[3]{%
19547     \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
19548   \renewcommand*{\ACRfullfmt}[3]{%
19549     \glslink[##1]{##2}{%
19550       \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
19551   \renewcommand*{\acrfullplfmt}[3]{%

```

```

19552 \glslink{##1}{##2}{\genplacrfullformat{##2}{##3}}}%
19553 \renewcommand*{\Acrfullplfmt}[3]{%
19554 \glslink{##1}{##2}{\Genplacrfullformat{##2}{##3}}}%
19555 \renewcommand*{\ACRfullplfmt}[3]{%
19556 \glslink{##1}{##2}{%
19557 \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}}}%
19558 \renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
19559 \renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
19560 \renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
19561 \renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
19562 }
19563 \newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
19564 \newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
19565 \newcommand*{\acronymsort}[2]{#1}
19566 \newcommand*{\setacronymstyle}[1]{%
19567 \ifcsundef{@glsacr@dispstyle@#1}
19568 {%
19569 \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
19570 }%
19571 {%
19572 \ifdefempty{\@glsacronymlists}%
19573 {%
19574 \DeclareAcronymList{\acronymtype}%
19575 }%
19576 }%
19577 \SetGenericNewAcronym
19578 \GlsUseAcrStyleDefs{#1}%
19579 \@for\@gls@type:=\@glsacronymlists\do{%
19580 \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDisplayStyle{#1}}%
19581 }%
19582 }%
19583 }
19584 \newcommand*{\newacronymstyle}[3]{%
19585 \ifcsdef{@glsacr@dispstyle@#1}%
19586 {%
19587 \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
19588 }%
19589 {%
19590 \csdef{@glsacr@dispstyle@#1}{#2}%
19591 \csdef{@glsacr@styledefs@#1}{#3}%
19592 }%
19593 }
19594 \newcommand*{\renewacronymstyle}[3]{%
19595 \ifcsdef{@glsacr@dispstyle@#1}%
19596 {%
19597 \csdef{@glsacr@dispstyle@#1}{#2}%
19598 \csdef{@glsacr@styledefs@#1}{#3}%
19599 }%
19600 {%

```

```

19601 \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
19602 }%
19603 }
19604 \newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
19605 \newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
19606 \newacronymstyle{long-short}%
19607 {%
19608 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19609 }%
19610 {%
19611 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19612 \renewcommand*{\genacrfullformat}[2]{%
19613 \glsentrylong{##1}##2\space
19614 (\protect\firstacronymfont{\glsentryshort{##1}})%
19615 }%
19616 \renewcommand*{\Genacrfullformat}[2]{%
19617 \Glsentrylong{##1}##2\space
19618 (\protect\firstacronymfont{\glsentryshort{##1}})%
19619 }%
19620 \renewcommand*{\genplacrfullformat}[2]{%
19621 \glsentrylongpl{##1}##2\space
19622 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19623 }%
19624 \renewcommand*{\Genplacrfullformat}[2]{%
19625 \Glsentrylongpl{##1}##2\space
19626 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19627 }%
19628 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19629 \renewcommand*{\acronymsort}[2]{##1}%
19630 \renewcommand*{\acronymfont}[1]{##1}%
19631 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19632 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19633 }
19634 \newacronymstyle{long-sp-short}%
19635 {%
19636 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19637 }%
19638 {%
19639 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19640 \renewcommand*{\genacrfullformat}[2]{%
19641 \glsentrylong{##1}##2\glsacspace{##1}%
19642 (\protect\firstacronymfont{\glsentryshort{##1}})%
19643 }%
19644 \renewcommand*{\Genacrfullformat}[2]{%
19645 \Glsentrylong{##1}##2\glsacspace{##1}%
19646 (\protect\firstacronymfont{\glsentryshort{##1}})%
19647 }%
19648 \renewcommand*{\genplacrfullformat}[2]{%
19649 \glsentrylongpl{##1}##2\glsacspace{##1}%

```

```

19650 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19651 }%
19652 \renewcommand*{\Genplacrfullformat}[2]{%
19653 \Glsentrylongpl{##1}##2\glsacspace{##1}%
19654 (\protect\firstacronymfont{\glsentryshortpl{##1}})%
19655 }%
19656 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19657 \renewcommand*{\acronymsort}[2]{##1}%
19658 \renewcommand*{\acronymfont}[1]{##1}%
19659 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19660 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19661 }
19662 \newcommand*{\glsacspace}[1]{%
19663 \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{#1}})}%
19664 \ifdim\dimen@<3em~\else\spacefi
19665 }
19666 \newacronymstyle{short-long}%
19667 {%
19668 \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
19669 }%
19670 {%
19671 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19672 \renewcommand*{\genacrfullformat}[2]{%
19673 \protect\firstacronymfont{\glsentryshort{##1}}##2\space
19674 (\glsentrylong{##1})%
19675 }%
19676 \renewcommand*{\Genacrfullformat}[2]{%
19677 \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
19678 (\glsentrylong{##1})%
19679 }%
19680 \renewcommand*{\genplacrfullformat}[2]{%
19681 \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
19682 (\glsentrylongpl{##1})%
19683 }%
19684 \renewcommand*{\Genplacrfullformat}[2]{%
19685 \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
19686 (\glsentrylongpl{##1})%
19687 }%
19688 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19689 \renewcommand*{\acronymsort}[2]{##1}%
19690 \renewcommand*{\acronymfont}[1]{##1}%
19691 \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
19692 \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
19693 }
19694 \newacronymstyle{long-sc-short}%
19695 {%
19696 \GlsUseAcrEntryDispStyle{long-short}%
19697 }%
19698 {%

```

```

19699 \GlsUseAcrStyleDefs{long-short}%
19700 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19701 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19702 }
19703 \newacronymstyle{long-sm-short}%
19704 {%
19705 \GlsUseAcrEntryDispStyle{long-short}%
19706 }%
19707 {%
19708 \GlsUseAcrStyleDefs{long-short}%
19709 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19710 \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
19711 }
19712 \newacronymstyle{sc-short-long}%
19713 {%
19714 \GlsUseAcrEntryDispStyle{short-long}%
19715 }%
19716 {%
19717 \GlsUseAcrStyleDefs{short-long}%
19718 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19719 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19720 }
19721 \newacronymstyle{sm-short-long}%
19722 {%
19723 \GlsUseAcrEntryDispStyle{short-long}%
19724 }%
19725 {%
19726 \GlsUseAcrStyleDefs{short-long}%
19727 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19728 \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
19729 }
19730 \newacronymstyle{long-short-desc}%
19731 {%
19732 \GlsUseAcrEntryDispStyle{long-short}%
19733 }%
19734 {%
19735 \GlsUseAcrStyleDefs{long-short}%
19736 \renewcommand*{\GenericAcronymFields}{}%
19737 \renewcommand*{\acronymsort}[2]{##2}%
19738 \renewcommand*{\acronymentry}[1]{%
19739 \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19740 }
19741 \newacronymstyle{long-sp-short-desc}%
19742 {%
19743 \GlsUseAcrEntryDispStyle{long-sp-short}%
19744 }%
19745 {%
19746 \GlsUseAcrStyleDefs{long-sp-short}%
19747 \renewcommand*{\GenericAcronymFields}{}%

```

```

19748 \renewcommand*{\acronymsort}[2]{##2}%
19749 \renewcommand*{\acronymentry}[1]{%
19750     \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##1}})}%
19751 }
19752 \newacronymstyle{long-sc-short-desc}%
19753 {%
19754     \GlsUseAcrEntryDisplayStyle{long-sc-short}%
19755 }%
19756 {%
19757     \GlsUseAcrStyleDefs{long-sc-short}%
19758     \renewcommand*{\GenericAcronymFields}{}%
19759     \renewcommand*{\acronymsort}[2]{##2}%
19760     \renewcommand*{\acronymentry}[1]{%
19761         \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19762 }
19763 \newacronymstyle{long-sm-short-desc}%
19764 {%
19765     \GlsUseAcrEntryDisplayStyle{long-sm-short}%
19766 }%
19767 {%
19768     \GlsUseAcrStyleDefs{long-sm-short}%
19769     \renewcommand*{\GenericAcronymFields}{}%
19770     \renewcommand*{\acronymsort}[2]{##2}%
19771     \renewcommand*{\acronymentry}[1]{%
19772         \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19773 }
19774 \newacronymstyle{short-long-desc}%
19775 {%
19776     \GlsUseAcrEntryDisplayStyle{short-long}%
19777 }%
19778 {%
19779     \GlsUseAcrStyleDefs{short-long}%
19780     \renewcommand*{\GenericAcronymFields}{}%
19781     \renewcommand*{\acronymsort}[2]{##2}%
19782     \renewcommand*{\acronymentry}[1]{%
19783         \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19784 }
19785 \newacronymstyle{sc-short-long-desc}%
19786 {%
19787     \GlsUseAcrEntryDisplayStyle{sc-short-long}%
19788 }%
19789 {%
19790     \GlsUseAcrStyleDefs{sc-short-long}%
19791     \renewcommand*{\GenericAcronymFields}{}%
19792     \renewcommand*{\acronymsort}[2]{##2}%
19793     \renewcommand*{\acronymentry}[1]{%
19794         \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
19795 }
19796 \newacronymstyle{sm-short-long-desc}%

```

```

19797 {%
19798   \GlsUseAcrEntryDispStyle{sm-short-long}%
19799 }%
19800 {%
19801   \GlsUseAcrStyleDefs{sm-short-long}%
19802   \renewcommand*{\GenericAcronymFields}{}%
19803   \renewcommand*{\acronymsort}[2]{##2}%
19804   \renewcommand*{\acronymentry}[1]{%
19805     \glstrylong{##1}\space (\acronymfont{\glstryshort{##1}})}%
19806 }
19807 \newacronymstyle{dua}%
19808 {%
19809   \ifdefempty\glscustomtext
19810   {%
19811     \ifglshaslong{\glslabel}%
19812     {%
19813       \glssifplural
19814       {%
19815         \glscapscase
19816         {%
19817           \glstrylongpl{\glslabel}\glssinsert
19818         }%
19819         {%
19820           \Glsentrylongpl{\glslabel}\glssinsert
19821         }%
19822         {%
19823           \mfirstucMakeUppercase
19824             {\glstrylongpl{\glslabel}\glssinsert}%
19825         }%
19826       }%
19827     }%
19828     \glscapscase
19829     {%
19830       \glstrylong{\glslabel}\glssinsert
19831     }%
19832     {%
19833       \Glsentrylong{\glslabel}\glssinsert
19834     }%
19835     {%
19836       \mfirstucMakeUppercase
19837         {\glstrylong{\glslabel}\glssinsert}%
19838     }%
19839   }%
19840 }%
19841 {%
19842   \glsgenentryfmt
19843 }%
19844 }%
19845 {\glscustomtext\glssinsert}%

```

```

19846 }%
19847 {%
19848 \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19849 \renewcommand*{\acrfullfmt}[3]{%
19850 \glslink{##1}{##2}{\glsentrylong{##2}##3\space
19851 (\acronymfont{\glsentryshort{##2}})}}%
19852 \renewcommand*{\Acrfullfmt}[3]{%
19853 \glslink{##1}{##2}{\Glsentrylong{##2}##3\space
19854 (\acronymfont{\glsentryshort{##2}})}}%
19855 \renewcommand*{\ACRfullfmt}[3]{%
19856 \glslink{##1}{##2}{%
19857 \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
19858 (\acronymfont{\glsentryshort{##2}})}}}%
19859 \renewcommand*{\acrfullplfmt}[3]{%
19860 \glslink{##1}{##2}{\glsentrylongpl{##2}##3\space
19861 (\acronymfont{\glsentryshortpl{##2}})}}%
19862 \renewcommand*{\Acrfullplfmt}[3]{%
19863 \glslink{##1}{##2}{\Glsentrylongpl{##2}##3\space
19864 (\acronymfont{\glsentryshortpl{##2}})}}%
19865 \renewcommand*{\ACRfullplfmt}[3]{%
19866 \glslink{##1}{##2}{%
19867 \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
19868 (\acronymfont{\glsentryshortpl{##2}})}}}%
19869 \renewcommand*{\glsentryfull}[1]{%
19870 \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
19871 }%
19872 \renewcommand*{\Glsentryfull}[1]{%
19873 \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
19874 }%
19875 \renewcommand*{\glsentryfullpl}[1]{%
19876 \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
19877 }%
19878 \renewcommand*{\Glsentryfullpl}[1]{%
19879 \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
19880 }%
19881 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19882 \renewcommand*{\acronymsort}[2]{##1}%
19883 \renewcommand*{\acronymfont}[1]{##1}%
19884 \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19885 }
19886 \newacronymstyle{dua-desc}%
19887 {%
19888 \GlsUseAcrEntryDisplayStyle{dua}%
19889 }%
19890 {%
19891 \GlsUseAcrStyleDefs{dua}%
19892 \renewcommand*{\GenericAcronymFields}{}%
19893 \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
19894 \renewcommand*{\acronymsort}[2]{##2}%

```



```

19895 }%
19896 \newacronymstyle{footnote}%
19897 {%
19898   \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
19899 }%
19900 {%
19901   \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
19902   \glshyperfirstfalse
19903   \renewcommand*{\genacrfullformat}[2]{%
19904     \protect\firstacronymfont{\glsentryshort{##1}}##2%
19905     \protect\footnote{\glsentrylong{##1}}%
19906   }%
19907   \renewcommand*{\Genacrfullformat}[2]{%
19908     \firstacronymfont{\Glsentryshort{##1}}##2%
19909     \protect\footnote{\glsentrylong{##1}}%
19910   }%
19911   \renewcommand*{\genplacrfullformat}[2]{%
19912     \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
19913     \protect\footnote{\glsentrylongpl{##1}}%
19914   }%
19915   \renewcommand*{\Genplacrfullformat}[2]{%
19916     \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
19917     \protect\footnote{\glsentrylongpl{##1}}%
19918   }%
19919   \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
19920   \renewcommand*{\acronymsort}[2]{##1}%
19921   \renewcommand*{\acronymfont}[1]{##1}%
19922   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
19923   \renewcommand*{\acrfullfmt}[3]{%
19924     \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
19925       (\glsentrylong{##2})}%
19926   \renewcommand*{\Acrfullfmt}[3]{%
19927     \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
19928       (\glsentrylong{##2})}%
19929   \renewcommand*{\ACRfullfmt}[3]{%
19930     \glslink[##1]{##2}{%
19931       \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
19932         (\glsentrylong{##2})}}}%
19933   \renewcommand*{\acrfullplfmt}[3]{%
19934     \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
19935       (\glsentrylongpl{##2})}%
19936   \renewcommand*{\Acrfullplfmt}[3]{%
19937     \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
19938       (\glsentrylongpl{##2})}%
19939   \renewcommand*{\ACRfullplfmt}[3]{%
19940     \glslink[##1]{##2}{%
19941       \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
19942         (\glsentrylongpl{##2})}}}%
19943   \renewcommand*{\glsentryfull}[1]{%

```

```

19944 \acronymfont{\glentryshort{##1}}\space(\glentrylong{##1})}%
19945 \renewcommand*{\Glsentryfull}[1]{%
19946 \acronymfont{\Glsentryshort{##1}}\space(\glentrylong{##1})}%
19947 \renewcommand*{\glentryfullpl}[1]{%
19948 \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
19949 \renewcommand*{\Glsentryfullpl}[1]{%
19950 \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
19951 }
19952 \newacronymstyle{footnote-sc}%
19953 {%
19954 \GlsUseAcrEntryDisplayStyle{footnote}%
19955 }%
19956 {%
19957 \GlsUseAcrStyleDefs{footnote}%
19958 \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
19959 \renewcommand{\acronymfont}[1]{\textsc{##1}}%
19960 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
19961 }%
19962 \newacronymstyle{footnote-sm}%
19963 {%
19964 \GlsUseAcrEntryDisplayStyle{footnote}%
19965 }%
19966 {%
19967 \GlsUseAcrStyleDefs{footnote}%
19968 \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
19969 \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
19970 \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
19971 }%
19972 \newacronymstyle{footnote-desc}%
19973 {%
19974 \GlsUseAcrEntryDisplayStyle{footnote}%
19975 }%
19976 {%
19977 \GlsUseAcrStyleDefs{footnote}%
19978 \renewcommand*{\GenericAcronymFields}{}%
19979 \renewcommand*{\acronymsort}[2]{##2}%
19980 \renewcommand*{\acronymentry}[1]{%
19981 \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})%
19982 }
19983 \newacronymstyle{footnote-sc-desc}%
19984 {%
19985 \GlsUseAcrEntryDisplayStyle{footnote-sc}%
19986 }%
19987 {%
19988 \GlsUseAcrStyleDefs{footnote-sc}%
19989 \renewcommand*{\GenericAcronymFields}{}%
19990 \renewcommand*{\acronymsort}[2]{##2}%
19991 \renewcommand*{\acronymentry}[1]{%
19992 \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})%

```

```

19993 }
19994 \newacronymstyle{footnote-sm-desc}%
19995 {%
19996   \GlsUseAcrEntryDispStyle{footnote-sm}%
19997 }%
19998 {%
19999   \GlsUseAcrStyleDefs{footnote-sm}%
20000   \renewcommand*{\GenericAcronymFields}{}%
20001   \renewcommand*{\acronymsort}[2]{##2}%
20002   \renewcommand*{\acronymentry}[1]{%
20003     \glstrylong{##1}\space (\acronymfont{\glstryshort{##1}})}%
20004 }
20005 \newcommand*{\DefineAcronymSynonyms}{%
20006   \let\acs\acrshort
20007   \let\Acs\Acrshort
20008   \let\acsp\acrshortpl
20009   \let\Acsp\Acrshortpl
20010   \let\acl\acrlong
20011   \let\aclp\acrlongpl
20012   \let\Acl\Acrlong
20013   \let\Aclp\Acrlongpl
20014   \let\acf\acrfull
20015   \let\acfp\acrfullpl
20016   \let\Acf\Acrfull
20017   \let\Acfp\Acrfullpl
20018   \let\ac\gls
20019   \let\Ac\Gls
20020   \let\acp\glspl
20021   \let\Acp\Glspl
20022 }
20023 \ifglssacrshortcuts
20024   \DefineAcronymSynonyms
20025 \fi
20026 \newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
20027   \defglstryfmt[#1]{\glsgenentryfmt}%
20028 }
20029 \newcommand*{\DefaultNewAcronymDef}{%
20030   \edef\@do@newglossaryentry{%
20031     \noexpand\newglossaryentry{\the\glslabeltok}%
20032     {%
20033       type=\acronymtype,%
20034       name={\the\glsshorttok},%
20035       sort={\the\glsshorttok},%
20036       text={\the\glsshorttok},%
20037       first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
20038       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20039       firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
20040                     {\noexpand\expandonce\noexpand\@glo@shortpl}},%
20041       short={\the\glsshorttok},%

```

```

20042     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20043     long={\the\glslongtok},%
20044     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20045     description={\the\glslongtok},%
20046     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20047     \the\glskeylisttok
20048   }%
20049 }%
20050 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20051 \let\@org@gls@assign@plural\gls@assign@plural
20052 \let\@org@gls@assign@descplural\gls@assign@descplural
20053 \def\gls@assign@firstpl##1##2{%
20054   \@@gls@expand@field{##1}{firstpl}{##2}%
20055 }%
20056 \def\gls@assign@plural##1##2{%
20057   \@@gls@expand@field{##1}{plural}{##2}%
20058 }%
20059 \def\gls@assign@descplural##1##2{%
20060   \@@gls@expand@field{##1}{descplural}{##2}%
20061 }%
20062 \do@newglossaryentry
20063 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20064 \let\gls@assign@plural\@org@gls@assign@plural
20065 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20066 }
20067 \newcommand*{\SetDefaultAcronymStyle}{%
20068   \@for\@gls@type:=\@glsacronymlists\do{%
20069     \SetDefaultAcronymDisplayStyle{\@gls@type}%
20070   }%
20071   \renewcommand{\newacronym}[4][ ]{%
20072     \ifx\@glsacronymlists\@empty
20073       \def\@glo@type{\acronymtype}%
20074       \setkeys{glossentry}{##1}%
20075       \DeclareAcronymList{\@glo@type}%
20076       \SetDefaultAcronymDisplayStyle{\@glo@type}%
20077     \fi
20078     \glskeylisttok{##1}%
20079     \glslabeltok{##2}%
20080     \glsshorttok{##3}%
20081     \glslongtok{##4}%
20082     \newacronymhook
20083     \DefaultNewAcronymDef
20084   }%
20085   \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
20086 }
20087 \newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
20088 \newcommand*{\acrlinkfootnote}[3]{%
20089   \footnote{\glslink[#1]{#2}{#3}}%
20090 }

```

```

20091 \newcommand*{\acrnolinkfootnote}[3]{%
20092   \footnote{#3}%
20093 }
20094 \newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
20095   \defglsentryfmt[#1]{%
20096     \ifdefempty\glscustomtext
20097     {%
20098       \ifglsused{\glslabel}%
20099       {%
20100         \acronymfont{\glsgenentryfmt}%
20101       }%
20102     {%
20103       \firstacronymfont{\glsgenentryfmt}%
20104       \ifgls hassymbol{\glslabel}%
20105       {%
20106         \expandafter\protect\expandafter\acrfootnote\expandafter
20107         {\@gls@link@opts}{\@gls@link@label}%
20108       {%
20109         \glsifplural
20110         {\glsentrysymbolplural{\glslabel}}%
20111         {\glsentrysymbol{\glslabel}}%
20112       }%
20113     }%
20114   }%
20115 }%
20116 {\glscustomtext\glsinsert}%
20117 }%
20118 }
20119 \newcommand*{\DescriptionFootnoteNewAcronymDef}{%
20120   \edef\@do@newglossaryentry{%
20121     \noexpand\newglossaryentry{\the\glslabeltok}%
20122     {%
20123       type=\acronymtype,%
20124       name={\noexpand\acronymfont{\the\glsshorttok}},%
20125       sort={\the\glsshorttok},%
20126       first={\the\glsshorttok},%
20127       firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20128       text={\the\glsshorttok},%
20129       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20130       short={\the\glsshorttok},%
20131       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20132       long={\the\glslongtok},%
20133       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20134       symbol={\the\glslongtok},%
20135       symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20136       \the\glskeylisttok
20137     }%
20138   }%
20139   \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

20140 \let\org@glsgls@assign@plural\glsgls@assign@plural
20141 \let\org@glsgls@assign@symbolplural\glsgls@assign@symbolplural
20142 \def\glsgls@assign@firstpl##1##2{%
20143   \@glsgls@expand@field{##1}{firstpl}{##2}%
20144 }%
20145 \def\glsgls@assign@plural##1##2{%
20146   \@glsgls@expand@field{##1}{plural}{##2}%
20147 }%
20148 \def\glsgls@assign@symbolplural##1##2{%
20149   \@glsgls@expand@field{##1}{symbolplural}{##2}%
20150 }%
20151 \do@newglossaryentry
20152 \let\glsgls@assign@plural\org@glsgls@assign@plural
20153 \let\glsgls@assign@firstpl\org@glsgls@assign@firstpl
20154 \let\glsgls@assign@symbolplural\org@glsgls@assign@symbolplural
20155 }
20156 \newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
20157   \renewcommand{\newacronym}[4][]{%
20158     \ifx\@glsgls@acronymlists\@empty
20159       \def\@glo@type{\acronymtype}%
20160       \setkeys{glossentry}{##1}%
20161       \DeclareAcronymList{\@glo@type}%
20162       \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
20163     \fi
20164     \glsglskeylisttok{##1}%
20165     \glsglslabeltok{##2}%
20166     \glsglsshorttok{##3}%
20167     \glsglslongtok{##4}%
20168     \newacronymhook
20169     \DescriptionFootnoteNewAcronymDef
20170   }%
20171   \@for\@glsgls@type:=\@glsgls@acronymlists\do{%
20172     \SetDescriptionFootnoteAcronymDisplayStyle{\@glsgls@type}%
20173   }%
20174   \ifglsglsacrsmallcaps
20175     \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
20176     \renewcommand*{\acrpluralsuffix}{\glsglsupacrpluralsuffix}%
20177   \else
20178     \ifglsglsacrsmaller
20179       \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20180     \fi
20181   \fi
20182   \ifglsglsacrdua
20183     \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
20184       can’t both be set}{}%
20185   \fi
20186 }%
20187 \newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
20188   \defglsglsentryfmt[##1]{\glsglsentryfmt}%

```

```

20189 }
20190 \newcommand*{\DescriptionDUANewAcronymDef}{%
20191   \edef\@do@newglossaryentry{%
20192     \noexpand\newglossaryentry{\the\glslabeltok}%
20193     {%
20194       type=\acronymtype,%
20195       name={\the\glslongtok},%
20196       sort={\the\glslongtok},
20197       text={\the\glslongtok},%
20198       first={\the\glslongtok},%
20199       plural={\noexpand\expandonce\noexpand\@glo@longpl},%
20200       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20201       short={\the\glsshorttok},%
20202       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20203       long={\the\glslongtok},%
20204       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20205       symbol={\the\glsshorttok},%
20206       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20207       \the\glskeylisttok
20208     }%
20209   }%
20210   \let\@org@gls@assign@firstpl\gls@assign@firstpl
20211   \let\@org@gls@assign@plural\gls@assign@plural
20212   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20213   \def\gls@assign@firstpl##1##2{%
20214     \@gls@expand@field{##1}{firstpl}{##2}%
20215   }%
20216   \def\gls@assign@plural##1##2{%
20217     \@gls@expand@field{##1}{plural}{##2}%
20218   }%
20219   \def\gls@assign@symbolplural##1##2{%
20220     \@gls@expand@field{##1}{symbolplural}{##2}%
20221   }%
20222   \@do@newglossaryentry
20223   \let\gls@assign@firstpl\@org@gls@assign@firstpl
20224   \let\gls@assign@plural\@org@gls@assign@plural
20225   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20226 }
20227 \newcommand*{\SetDescriptionDUAAcronymStyle}{%
20228   \ifglsacrsmallcaps
20229     \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
20230       can’t both be set}{}%
20231   \else
20232     \ifglsacrsmaller
20233       \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
20234         can’t both be set}{}%
20235     \fi
20236   \fi
20237   \renewcommand{\newacronym}[4][\]{%

```

```

20238 \ifx\@glsacronymlists\@empty
20239 \def\@glo@type{\acronymtype}%
20240 \setkeys{glossentry}{##1}%
20241 \DeclareAcronymList{\@glo@type}%
20242 \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
20243 \fi
20244 \glskeylisttok{##1}%
20245 \glslabeltok{##2}%
20246 \glsshorttok{##3}%
20247 \glslongtok{##4}%
20248 \newacronymhook
20249 \DescriptionDUANewAcronymDef
20250 }%
20251 \@for\@gls@type:=\@glsacronymlists\do{%
20252 \SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
20253 }%
20254 }%
20255 \newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
20256 \defglsentryfmt[#1]{%
20257 \ifdefempty\glscustomtext
20258 {%
20259 \ifglsused{\glslabel}%
20260 {%
20261 \let\gls@org@insert\glsinsert
20262 \let\glsinsert\@empty
20263 \acronymfont{\gls@genentryfmt}\gls@org@insert
20264 }%
20265 {%
20266 \gls@genentryfmt
20267 \ifgls@has@symbol{\glslabel}%
20268 {%
20269 \gls@if@plural
20270 {%
20271 \def\@glo@symbol{\gls@entrysymbolplural{\glslabel}}%
20272 }%
20273 {%
20274 \def\@glo@symbol{\gls@entrysymbol{\glslabel}}%
20275 }%
20276 \space(\protect\firstacronymfont
20277 {\gls@scapscase
20278 {\@glo@symbol}
20279 {\@glo@symbol}
20280 {\mfirstucMakeUppercase{\@glo@symbol}}})%
20281 }%
20282 {}%
20283 }%
20284 }%
20285 {\glscustomtext\glsinsert}%
20286 }%

```



```

20287 }
20288 \newcommand*{\DescriptionNewAcronymDef}{%
20289   \edef\@do@newglossaryentry{%
20290     \noexpand\newglossaryentry{\the\glslabeltok}%
20291     {%
20292       type=\acronymtype,%
20293       name={\noexpand
20294         \acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
20295       sort={\the\glsshorttok},%
20296       first={\the\glslongtok},%
20297       firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20298       text={\the\glsshorttok},%
20299       plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20300       short={\the\glsshorttok},%
20301       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20302       long={\the\glslongtok},%
20303       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20304       symbol={\noexpand\@glo@text},%
20305       symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20306       \the\glskeylisttok}%
20307   }%
20308   \let\@org@gls@assign@firstpl\gls@assign@firstpl
20309   \let\@org@gls@assign@plural\gls@assign@plural
20310   \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20311   \def\gls@assign@firstpl##1##2{%
20312     \@gls@expand@field{##1}{firstpl}{##2}%
20313   }%
20314   \def\gls@assign@plural##1##2{%
20315     \@gls@expand@field{##1}{plural}{##2}%
20316   }%
20317   \def\gls@assign@symbolplural##1##2{%
20318     \@gls@expand@field{##1}{symbolplural}{##2}%
20319   }%
20320   \@do@newglossaryentry
20321   \let\gls@assign@firstpl\@org@gls@assign@firstpl
20322   \let\gls@assign@plural\@org@gls@assign@plural
20323   \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20324 }
20325 \newcommand*{\SetDescriptionAcronymStyle}{%
20326   \renewcommand{\newacronym}[4][ ]{%
20327     \ifx\@glsacronymlists\@empty
20328       \def\@glo@type{\acronymtype}%
20329       \setkeys{glossentry}{##1}%
20330       \DeclareAcronymList{\@glo@type}%
20331       \SetDescriptionAcronymDisplayStyle{\@glo@type}%
20332     \fi
20333     \glskeylisttok{##1}%
20334     \glslabeltok{##2}%
20335     \glsshorttok{##3}%

```

```

20336 \glslongtok{##4}%
20337 \newacronymhook
20338 \DescriptionNewAcronymDef
20339 }%
20340 \@for\@gls@type:=\@glsacronymlists\do{%
20341 \SetDescriptionAcronymDisplayStyle{\@gls@type}%
20342 }%
20343 \ifglsacrsmallcaps
20344 \renewcommand{\acronymfont}[1]{\textsc{##1}}
20345 \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20346 \else
20347 \ifglsacrsmaller
20348 \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20349 \fi
20350 \fi
20351 }%
20352 \newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
20353 \defglsentryfmt[#1]{%
20354 \ifdefempty\glscustomtext
20355 {%
20356 \let\gls@org@insert\glsinsert
20357 \let\glsinsert\@empty
20358 \ifglsused{\glslabel}%
20359 {%
20360 \acronymfont{\glsgenentryfmt}\gls@org@insert
20361 }%
20362 {%
20363 \firstacronymfont{\glsgenentryfmt}\gls@org@insert
20364 \ifglsashaslong{\glslabel}%
20365 {%
20366 \expandafter\protect\expandafter\acrfootnote\expandafter
20367 {\@gls@link@opts}{\@gls@link@label}%
20368 }%
20369 \glsifplural
20370 {\glsentrylongpl{\glslabel}}%
20371 {\glsentrylong{\glslabel}}%
20372 }%
20373 }%
20374 {}}%
20375 }%
20376 }%
20377 {\glscustomtext\glsinsert}%
20378 }%
20379 }
20380 \newcommand*{\FootnoteNewAcronymDef}{%
20381 \edef\@do@newglossaryentry{%
20382 \noexpand\newglossaryentry{\the\glslabeltok}%
20383 {%
20384 type=\acronymtype,%

```

```

20385     name={\noexpand\acronymfont{\the\glsshorttok}},%
20386     sort={\the\glsshorttok},%
20387     text={\the\glsshorttok},%
20388     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20389     first={\the\glsshorttok},%
20390     firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20391     short={\the\glsshorttok},%
20392     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20393     long={\the\glslongtok},%
20394     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20395     description={\the\glslongtok},%
20396     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20397     \the\glskeylisttok
20398   }%
20399 }%
20400 \let\@org@gls@assign@plural\gls@assign@plural
20401 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20402 \let\@org@gls@assign@descplural\gls@assign@descplural
20403 \def\gls@assign@firstpl##1##2{%
20404   \@gls@expand@field{##1}{firstpl}{##2}%
20405 }%
20406 \def\gls@assign@plural##1##2{%
20407   \@gls@expand@field{##1}{plural}{##2}%
20408 }%
20409 \def\gls@assign@descplural##1##2{%
20410   \@gls@expand@field{##1}{descplural}{##2}%
20411 }%
20412 \@do@newglossaryentry
20413 \let\gls@assign@plural\@org@gls@assign@plural
20414 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20415 \let\gls@assign@descplural\@org@gls@assign@descplural
20416 }
20417 \newcommand*{\SetFootnoteAcronymStyle}{%
20418   \renewcommand{\newacronym}[4][]{%
20419     \ifx\@glsacronymlists\@empty
20420       \def\@glo@type{\acronymtype}%
20421       \setkeys{glossentry}{##1}%
20422       \DeclareAcronymList{\@glo@type}%
20423       \SetFootnoteAcronymDisplayStyle{\@glo@type}%
20424     \fi
20425     \glskeylisttok{##1}%
20426     \glslabeltok{##2}%
20427     \glsshorttok{##3}%
20428     \glslongtok{##4}%
20429     \newacronymhook
20430     \FootnoteNewAcronymDef
20431   }%
20432   \@for\@gls@type:=\@glsacronymlists\do{%
20433     \SetFootnoteAcronymDisplayStyle{\@gls@type}%

```

```

20434 }%
20435 \ifglsmallcaps
20436   \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
20437   \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20438 \else
20439   \ifglsmaller
20440     \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
20441   \fi
20442 \fi
20443 \ifglacrdua
20444   \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
20445     can’t both be set}{}%
20446 \fi
20447 }%
20448 \DeclareRobustCommand*{\glsdoparenifnotempty}[2]{%
20449   \protected@edef\gls@tmp{#1}%
20450   \ifdefempty\gls@tmp
20451   {%
20452   {%
20453     \ifx\gls@tmp\@gls@default@value
20454     \else
20455       \space (#2{#1})%
20456     \fi
20457   }%
20458 }
20459 \newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
20460   \defglentryfmt[#1]{%
20461     \ifdefempty\glscustomtext
20462     {%
20463       \let\gls@org@insert\glsinsert
20464       \let\glsinsert\@empty
20465       \ifglused{\glslabel}%
20466       {%
20467         \acronymfont{\glsgenentryfmt}\gls@org@insert
20468       }%
20469     {%
20470       \glsgenentryfmt
20471       \ifglshassymbol{\glslabel}%
20472     {%
20473       \glsifplural
20474       {%
20475         \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
20476       }%
20477     {%
20478       \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
20479     }%
20480     \space
20481     (\glscapscase
20482     {\firstacronymfont{\@glo@symbol}}%

```

```

20483         {\firstacronymfont{\@glo@symbol}}}%
20484         {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})}%
20485     }%
20486     {}%
20487 }%
20488 }%
20489 {\glscustomtext\glsinsert}%
20490 }%
20491 }
20492 \newcommand*{\SmallNewAcronymDef}{%
20493 \edef\@do@newglossaryentry{%
20494 \noexpand\newglossaryentry{\the\glslabeltok}%
20495 {%
20496     type=\acronymtype,%
20497     name={\noexpand\acronymfont{\the\glsshorttok}},%
20498     sort={\the\glsshorttok},%
20499     text={\the\glsshorttok},%
20500     plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20501     first={\the\glslongtok},%
20502     firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20503     short={\the\glsshorttok},%
20504     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20505     long={\the\glslongtok},%
20506     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20507     description={\noexpand\@glo@first},%
20508     descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20509     symbol={\the\glsshorttok},%
20510     symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20511     \the\glskeylisttok
20512 }%
20513 }%
20514 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20515 \let\@org@gls@assign@plural\gls@assign@plural
20516 \let\@org@gls@assign@descplural\gls@assign@descplural
20517 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20518 \def\gls@assign@firstpl##1##2{%
20519 \@@gls@expand@field{##1}{firstpl}{##2}%
20520 }%
20521 \def\gls@assign@plural##1##2{%
20522 \@@gls@expand@field{##1}{plural}{##2}%
20523 }%
20524 \def\gls@assign@descplural##1##2{%
20525 \@@gls@expand@field{##1}{descplural}{##2}%
20526 }%
20527 \def\gls@assign@symbolplural##1##2{%
20528 \@@gls@expand@field{##1}{symbolplural}{##2}%
20529 }%
20530 \@do@newglossaryentry
20531 \let\gls@assign@firstpl\@org@gls@assign@firstpl

```

```

20532 \let\gls@assign@plural\@org@gls@assign@plural
20533 \let\gls@assign@descplural\@org@gls@assign@descplural
20534 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20535 }
20536 \newcommand*{\SetSmallAcronymStyle}{%
20537   \renewcommand{\newacronym}[4][1]{%
20538     \ifx\@glsacronymlists\@empty
20539       \def\@glo@type{\acronymtype}%
20540       \setkeys{glossentry}{##1}%
20541       \DeclareAcronymList{\@glo@type}%
20542       \SetSmallAcronymDisplayStyle{\@glo@type}%
20543     \fi
20544     \glskeylisttok{##1}%
20545     \glslabeltok{##2}%
20546     \glsshorttok{##3}%
20547     \glslongtok{##4}%
20548     \newacronymhook
20549     \SmallNewAcronymDef
20550   }%
20551   \@for\@gls@type:=\@glsacronymlists\do{%
20552     \SetSmallAcronymDisplayStyle{\@gls@type}%
20553   }%
20554   \ifglsacrsmallcaps
20555     \renewcommand*{\acronymfont}[1]{\textsc{##1}}
20556     \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
20557   \else
20558     \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}
20559   \fi
20560   \ifglsacrdua
20561     \ifglsacrsmallcaps
20562       \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
20563         can’t both be set}{}%
20564     \else
20565       \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
20566         can’t both be set}{}%
20567     \fi
20568   \fi
20569 }%
20570 \newcommand*{\SetDUADisplayStyle}[1]{%
20571   \defglsentryfmt[#1]{\glsentryfmt}%
20572 }
20573 \newcommand*{\DUANewAcronymDef}{%
20574   \edef\@do@newglossaryentry{%
20575     \noexpand\newglossaryentry{\the\glslabeltok}%
20576     {%
20577       type=\acronymtype,%
20578       name={\the\glsshorttok},%
20579       text={\the\glslongtok},%
20580       first={\the\glslongtok},%

```

```

20581 plural={\noexpand\expandonce\noexpand\@glo@longpl},%
20582 firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20583 short={\the\glsshorttok},%
20584 shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20585 long={\the\glslongtok},%
20586 longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20587 description={\the\glslongtok},%
20588 descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
20589 symbol={\the\glsshorttok},%
20590 symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
20591 \the\glskeylisttok
20592 }%
20593 }%
20594 \let\@org@gls@assign@firstpl\gls@assign@firstpl
20595 \let\@org@gls@assign@plural\gls@assign@plural
20596 \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
20597 \let\@org@gls@assign@descplural\gls@assign@descplural
20598 \def\gls@assign@firstpl##1##2{%
20599   \@gls@expand@field{##1}{firstpl}{##2}%
20600 }%
20601 \def\gls@assign@plural##1##2{%
20602   \@gls@expand@field{##1}{plural}{##2}%
20603 }%
20604 \def\gls@assign@symbolplural##1##2{%
20605   \@gls@expand@field{##1}{symbolplural}{##2}%
20606 }%
20607 \def\gls@assign@descplural##1##2{%
20608   \@gls@expand@field{##1}{descplural}{##2}%
20609 }%
20610 \do@newglossaryentry
20611 \let\gls@assign@firstpl\@org@gls@assign@firstpl
20612 \let\gls@assign@plural\@org@gls@assign@plural
20613 \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
20614 \let\gls@assign@descplural\@org@gls@assign@descplural
20615 }
20616 \newcommand*\SetDUASstyle{%
20617   \renewcommand{\newacronym}[4][\]{%
20618     \ifx\@glsacronymlists\@empty
20619       \def\@glo@type{\acronymtype}%
20620       \setkeys{glossentry}{##1}%
20621       \DeclareAcronymList{\@glo@type}%
20622       \SetDUADisplayStyle{\@glo@type}%
20623     \fi
20624     \glskeylisttok{##1}%
20625     \glslabeltok{##2}%
20626     \glsshorttok{##3}%
20627     \glslongtok{##4}%
20628     \newacronymhook
20629     \DUANewAcronymDef

```

```

20630 }%
20631 \@for\@gls@type:=\@glsacronymlists\do{%
20632   \SetDUADisplayStyle{\@gls@type}%
20633 }%
20634 }
20635 \newcommand*{\SetAcronymStyle}{%
20636   \SetDefaultAcronymStyle
20637   \ifglsacrdescription
20638     \ifglsacrfootnote
20639       \SetDescriptionFootnoteAcronymStyle
20640     \else
20641       \ifglsacrdua
20642         \SetDescriptionDUAAcronymStyle
20643       \else
20644         \SetDescriptionAcronymStyle
20645       \fi
20646     \fi
20647   \else
20648     \ifglsacrfootnote
20649       \SetFootnoteAcronymStyle
20650     \else
20651       \ifthenelse{\boolean{glsacrsmalldcaps}\OR
20652         \boolean{glsacrsmaller}}{%
20653         {%
20654           \SetSmallAcronymStyle
20655         }%
20656         {%
20657           \ifglsacrdua
20658             \SetDUASStyle
20659           \fi
20660         }%
20661       \fi
20662     \fi
20663 }
20664 \SetAcronymStyle
20665 \newcommand*{\SetCustomDisplayStyle}[1]{%
20666   \defglsentryfmt[#1]{\glsentryfmt}%
20667 }
20668 \newcommand*{\CustomAcronymFields}{%
20669   name={\the\glsshorttok},%
20670   description={\the\glslongtok},%
20671   first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
20672   firstplural={\acrfullformat
20673     {\noexpand\glsentrylongpl{\the\glslabeltok}}}%
20674     {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
20675   text={\the\glsshorttok},%
20676   plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
20677 }
20678 \newcommand*{\CustomNewAcronymDef}{%

```



```

20679 \protected@edef\do@newglossaryentry{%
20680   \noexpand\newglossaryentry{\the\glslabeltok}%
20681   {%
20682     type=\acronymtype,%
20683     short={\the\glsshorttok},%
20684     shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
20685     long={\the\glslongtok},%
20686     longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
20687     user1={\the\glsshorttok},%
20688     user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
20689     user3={\the\glslongtok},%
20690     user4={\the\glslongtok\noexpand\acrpluralsuffix},%
20691     \CustomAcronymFields,%
20692     \the\glskeylisttok
20693   }%
20694 }%
20695 \do@newglossaryentry
20696 }
20697 \newcommand*\SetCustomStyle{%
20698   \renewcommand{\newacronym}[4][\%
20699     \ifx\@glsacronymlists\empty
20700       \def\@glo@type{\acronymtype}%
20701       \setkeys{glossentry}{##1}%
20702       \DeclareAcronymList{\@glo@type}%
20703       \SetCustomDisplayStyle{\@glo@type}%
20704     \fi
20705     \glskeylisttok{##1}%
20706     \glslabeltok{##2}%
20707     \glsshorttok{##3}%
20708     \glslongtok{##4}%
20709     \newacronymhook
20710     \CustomNewAcronymDef
20711   }%
20712   \@for\@gls@type:=\@glsacronymlists\do{%
20713     \SetCustomDisplayStyle{\@gls@type}%
20714   }%
20715 }
20716 \RequirePackage{glossary-hypernav}
20717 \@gls@loadlist
20718 \@gls@loadlong
20719 \@gls@loadsuper
20720 \@gls@loadtree
20721 \ifx\@glossary@default@style\relax
20722 \else
20723   \setglossarystyle{\@glossary@default@style}
20724 \fi
20725 \newcommand*\showgloparent[1]{%
20726   \expandafter\show\csname glo@glsdetoklabel{#1}@parent\endcsname
20727 }

```

```

20728 \newcommand*{\showglolevel}[1]{%
20729   \expandafter\show\csname glo@glstetoklabel{#1}@level\endcsname
20730 }
20731 \newcommand*{\showglotext}[1]{%
20732   \expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
20733 }
20734 \newcommand*{\showgloplural}[1]{%
20735   \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
20736 }
20737 \newcommand*{\showglofirst}[1]{%
20738   \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
20739 }
20740 \newcommand*{\showglofirstpl}[1]{%
20741   \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
20742 }
20743 \newcommand*{\showglotype}[1]{%
20744   \expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
20745 }
20746 \newcommand*{\showglocounter}[1]{%
20747   \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
20748 }
20749 \newcommand*{\showglouserii}[1]{%
20750   \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
20751 }
20752 \newcommand*{\showglouseriii}[1]{%
20753   \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
20754 }
20755 \newcommand*{\showglouseriv}[1]{%
20756   \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
20757 }
20758 \newcommand*{\showglouseriv}[1]{%
20759   \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
20760 }
20761 \newcommand*{\showglouseriv}[1]{%
20762   \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
20763 }
20764 \newcommand*{\showglouseriv}[1]{%
20765   \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
20766 }
20767 \newcommand*{\showgloname}[1]{%
20768   \expandafter\show\csname glo@glstetoklabel{#1}@name\endcsname
20769 }
20770 \newcommand*{\showglodesc}[1]{%
20771   \expandafter\show\csname glo@glstetoklabel{#1}@desc\endcsname
20772 }
20773 \newcommand*{\showglodescplural}[1]{%
20774   \expandafter\show\csname glo@glstetoklabel{#1}@descplural\endcsname
20775 }
20776 \newcommand*{\showglosort}[1]{%

```

```

20777 \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
20778 }
20779 \newcommand*{\showglosymbol}[1]{%
20780 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
20781 }
20782 \newcommand*{\showglosymbolplural}[1]{%
20783 \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
20784 }
20785 \newcommand*{\showgloshort}[1]{%
20786 \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
20787 }
20788 \newcommand*{\showglolong}[1]{%
20789 \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
20790 }
20791 \newcommand*{\showgloindex}[1]{%
20792 \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
20793 }
20794 \newcommand*{\showgloflag}[1]{%
20795 \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
20796 }
20797 \newcommand*{\showgloloclist}[1]{%
20798 \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
20799 }
20800 \newcommand*{\showglofield}[2]{%
20801 \csshow{glo@\glsdetoklabel{#1}@#2}%
20802 }
20803 \newcommand*{\showacronymlists}{%
20804 \show@glsacronymlists
20805 }
20806 \newcommand*{\showglossaries}{%
20807 \show@glo@types
20808 }
20809 \newcommand*{\showglossaryin}[1]{%
20810 \expandafter\show\csname @glotype@#1@in\endcsname
20811 }
20812 \newcommand*{\showglossaryout}[1]{%
20813 \expandafter\show\csname @glotype@#1@out\endcsname
20814 }
20815 \newcommand*{\showglossarytitle}[1]{%
20816 \expandafter\show\csname @glotype@#1@title\endcsname
20817 }
20818 \newcommand*{\showglossarycounter}[1]{%
20819 \expandafter\show\csname @glotype@#1@counter\endcsname
20820 }
20821 \newcommand*{\showglossaryentries}[1]{%
20822 \expandafter\show\csname glolist@#1\endcsname
20823 }
20824 \csname ifglcompatible-2.07\endcsname
20825 \RequirePackage{glossaries-compatible-207}

```

20826 \fi

6.3 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
20827 \ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
20828 \providecommand{\indexspace}{%
20829   \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
20830 }
20831 \newcommand*{\glslistgroupheaderfmt}[1]{#1}
20832 \newcommand*{\glslistnavigationitem}[1]{\item[#1]}
20833 \newglossarystyle{list}{%
20834   \renewenvironment{theglossary}%
20835     {\begin{description}}{\end{description}}%
20836   \renewcommand*{\glossaryheader}{}%
20837   \renewcommand*{\glsgroupheading}[1]{}%
20838   \renewcommand*{\glossentry}[2]{%
20839     \item[\glsentryitem{##1}%
20840       \glstarget{##1}{\glossentryname{##1}}]
20841     \glossentrydesc{##1}\glspostdescription\space ##2}%
20842   \renewcommand*{\subglossentry}[3]{%
20843     \glssubentryitem{##2}%
20844     \glstarget{##2}{\strut}\space
20845     \glossentrydesc{##2}\glspostdescription\space ##3.}%
20846   \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
20847 }
20848 \newglossarystyle{listgroup}{%
20849   \setglossarystyle{list}%
20850   \renewcommand*{\glsgroupheading}[1]{%
20851     \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
20852 \newglossarystyle{listhypergroup}{%
20853   \setglossarystyle{list}%
20854   \renewcommand*{\glossaryheader}{%
20855     \glslistnavigationitem{\glsnavigation}}%
20856   \renewcommand*{\glsgroupheading}[1]{%
20857     \item[\glslistgroupheaderfmt
20858       {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}
20859 \newglossarystyle{altlist}{%
20860   \setglossarystyle{list}%
20861   \renewcommand*{\glossentry}[2]{%
20862     \item[\glsentryitem{##1}%
20863       \glstarget{##1}{\glossentryname{##1}}]
20864     \mbox{} \par \nobreak \afterheading
20865     \glossentrydesc{##1}\glspostdescription\space ##2}%
20866   \renewcommand*{\subglossentry}[3]{%
20867     \par
20868     \glssubentryitem{##2}%
20869     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%

```

```

20870 }
20871 \newglossarystyle{altlistgroup}{%
20872   \setglossarystyle{altlist}%
20873   \renewcommand*{\glsgroupheading}[1]{%
20874     \item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
20875 \newglossarystyle{altlisthypergroup}{%
20876   \setglossarystyle{altlist}%
20877   \renewcommand*{\glossaryheader}{%
20878     \glslistnavigationitem{\glsnavigation}}%
20879   \renewcommand*{\glsgroupheading}[1]{%
20880     \item[\glslistgroupheaderfmt
20881       {\glsnavhypertarget{##1}{\glsgrouptitle{##1}}]}
20882 \newglossarystyle{listdotted}{%
20883   \setglossarystyle{list}%
20884   \renewcommand*{\glossentry}[2]{%
20885     \item[]\makebox[\glslistdottedwidth][l]{%
20886       \glstryitem{##1}%
20887       \glstarget{##1}{\glossentryname{##1}}%
20888       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
20889   \renewcommand*{\subglossentry}[3]{%
20890     \item[]\makebox[\glslistdottedwidth][l]{%
20891       \glssubentryitem{##2}%
20892       \glstarget{##2}{\glossentryname{##2}}%
20893       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
20894 }
20895 \newlength\glslistdottedwidth
20896 \setlength{\glslistdottedwidth}{.5\hsize}
20897 \newglossarystyle{sublistdotted}{%
20898   \setglossarystyle{listdotted}%
20899   \renewcommand*{\glossentry}[2]{%
20900     \item[\glstryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}
20901 }

```

Glossary

`makeindex` An indexing application [10](#), [14](#), [30](#), [33](#), [190](#), [200](#)

`xindy` An flexible indexing application with multilingual support written in Perl [10](#), [14](#), [30](#), [33](#), [190](#), [200](#)

Change History

1.01 (2007-05-17)	numberline: numberline option added . . . 8
General: Added range facility in format key 121	1.12 (2008-03-08)
\writeist: Added spaces after \delimN and \delimR in ist file 170	\@GLSpl: now uses
1.04 (2007-08-03)	\glentrydescplural and
General: Added \glstextformat 106	\glentrysymbolplural instead of
1.05 (2007-08-10)	\glentrydesc and
\glossarysection: added \@mkboth to \glossarysection 47	\glentrysymbol 136
\gls@defglossaryentry: Changed the default value of the sort key to just the value of the name key 89	\@Glspl@: now uses
1.07 (2007-09-13)	\glentrydescplural and
\@gls@link: fixed bug caused by \theglentrycounter setting the page number too soon 119	\glentrysymbolplural instead of
\glsadd: fixed bug caused by \theglentrycounter setting the page number too soon 166	\glentrydesc and
1.08 (2007-10-13)	\glentrysymbol 134
General: Added babel support 41	General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget) 130
listgroup: changed listgroup style to use \glsgetgrouptitle 288	descriptionplural: new 70
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle 289	\gls@defglossaryentry: Changed default first plural to be first key with s appended (was text key with s appended) 89
1.1 (2008-02-22)	descriptionplural support added 89
\@glossarysection: numbered sections and auto label added 48	symbolplural support added 89
\@gls@tmpb: changed \toksdef to \newtoks 123	\Glsentrydescplural: New 159
\@gls@toc: numberline added 49	\glentrydescplural: New 159
\@p@glossarysection: numbered sections and auto label added 48	\Glsentrysymbolplural: New 160
General: amsgen now loaded (\new@ifnextchar needed) 4	\glentrysymbolplural: New 160
translate: translate option added 28	\SetDescriptionFootnoteAcronymStyle: Added \protect before \footnote and \glslink 252
\setglossarysection: new 48	\SetFootnoteAcronymStyle: Added \protect before \footnote and \glslink 258
numberedsection: numberedsection package option added 9	symbolplural: new 71

1.13 (2008-05-10)	
General: fixed bug that ignored 3rd parameter	137-144
\ACRfullpl: new	232
\Acrfullpl: new	231
\acrfullpl: new	231
\acrpluralsuffix: New	229
\gls@defglossaryentry: Changed default first value	89
Changed default firstplural value	89
Removed restriction on only using \newglossaryentry in the preamble	94
\newacronym: Removed restriction on only using \newacronym in the preamble	229
1.14 (2008-06-17)	
\@gls@hypergroup: new	281
General: added nonumberlist key to \printglossary	216
added numberedsection key to \printglossary	215
\firstacronymfont: new	232
\glsautoprefix: new	9
\glsnavhyperlink: changed \edef to \protected@edef	280
\glsnavhypertarget: added write to aux file	280
\glsnavigation: changed to only use labels for groups that are present ..	282
1.15 (2008-08-15)	
\@gls@link: added \glslabel	119
\gls@defglossaryentry: check for \@glo@first in description	93
check for \@glo@text in symbol	93
\gls@hypergroup: new	281
\glsnavhypertarget: added check if rerun required	280
\glssettoctitle: new	40
\printglossary: changed the way the TOC title is set	200
1.16 (2008-08-27)	
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	133
\@GLSpl: Test glossary type is \acronymtype in addition to checking if footnote option has been used	136
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	132
\@GLspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	135
\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	132
\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used	136
\@glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	134
\@glstarget: raised the hypertarget so the target text doesn't scroll off the top of the page	130
\gls@defglossaryentry: Changed def to let	89
1.17 (2008-12-26)	
\@do@esc@wrglossary: new	194
\@do@seeglossary: new	197
\@glo@storeentry: new	95
\@gls@glossary: changed definition to use \index instead of \@index	190
\@glsdefaultplural: new	75
\@glsdefaultsort: new	75
\@glshypernumber: new	226
\@glsnoname: new	74
\@glsnonextpages: new	216
General: added xindy support	30
parent: new	72
see: new	72
\gls@defglossaryentry: added nonumberlist key	89
added parent key	89
added see key	89
Stored main part of entry format when entry is defined	94
\gls@suffixF: new	45
\gls@suffixFF: new	45
\gls@wrglossary: modified to allow for xindy support	191

\glshyperlink: new	166	\SetDescriptionFootnoteAcronymStyle:	
\glshypernumber: modified to allow		changed \acronymfont to use	
material to be attached to location .	226	\textsmaller instead of \smaller	252
\glshnavhyperlink: replaced		\SetFootnoteAcronymStyle: changed	
\hyperlink to \@glslink	280	\acronymfont to use \textsmaller	
\glshnavhypertarget: replaced		instead of \smaller	258
\hypertarget to \@glstarget ...	280	\SetSmallAcronymStyle: changed	
\glsssee: new	198	\acronymfont to use \textsmaller	
\glssseeformat: new	198	instead of \smaller	261
\glssSetSuffixF: new	45	1.46 (2021-09-20)	
\glssSetSuffixFF: new	45	\@printglossary: changed \xdef to	
\ifglsxindy: new	30	\protected\xdef	203
\istfilename: added xindy support ...	44	General: changed \edef to	
\newglossarystyle: made		\protected@edef	396
\newglossarystyle long	225	\glss@defglossaryentry: changed	
\nopostdesc: new	43	\xdef to \protected\xdef	91, 94
nonumberlist: new	72	\glssaddprotectedpagefmt: changed	
\printglossary: added check to		\eappto to \protected@eappto ..	192
determine if \printglossary is		2.01 (2009 May 30)	
already defined	200	\@glss@link: moved \@do@wrglossary	
added print language to aux file	200	before term is displayed to prevent	
order: order package option added	30	unwanted whatsit	120
\writeist: added xindy support	170	\forallglossaries: replaced	
1.18 (2009-01-14)		\ifthenelse with \ifx	59
\@glss@loadlist: new	11	\forallglsentries: replaced \ifthenelse	
\@glss@loadlong: new	10	with \ifx	59
\@glss@loadsuper: new	11	\glssdefmain: new	17
\@glss@loadtree: new	11	\glssdescwidth: changed \linewidth to	
\glss@defglossaryentry: Changed		\hsize	290, 312
default value of sort to		\glsslistdottedwidth: changed	
\@glss@defaultsort	89	\linewidth to \hsize	290
moved sort sanitization to		\glsspagelistwidth: changed	
\newglossaryentry	93	\linewidth to \hsize	291, 313
\glstarget: new	219	nomain: added nomain package option .	17
\oldacronym: new	228	\writeist: removed item_02 - no such	
nolist: new	11	makeindex key	174
nolong: new	11	2.02 (2007-07-13)	
sort: moved sanitization to		\@printglossary: suppressed warning	
\newglossaryentry	70	globally rather than locally	203
nostyles: new	11	2.02 (2009-07-13)	
nosuper: new	11	\glossarysection: changed \@mkboth	
notree: new	11	to \glossarymark	47
1.19 (2009-03-02)		\glsglossarymark: New	47
\glsclearpage: new	49	2.03 (2009-09-23)	
\glssdisp: new	136	\@GLS@: Added check for hyperfirst	133
\SetDescriptionAcronymStyle:		\@GLSpl: Added check for hyperfirst ...	136
changed \acronymfont to use		\@Gls@: Added check for hyperfirst	132
\textsmaller instead of \smaller	256	\@Glspl@: Added check for hyperfirst ..	135
		\@Gls@: Added check for hyperfirst	132

\@gls@link: new	118	\glsentryuseriv: new	163
\@gls@link: added \leavevmode	119	\Glsentryuseriv: new	163
Moved entry existence check to avoid		\glsentryuseriv: new	163
duplicate code	119	\Glsentryuseriv: new	163
\@glsdisp: Added check for hyperfirst .	136	\glsentryuseriv: new	163
\@glspl0: Added check for hyperfirst ..	134	\ns@newglossary: added check to	
\gls glossarymark: Added check to see		determine if \@gls@<type>@display	
if it's already defined	47	and \@gls@<type>@displayfirst	
hyperfirst: new	29	have been defined.	67
2.04 (2009-11-10)		\SetAcronymLists: new	20
\@GLS@: Changed test to check if glossary		\SetDefaultAcronymDisplayStyle:	
type has been identified as a list of		new	247
acronyms	133	\SetDefaultAcronymStyle: new	249
\@GLSp1@: Changed test to check if		\SetDescriptionAcronymDisplayStyle:	
glossary type has been identified as a		new	254
list of acronyms	136	\SetDescriptionDUAAcronymDisplayStyle:	
\@GLS@: Changed test to check if glossary		new	252
type has been identified as a list of		\SetDescriptionFootnoteAcronymDisplayStyle:	
acronyms	132	new	250
\@GLSp10: Changed test to check if		\SetDUADisplayStyle: new	262
glossary type has been identified as a		\SetFootnoteAcronymDisplayStyle:	
list of acronyms	135	new	256
\@glossaryentryfield: new	95	\SetSmallAcronymDisplayStyle: new	259
\@glossarysubentryfield: new	95	2.05 (2010-02-06)	
\@gls@: Changed test to check if glossary		\@glsdisp: Added closing brace. Patch	
type has been identified as a list of		provided by Sergiu Dotenco	136
acronyms	132	Removed spurious brace. Patch	
\@glsacronymlists: new	19	provided by Sergiu Dotenco	137
\@glsdisp: Changed test to check if		\writeist: Added \string before	
glossary type has been identified as a		opening and closing braces. Patch	
list of acronyms	136	provided by Segiu Dotenco	175
\@glspl0: Changed test to check if		2.06 (2010-06-14)	
glossary type has been identified as a		\altnewglossary: new	68
list of acronyms	134	\CustomAcronymFields: new	264
\@newglossaryentryposthook: new ..	95	\CustomNewAcronymDef: new	265
\@newglossaryentryprehook: new ...	95	\SetCustomDisplayStyle: new	264
acronymlists: new	20	\SetCustomStyle: new	265
\DeclareAcronymList: new	19	2.07 (2010-07-10)	
\DefineAcronymSynonyms: new	245	General: glsadd format key stored in	
\@gls@defglossaryentry: added user1-6		\@glsnumberformat (was mistakenly	
keys	90	stored in \@glo@format)	166
\glsadd: fixed bug that ignored counter	166	3.0 (2010-07-12)	
\Glsentryuseri: new	162	\@makeglossary: Added check for	
\glsentryuseri: new	162	savewrites	179
\Glsentryuserii: new	162	\gls@wrglossary: modified to take into	
\glsentryuserii: new	162	account savewrites	191
\Glsentryuseriii: new	162	3.0 (2010/03/31)	
\glsentryuseriii: new	162	\@set@glo@numformat: added 4th	
\Glsentryuseriv: new	163	argument	121

3.0 (2011-04-02)

\@do@esc@wrglossary: added check for hyper location prefix	196	\gls@codepage: replaced \ifundefined with \ifcsundef ..	31
modified to use new format	194	\gls@defglossaryentry: added \gls@defsort	93
\@glossarysec: replaced \ifundefined with \ifcsundef ...	8	added short and long keys	90
\@do@seeglossary: Sanitize and escape cross-referencing information	197	replaced \@ifundefined with \ifcsundef	90
\@gls@counterwithin: new	13	\gls@doclearpage: replaced \ifundefined with \ifcsundef ..	49
\@gls@ifinlist: new	50	\glsadd: added \@gls@saveentrycounter	167
\@gls@link: added \@gls@saveentrycounter	120	\GlsAddXdyCounters: new	50
added \@gls@setsort	120	\glsentrycounterlabel: new	218
\@gls@saveentrycounter: new	120	\glsentryitem: new	218
\@gls@setupsort@def: new	15	\Glsentrylong: new	164
\@gls@setupsort@standard: new	14	\glsentrylong: new	164
\@gls@setupsort@use: new	15	\Glsentrylongpl: new	164
\@gls@xdy@locationlist: new	53	\glsentrylongpl: new	164
\@glslink: replaced \@ifundefined with \ifcsundef	130	\Glsentryshort: new	163
\@glsnextpages: new	216	\glsentryshort: new	163
\@print@glossary: replaced \ifundefined with \ifcsundef .	204	\Glsentryshortpl: new	164
\@printglossary: added \currentglossary	202	\glsentryshortpl: new	163
added \glsnextpages	202	\glsgetgrouptitle: replaced \ifundefined with \ifcsundef .	222
make toctitle default to title	202	\gls glossarymark: replaced \ifundefined with \ifcsundef ..	47
\@xdy@attributelist: new	50	\glshyperlink: changed default from \glsentryname to \glsentrytext	166
General: added prefix to hyperlink	227	\glshypernumber: replaced \ifundefined with \ifcsundef .	226
etoolbox now loaded	4	\glsnumberformat: replaced \ifundefined with \ifcsundef ..	45
replaced \@ifundefined with \ifcsundef	39, 41, 116, 214	\glsrefentry: new	218
\acrfootnote: new	249	\glsresetsubentrycounter: new ...	217
\ACRfull: added starred version	231	\glsseeitem: hyperlink uses \glsseeitemformat instead of	
\Acrfull: added starred version	230	\glsentryname	199
\acrfull: added starred version	229	\glsseeitemformat: new	199
\ACRfullpl: added starred version ...	232	\glssortnumberfmt: new	15
\Acrfullpl: added starred version ...	231	\glsstepentry: new	217
\acrfullpl: added starred version ...	231	\glsstepsubentry: new	217
\acrlinkfootnote: new	249	\glssubentrycounterlabel: new ...	218
\acrnolinkfootnote: new	249	\glssubentryitem: new	218
savewrites: new	34	theglossary: replaced \@ifundefined with \ifcsundef	218
see: added \@glo@seeautonumberlist	72	short: new	74
seeautonumberlist: new	10	shortplural: new	74
\glossarysection: replaced \ifundefined with \ifcsundef ..	47	\ifglossaryexists: replaced \ifundefined with \ifcsundef ..	60
\glossarystyle: replaced \ifundefined with \ifcsundef .	224		

<code>\ifglentryexists</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	60
<code>\istfile</code> : deprecated	189
<code>glossaryentry</code> : new	12
<code>glossarysubentry</code> : new	13
<code>\newglossaryentry</code> : replaced	
<code>\DeclareRobustCommand</code> with	
<code>\newrobustcmd</code>	77
<code>\newglossarystyle</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	225
<code>\ns@newglossary</code> : added	
<code>\@gls@defsortcount</code>	68
replaced <code>\@ifundefined</code> with	
<code>\ifcsundef</code>	67
<code>entrycounter</code> : new	13
<code>\oldacronym</code> : replaced <code>\@ifundefined</code>	
with <code>\ifcsundef</code>	228
compatible-2.07: compatible-2.07	
option added	34
<code>long</code> : new	74
<code>longplural</code> : new	74
<code>nonumberlist</code> : now boolean	72
<code>sort</code> : new	14
<code>counter</code> : replaced <code>\@ifundefined</code> with	
<code>\ifcsundef</code>	71
<code>counterwithin</code> : new	13
<code>\printglossary</code> : replaced	
<code>\@ifundefined</code> with <code>\ifcsundef</code> ..	200
<code>\SetDescriptionFootnoteAcronymDisplayStyle</code>	
expanded options link options	250
<code>\setentrycounter</code> : added optional	
argument	223
<code>\showacronymlists</code> : new	270
<code>\showglocounter</code> : new	267
<code>\showglodesc</code> : new	269
<code>\showglodescplural</code> : new	269
<code>\showglofirst</code> : new	267
<code>\showglofirstpl</code> : new	267
<code>\showgloflag</code> : new	270
<code>\showgloindex</code> : new	270
<code>\showglolevel</code> : new	266
<code>\showglolevel</code> : new	268
<code>\showgloparent</code> : new	266
<code>\showgloplural</code> : new	267
<code>\showglosort</code> : new	269
<code>\showglossaries</code> : new	270
<code>\showglossarycounter</code> : new	271
<code>\showglossaryentries</code> : new	271
<code>\showglossaryin</code> : new	271
<code>\showglossaryout</code> : new	271
<code>\showglossarytitle</code> : new	271
<code>\showglosymbol</code> : new	269
<code>\showglosymbolplural</code> : new	269
<code>\showglotext</code> : new	266
<code>\showglotype</code> : new	267
<code>\showglouserii</code> : new	267
<code>\showglouseriii</code> : new	268
<code>\showglouseriv</code> : new	268
<code>\showglouserv</code> : new	268
<code>\showglouservi</code> : new	268
<code>subentrycounter</code> : new	13
<code>\writeist</code> : added xindy-only macro	
definitions to glossary open tag	172
modified to support new format	170
3.01 (2011-04-12)	
<code>\@glswritefiles</code> : added check for	
empty glossaries	189
General: made robust	133
<code>\ACRfull</code> : made robust	231
<code>\Acrfull</code> : made robust	230
<code>\acrfull</code> : made robust	229
<code>\acrfullformat</code> : removed	
<code>\acronymfont</code> as it should already be	
set in the second argument.	230
<code>\ACRfullpl</code> : made robust	232
<code>\Acrfullpl</code> : made robust	231
<code>\acrfullpl</code> : made robust	231
<code>\ACRlong</code> : made robust	154
<code>\Acrlong</code> : made robust	153
<code>\acrlong</code> : made robust	153
<code>\ACRlongpl</code> : made robust	156
<code>\Acrlongpl</code> : made robust	155
<code>\acrlongpl</code> : made robust	155
<code>\ACRshort</code> : made robust	150
<code>\Acrshort</code> : made robust	150
<code>\acrshort</code> : made robust	149
<code>\ACRshortpl</code> : made robust	152
<code>\Acrshortpl</code> : made robust	152
<code>\acrshortpl</code> : made robust	151
<code>\Gls</code> : made robust	132
<code>\glsadd</code> : made robust	166
<code>\glsaddall</code> : made robust	167
<code>\GLSdesc</code> : made robust	142
<code>\Glsdesc</code> : made robust	141
<code>\glsdesc</code> : made robust	141
<code>\GLSdescplural</code> : made robust	143
<code>\Glsdescplural</code> : made robust	142

\glsdescplural: made robust	142	\do@wr@glossary code into	
\glsfirst: made robust	138	\@do@wrglossary	191
\GLSfirstplural: made robust	140	\@gls@missingnumberlist: new	75
\Glsfirstplural: made robust	140	\@gls@writefiles: added check for	
\glsfirstplural: made robust	140	existence of token in case	
\glslink: made robust	118	\makeglossaries has been	
\GLSname: made robust	141	omitted	189
\Glsname: made robust	141	\@printglossary: add a way to fetch	
\glsname: made robust	140	current entry label	203
\GLSpl: made robust	135	savenumberlist: new	10
\Glspl: made robust	134	ucmark: new	12
\glspl: made robust	134	\gls@defglossaryentry: added	
\GLSplural: made robust	139	numberlist element	93
\GLSsymbol: made robust	143	\gls@save@numberlist: new	199
\Glsymbol: made robust	143	\gls@wrglossary: added check for	
\glssymbol: made robust	143	glossary file defined	191
\GLSsymbolplural: made robust	144	\glsdisplaynumberlist: new	165
\Glsymbolplural: made robust	144	\glstentrycounter: set default value ..	120
\glssymbolplural: made robust	144	\Glsentryfull: fixed bug (replaced	
\Glstext: made robust	138	\glstentryshortpl with	
\glstext: made robust	137	\glstentryshort)	164
\GLSuseri: made robust	145	\glstentryfullpl: fixed bug (replaced	
\Glsuseri: made robust	145	\glstentryshort with	
\glsuseri: made robust	145	\glstentryshortpl)	164
\GLSuserii: made robust	146	\glstentrynumberlist: new	165
\Glsuserii: made robust	146	\glsmoveentry: new	95
\glsuserii: made robust	145	\glsresetsubentrycounter: new ...	217
\GLSuseriii: made robust	147	\ifglshaschildren: new	62
\Glsuseriii: made robust	146	\ifglshasparent: new	62
\glsuseriii: made robust	146	\makeglossaries: added list parser ..	184
\GLSuseriv: made robust	147	indexonlyfirst: new	29
\Glsuseriv: made robust	147	\renewglossarystyle: new	225
\glsuseriv: made robust	147	\showglossaryentries: fixed misspelt	
\GLSuserv: made robust	148	command	271
\Glsuserv: made robust	148	\SmallNewAcronymDef: fixed broken	
\glsuserv: made robust	148	short and long plural	260
\GLSuservi: made robust	149	3.03 (2012/09/21)	
\Glsuservi: made robust	149	\@gls@sanitizesort: new	23
\glsuservi: made robust	149	\@gls@setupsort@standard: used	
3.02 (2012-05-19)		\@gls@sanitizesort	14
\glsnumlistlastsep: new	166	\@printglossary: allow title to override	
\glsnumlistsep: new	166	default toctitle	202
3.02 (2012-05-21)		General: allow title to set toctitle	214
\@do@@wrglossary: changed		\glsinlinedescformat: new	284
\@glslocref to		\glsinlineemptydescformat: new ..	284
\theglstentrycounter	196	\glsinlinenameformat: new	284
\@do@wrglossary: changed		\glsinlinepostchild: new	284
\@do@wr@glossary to test for		\glsinlinesubdescformat: new	284
indexonlyfirst option; put old		\glsinlinesubnameformat: new	284

\glspostinline: replaced “.” with	unwanted whatsit occurring here. . .	204
\glspostdescription	General: Added check for doc package . . .	4
list: added check for glsnogroupskip .	added datatool-base as a required	
altlongragged4col: added check for	package	4
glsnogroupskip	added local key	116
altsuperragged4col: added check for	\gls@Alphpage: new	192
glsnogroupskip	\gls@alphpage: new	192
alttree: added check for	\gls@disablepagerefexpansion: new	192
glsnogroupskip	\gls@numberpage: new	192
index: added check for glsnogroupskip	\gls@protected@pagefmts: new . . .	192
327	\gls@romanpage: new	192
nogroupskip: new	\glsdefmain: added check for doc	
12	package	17
long: added check for glsnogroupskip .	\glsorg@endtheglossary: new	5
291	\glsorg@theglossary: new	5
long3col: added check for	\PrintChanges: new	5
glsnogroupskip	3.05 (2013-04-21)	
293	\@@do@esc@wrglossary: add Roman	
long4col: added check for	case. Fixed bugs in the else	
glsnogroupskip	statements	195
294	\@gls@link: added check for	
longragged: added check for	“nohypertypes”	119
glsnogroupskip	mcolalttree: replaced ‘2’ with	
302	\glsmcols	311
longragged3col: added check for	mcolindex: replaced ‘2’ with \glsmcols	307
glsnogroupskip	mcolindexspannav: replaced ‘2’ with	
304	\glsmcols	308
no postdot: new	mcoltree: replaced ‘2’ with \glsmcols	308
12	mcoltreename: replaced ‘2’ with	
tree: added check for glsnogroupskip .	\glsmcols	310
328	mcoltreespannav: replaced ‘2’ with	
treename: added check for	\glsmcols	309
glsnogroupskip	\gls@protected@pagefmts: added	
330	Roman to list	192
super: added check for glsnogroupskip	\gls@Romanpage: new	192
313	\glsgetgrouplabel: fixed bug (typo in	
super3col: added check for	\equal)	223
glsnogroupskip	\nopostdesc: made robust	43
315	3.05 (2013/04/21)	
super4col: added check for	\@gls@nohyperlist: new	20
glsnogroupskip	\GlsDeclareNoHyperList: new	20
317	nohypertypes: new	20
superragged: added check for	3.06 (2013/06/17)	
glsnogroupskip	\@xdy@main@language: Changed back to	
320	using \language	30
superragged3col: added check for	\findrootlanguage: Obsoleted	57
glsnogroupskip	3.07 (2013-07-05)	
322	\@gls@link: fixed bug that failed to find	
3.04 (2012-11-11)	entry in list	119
altlist: replaced \newline with		
paragraph break		
289		
3.04 (2012-11-18)		
\@@do@@wrglossary: changed		
\theglsentrycounter back to		
\@glslocref		196
\@@do@esc@wrglossary: modified to		
compensate for possible incorrect		
page number		195
\@gls@escbsdq: unsanitize		
\gls@numberpage, \gls@alphpage,		
\gls@Alphpage and		
\gls@romanpage		122
\@print@glossary: Moved aux write to		
end of document to prevent		

\glossarypreamble: modified to work with \setglossarypreamble	46	\glsaddallunused: new	167
\gls@docclearpage: added check for openright	49	\glsexpandfields: new	77
\glspostdescription: Added spacefactor code	12	\glsnoexpandfields: new	77
\GlsSetXdyCodePage: Added check for fontspec	58	\glssee: made robust	198
\SetDescriptionAcronymDisplayStyle: now using \glsdoparenifnotempty	254	\glsseeformat: made robust	198
\setglossarypreamble: new	46	\glsseeitem: made robust	199
3.08a (2013-08-30)		\glsseelist: made robust	198
list: updated list style to use \glossentry and \subglossentry	288	\ifglsdescsuppressed: new	63
listdotted: updated listdotted style to use \glossentry and \subglossentry	290	\ifglshasdesc: new	63
altlist: updated altlist style to use \glossentry and \subglossentry	289	\ifglshassymbol: new	63
inline: updated inline style to use \glossentry and \subglossentry	283	altlongragged4col: updated to use \glossentry and \subglossentry	305
3.08a (2013-09-28)		alttree: updated to use \glossentry and \subglossentry	332
\@glo@storeentry: no longer need to check for special characters in any of the fields other than sort	96	index: added paragraph break at end of environment	326
updated for \glossentry	96	updated to use \glossentry and \subglossentry	326
\@glossaryentryfield: switched to \glossentry	95	long: updated to use \glossentry and \subglossentry	291
\@glossarysubentryfield: switched to \subglossentry	95	longragged: updated to use \glossentry and \subglossentry	302
General: added nogroupskip key to \printglossary	215	longragged3col: updated to use \glossentry and \subglossentry	303
removed definition of \@glossaryentryfield	386	tree: updated to use \glossentry and \subglossentry	328
removed definition of \@glossarysubentryfield	386	\setglossarystyle: new	224
\compatibleglossentry: new	219	\setglossentrycompatibility: new	221
\compatiblesubglossentry: new ...	220	superragged: updated to use \glossentry and \subglossentry	320
\glossaryentryfield: deprecated ...	221	3.09a (2013-10-09)	
\Glossentrydesc: new	220	\@gls@assign@symbolplural@field: new	23
\glossentrydesc: new	220	\@gls@default@value: new	71
\Glossentryname: new	220	\Glsentrydesc: made robust	159
\glossentryname: new	219	\Glsentrydescplural: made robust ..	159
\Glossentrysymbol: new	220	\Glsentryfirst: made robust	161
\glossentrysymbol: new	220	\Glsentryfirstplural: made robust .	161
\gls@assign@desc@field: new	22	\Glsentryfull: made robust	164
\gls@assign@descplural@field: new	22	\Glsentryfullpl: made robust	164
\gls@assign@field: new	77	\Glsentrylong: made robust	164
\gls@ifnotmeasuring: new	97	\Glsentrylongpl: made robust	164
		\Glsentryname: made robust	158
		\Glsentryplural: made robust	160
		\Glsentryshort: made robust	163
		\Glsentryshortpl: made robust	164
		\Glsentrysymbol: made robust	160
		\Glsentrysymbolplural: made robust	160
		\Glsentrytext: made robust	160

\Glsentryuseri: made robust	162	\@GLSpl: add \glslabel,	
\Glsentryuserii: made robust	162	\glsifplural, \glscapscase,	
\Glsentryuseriii: made robust	162	\glscustomtext and \glsinsert	135
\Glsentryuseriv: made robust	163	change to using \glsentryfmt style	
\Glsentryuserv: made robust	163	commands	136
\Glsentryuservi: made robust	163	removed \MakeUppercase as now	
\glstextup: new	229	dealt with in \glsentryfmt	136
\ifglshassymbol: changed test to check		\@Gls@: add \glsifplural,	
for \@gls@default@symbol	63	\glscapscase, \glscustomtext and	
3.10a (2013-09-28)		\glsinsert	132
\gls@assign@type@field: new	22	change to using \glsentryfmt style	
3.10a (2013-10-13)		commands	132
\@gls@keymap: new	80	removed \makefirstuc (now dealt	
\@gls@provide@newglossary: new . . .	66	with in \glsentryfmt)	132
\@gls@writedef: new	80	\@Glspl@: add \glsifplural,	
\@glsdefaultplural: Obsolete	75	\glscapscase, \glscustomtext and	
\@glsnodesc: new	75	\glsinsert	135
\@print@glossary: Added		change to using \glsentryfmt style	
providecommand code to aux file . .	204	commands	135
\gls@defglossaryentry: Changed to		removed \makefirstuc (now dealt	
using \@gls@default@value	89	with in \glsentryfmt)	135
new	89	\@acrlong: added \glslabel,	
\glswritedefhook: new	88	\glsifplural, \glscapscase,	
\makeglossaries: Added		\glsinsert and \glscustomtext	379
providecommand code to aux file . .	183	\@acrshort: added \glslabel,	
\new@glossaryentry: new	78	\glsifplural, \glscapscase,	
\ns@newglossary: added		\glsinsert and \glscustomtext	378
\@gls@provide@newglossary	67	\@gls@: add \glslabel, \glsifplural,	
3.11a (2013-10-15)		\glscapscase, \glscustomtext and	
\@ACRlong: added \glslabel,		\glsinsert	131
\glsifplural, \glscapscase,		change to using \glsentryfmt style	
\glsinsert and \glscustomtext	380	commands	132
\@ACRshort: added \glslabel,		\@gls@noexpand@fields: Fixed bug	
\glsifplural, \glscapscase,		expand replaced with noexpand	76
\glsinsert and \glscustomtext	379	\@glsdisp: add \glslabel,	
\@Acrlong: added \glslabel,		\glsifplural, \glscapscase,	
\glsifplural, \glscapscase,		\glscustomtext and \glsinsert	136
\glsinsert and \glscustomtext	380	change to using \glsentryfmt style	
\@Acrshort: added \glslabel,		commands	136
\glsifplural, \glscapscase,		\@glspl@: add \glslabel,	
\glsinsert and \glscustomtext	378	\glsifplural, \glscapscase,	
\@GLS@: add \glslabel, \glsifplural,		\glscustomtext and \glsinsert	134
\glscapscase, \glscustomtext and		change to using \glsentryfmt style	
\glsinsert	133	commands	134
change to using \glsentryfmt style		General: added \glslabel,	
commands	133	\glsifplural, \glscapscase,	
removed \MakeUppercase (now		\glsinsert and	
moved to \glsentryfmt)	133	\glscustomtext	150–156

changed to just use		<code>\defglentryfmt:new</code>	66
<code>\Glsentrydescplural</code>	142	<code>\forglentries:replaced \ifx with</code>	
changed to just use		<code>\ifdefempty</code>	59
<code>\glentrydescplural</code>	142, 143	<code>\gls@assign@desc:new</code>	88
changed to just use <code>\Glsentrydesc</code> .	142	<code>\gls@defglossaryentry: Fixed default</code>	
changed to just use		counter if none supplied	92
<code>\glentrydesc</code>	141, 142	<code>\gls@doentryfmt:new</code>	66
changed to just use		<code>\glsdisplay:obsoleted</code>	115
<code>\Glsentryfirstplural</code>	140	<code>\glsdisplayfirst:obsoleted</code>	115
changed to just use		<code>\glsgenentryfmt:new</code>	110
<code>\glentryfirstplural</code>	140	<code>\glsgetgroupitle: Added check in</code>	
changed to just use <code>\Glsentryfirst</code>	138	case non-Latin alphabet in use	222
changed to just use		<code>\glsglossarymark:replaced</code>	
<code>\glentryfirst</code>	138, 139	<code>\MakeUppercase with</code>	
changed to just use <code>\Glsentryname</code> .	141	<code>\mfirstucMakeUppercase</code>	47
changed to just use <code>\glentryname</code> .	141	<code>\glsnavigation: switched to using</code>	
changed to just use <code>\Glsentryplural</code>	139	<code>\@gls@getgroupitle</code>	282
changed to just use <code>\glentryplural</code>	139	<code>\ifglshasdesc:replaced \ifdefempty</code>	
changed to just use		with <code>\ifcempty</code>	63
<code>\Glsentrysymbolplural</code>	144	<code>\ifglshaslong:new</code>	63
changed to just use		<code>\ifglshasshort:new</code>	63
<code>\glentrysymbolplural</code>	144	<code>\ifglshassymbol:replaced</code>	
changed to just use <code>\Glsentrysymbol</code>	143	<code>\ifdefempty with \ifcempty</code>	63
changed to just use		<code>\ifglused:replaced \ifthenelse with</code>	
<code>\glentrysymbol</code>	143, 144	<code>\ifbool</code>	60
Changed to just use <code>\Glsentrytext</code> .	138	<code>\longnewglossaryentry:new</code>	88
changed to just use <code>\glentrytext</code> .	137	<code>\ns@newglossary:replaced</code>	
changed to just use		<code>\glsdisplay and</code>	
<code>\Glsentryuseriii</code>	147	<code>\glsdisplayfirst with</code>	
changed to just use		<code>\glentryfmt</code>	67
<code>\glentryuseriii</code>	146, 147	compatible-3.07: <code>cnew</code>	34
changed to just use <code>\Glsentryuserii</code>	146	<code>\SetCustomDisplayStyle: updated to</code>	
changed to just use		use <code>\defglentryfmt</code>	264
<code>\glentryuserii</code>	145, 146	<code>\SetDefaultAcronymDisplayStyle:</code>	
changed to just use <code>\Glsentryuseriv</code>	147	changed to use <code>\defglentryfmt</code> .	247
changed to just use		<code>\SetDescriptionAcronymDisplayStyle:</code>	
<code>\glentryuseriv</code>	147, 148	updated to use <code>\defglentryfmt</code> .	254
changed to just use <code>\Glsentryuseri</code>	145	<code>\SetDescriptionDUAAcronymDisplayStyle:</code>	
changed to just use <code>\glentryuseri</code>	145	updated to use <code>\defglentryfmt</code> .	252
changed to just use <code>\Glsentryuservi</code>	149	<code>\SetDescriptionFootnoteAcronymDisplayStyle:</code>	
changed to just use <code>\glentryuservi</code>	149	updated to use <code>\defglentryfmt</code> .	250
changed to just use <code>\Glsentryuserv</code>	148	<code>\SetDUADisplayStyle: updated to use</code>	
changed to just use <code>\glentryuserv</code>	148	<code>\defglentryfmt</code>	262
Now requires textcase	4	<code>\SetFootnoteAcronymDisplayStyle:</code>	
acronymlists:replaced		updated to use <code>\defglentryfmt</code> .	256
<code>\@addtoacronymlists with</code>		<code>\SetSmallAcronymDisplayStyle:</code>	
<code>\DeclareAcronymList</code>	20	updated to use <code>\defglentryfmt</code> .	259
<code>\defgldisplay:obsoleted</code>	115	<code>\setupglossaries:new</code>	38
<code>\defgldisplayfirst:obsoleted</code>	115	<code>\showglolong:new</code>	270

\showgloshort: new	269	\glsprestandardssort: new	14
numbers: new	36	\glsssetexpandfield: new	22
symbols: new	36	\glsssetnoexpandfield: new	22
3.12a (2013-10-16)		altsuper4colheader: switched to	
\gls@defglossaryentry: added		\tabularnewline	318
\glslabel	89	altsuper4colheaderborder: switched	
\glsaddkey: new	82	to \tabularnewline	319
3.13a (2013-11-05)		long: switched to \tabularnewline ..	291
\@gls@assign@symbol@field: changed		long3col: switched to	
to use \glsssetnoexpandfield	23	\tabularnewline	292
\@gls@assign@symbolplural@field:		long3colheader: switched to	
changed to use		\tabularnewline	293
\glsssetnoexpandfield	23	long3colheaderborder: switched to	
\@gls@link: removed \relax	120	\tabularnewline	294
\@gls@notranslatorhook: new	27	long4col: switched to	
\@gls@setupsort@standard: moved		\tabularnewline	294
\@gls@santizesort		long4colheader: switched to	
\glsprestandardssort	14	\tabularnewline	295
ucmark: added check for memoir	12	longheader: switched to	
see: added \gls@checkseeallowed ...	72	\tabularnewline	292
\glossarysection: changed		longheaderborder: switched to	
\glossarymark to		\tabularnewline	292
\gls glossarymark	47	\SetFootnoteAcronymDisplayStyle:	
\glossarystyle: fixed bug caused by		fixed missing argument bug	257
using \ifdef instead of \ifcsdef .	224	super: switched to \tabularnewline .	313
\gls@assign@desc@field: changed to		super3col: switched to	
use \glsssetnoexpandfield	22	\tabularnewline	315
\gls@assign@descplural@field:		super3colheader: switched to	
changed to use		\tabularnewline	315
\glsssetnoexpandfield	22	super4col: switched to	
\gls@assign@name@field: changed to		\tabularnewline	316
use \glsssetnoexpandfield	23	super4colheader: switched to	
\gls@assign@type@field: changed to		\tabularnewline	317
use \glsssetexpandfield	22	super4colheaderborder: switched to	
\gls@checkseeallowed: new	72	\tabularnewline	317
\glsaddallunused: set default to		superheader: switched to	
\@glo@types	167	\tabularnewline	314
\Glsentryfull: changed to use		superheaderborder: switched to	
\acrfullformat	164	\tabularnewline	314
\Glsentryfull: changed to use		3.14a (2013-11-12)	
\acrfullformat	164	\@glswritefiles: renamed	
\Glsentryfullpl: changed to use		\glswritefiles to	
\acrfullformat	164	\@glswritefiles and used	
\Glsentryfullpl: changed to use		“savewrites” option to set	
\acrfullformat	164	\glswritefiles	189
\gls glossarymark: renamed		General: new	273
\glossarymark to		acronyms: new	18
\gls glossarymark to avoid conflict		\gls@defglossaryentry: added check	
with memoir	47	for existence of default glossary	90

set the default for firstplural to be the value of plural	92	\GlsUseAcrEntryDispStyle: new ...	235
xindygloss: new	31	\GlsUseAcrStyleDefs: new	235
\longprovideglossaryentry: new ...	89	short-long: new	237
compatible-2.07: added check for 2.07 before setting 3.07 compatibility	34	short-long-desc: new	240
notranslate: new	27	xindynoglsnumbers: new	31
\provideglossaryentry: new	78	sm-short-long: new	238
4.0 (2013-11-14)		sm-short-long-desc: new	240
\gls@defglossaryentry: added check for first key	92	index: new	37
super: fixed typo in \subglossentry (\glossentrydesc)	313	\newacronymstyle: new	235
4.01 (2013-11-16)		long-sc-short: new	238
General: fixed non-value options so that they can be passed to document class .	9	long-sc-short-desc: new	239
\CustomAcronymFields: inserted missing comma	265	long-short: new	236
4.02 (2013-12-05)		long-short-desc: new	239
\@acrfull: now using \acrfullfmt ..	230	long-sm-short: new	238
\@gls@indexdef: new	37	long-sm-short-desc: new	239
\@gls@numbersdef: new	37	long-sp-short-desc: new	239
\@gls@symbolsdef: new	36	footnote: new	243
General: Removed \acronymfont .	153–157	footnote-desc: new	245
\ACRfullfmt: new	231	footnote-sc: new	244
\Acrfullfmt: new	230	footnote-sc-desc: new	245
\acrfullfmt: new	230	footnote-sm: new	244
\ACRfullplfmt: new	232	footnote-sm-desc: new	245
\Acrfullplfmt: new	232	\setacronymstyle: new	234
\acrfullplfmt: new	231	\SetDescriptionAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	254
\acronymentry: new	234	\SetDescriptionFootnoteAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	250
sanitize: fixed bug that caused an error here	26	\SetFootnoteAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	256
sc-short-long: new	238	\SetGenericNewAcronym: new	233
sc-short-long-desc: new	240	\SetSmallAcronymDisplayStyle: Moved check for empty custom text to prevent unwanted parenthetical material	259
\Genacrfullformat: new	114	dua: new	241
\genacrfullformat: new	114	dua-desc: new	242
\GenericAcronymFields: new	234	numberedsection: added nameref option	9
\Genplacrfullformat: new	114	4.02 (2013-13-05)	
\Glsentryfull: bug fix: added missing \acronymfont	164	\makeglossaries: made preamble only	184
\glsentryfull: bug fix: added missing \acronymfont	164	4.03 (2014-01-17)	
\Glsentryfullpl: bug fix: added missing \acronymfont	164	General: changed default to \@empty instead of \relax	34
\glsentryfullpl: bug fix: added missing \acronymfont	164		
\glsgenacfmt: new	112		

4.03 (2014-01-20)

\@do@esc@wrglossary: added		\compatiblesubglossentry: added	
\glsdetoklabel	196	\glsdetoklabel	356
\@do@noesc@wrglossary: added		\Genacrfullformat: redefined to use	
\glsdetoklabel	193	accessibility information	377
\@ACRlong: removed \glslabel		\genacrfullformat: redefined to use	
(defined in \@gls@link)	380	accessibility information	377
\@ACRshort: removed \glslabel		\Genplacrfullformat: redefined to use	
(defined in \@gls@link)	379	accessibility information	378
\@Acrlong: removed \glslabel		\genplacrfullformat: redefined to use	
(defined in \@gls@link)	380	accessibility information	378
\@Acrshort: removed \glslabel		\glossentryname: added	
(defined in \@gls@link)	378	\glsdetoklabel	219
\@GLS@: removed \glslabel (defined in		\gls@defglossaryentry: added	
\@gls@link)	133	\glsdetoklabel	89
\@GLSpl: removed \glslabel (defined		replaced #1 with \@glo@label	90
in \@gls@link)	135	replaced \ifthenelse with	
\@Gls@: removed \glslabel (defined in		\ifdefequal	91
\@gls@link)	132	\glsadd: added \glsdetoklabel	166
\@Gls@entry@field: new	157	\glsaddkey: switched to using	
\@Glspl@: removed \glslabel (defined		\@gls@field@link	83
in \@gls@link)	135	\glsdetoklabel: new	60
\@acrlong: removed \glslabel		\glsdisplaynumberlist: added	
(defined in \@gls@link)	379	\glsdetoklabel	165
\@acrshort: removed \glslabel		\glsdoifexistsorwarn: new	61
(defined in \@gls@link)	378	\glsentryaccess: switched to using	
\@gls@: removed \glslabel (defined in		\@gls@entry@field	361
\@gls@link)	131	\glsentrydescaccess: switched to	
\@gls@access@display: new	365	using \@gls@entry@field	362
\@gls@entry@field: new	157	\glsentrydescpluralaccess: switched	
\@gls@fetchfield: new	81	to using \@gls@entry@field	362
\@gls@field@link: new	137	\glsentryfirstaccess: switched to	
\@gls@link: added \glsdetoklabel .	119	using \@gls@entry@field	362
moved \@gls@link@opts and		\glsentryfirstplural: added	
\@gls@link@label to \@gls@link	119	\glsdetoklabel	161
\@gls@writedef: added		\glsentrylongaccess: switched to	
\glsdetoklabel	80	using \@gls@entry@field	363
\@glsdisp: removed \glslabel		\glsentrylongpluralaccess: switched	
(defined in \@gls@link)	136	to using \@gls@entry@field	363
\@glspl@: removed \glslabel (defined		\glsentrypluralaccess: switched to	
in \@gls@link)	134	using \@gls@entry@field	362
\@printglossary: added		\glsentryshortaccess: switched to	
\glsdetoklabel	203	using \@gls@entry@field	362
General: removed \glslabel (defined in		\glsentryshortpluralaccess:	
\@gls@link)	150	switched to using	
sc-short-long-desc: redefined to use		\@gls@entry@field	362
accessibility information	390	\glsentrysymbolaccess: switched to	
\compatibleglossentry: added		using \@gls@entry@field	362
\glsdetoklabel	355	\glsentrysymbolpluralaccess:	
		switched to using	

\@gls@entry@field	362	replaced \ifcsempy with	
\glsentrytextaccess: switched to		\ifdefempty and replaced \ifx with	
using \@gls@entry@field	361	\ifdefequal	63
\glsgenacfmt: redefined to use		\ifglsused: added \glsdetoklabel ..	60
accessibility information	375	sm-short-long-desc: redefined to use	
\glsgenentryfmt: redefined to use		accessibility information	390
accessibility information	372	long-sc-short-desc: redefined to use	
\glshyperlink: added		accessibility information	389
\glsdetoklabel	166	long-short: redefined to use	
\glslocalreset: added		accessibility information	387
\glsdetoklabel	98	long-short-desc: redefined to use	
\glslocalunset: added		accessibility information	388
\glsdetoklabel	98	long-sm-short-desc: redefined to use	
\glsmoveentry: added		accessibility information	389
\glsdetoklabel	95	footnote: redefined to use accessibility	
replaced \ifthenelse with		information	393
\ifdefequal	95	footnote-desc: redefined to use	
\glsrefentry: added \glsdetoklabel	218	accessibility information	395
\glsreset: added \glsdetoklabel ...	97	footnote-sc: redefined to use	
\glsseelist: added \expandafter		accessibility information	395
commands	199	footnote-sc-desc: redefined to use	
\glsstepentry: added		accessibility information	396
\glsdetoklabel	217	footnote-sm: redefined to use	
\glsstepsubentry: added		accessibility information	395
\glsdetoklabel	217	footnote-sm-desc: redefined to use	
\glsunset: added \glsdetoklabel ...	98	accessibility information	396
short-long: commented spurious EOL	237	\renewacronymstyle: new	235
redefined to use accessibility		\showglocounter: added	
information	388	\glsdetoklabel	267
short-long-desc: redefined to use		\showglodesc: added \glsdetoklabel	269
accessibility information	389	\showglodescaccess: added	
\ifglsdescsuppressed: added		\glsdetoklabel	402
\glsdetoklabel	63	\showglodescplural: added	
fixed typo	63	\glsdetoklabel	269
\ifglsentryexists: added		\showglodescpluralaccess: added	
\glsdetoklabel	60	\glsdetoklabel	402
\ifglschaschildren: added		\showglofirst: added	
\glsdetoklabel	62	\glsdetoklabel	267
\ifglschasdesc: added		\showglofirstaccess: added	
\glsdetoklabel	63	\glsdetoklabel	402
\ifglschasfield: new	64	\showglofirstpl: added	
\ifglschaslong: added		\glsdetoklabel	267
\glsdetoklabel	63	\showglofirstpluralaccess: added	
\ifglschasparent: added		\glsdetoklabel	402
\glsdetoklabel	62	\showgloflag: added \glsdetoklabel	270
\ifglschasshort: added		\showgloindex: added	
\glsdetoklabel	63	\glsdetoklabel	270
\ifglschassymbol: added		\showglolevel: added	
\glsdetoklabel	63	\glsdetoklabel	266

\showglolong: added \glsdetoklabel	270	redefined to use accessibility	
\showglolongaccess: added		information	390
\glsdetoklabel	403	dua-desc: commented spurious EOL ..	243
\showglolongpluralaccess: added		redefined to use accessibility	
\glsdetoklabel	403	information	393
\showglongname: added \glsdetoklabel	268	4.04 (2014-03-04)	
\showglongnameaccess: added		\@gls@getcounterprefix: added	
\glsdetoklabel	401	warning if no prefix can be formed ..	197
\showgloparent: added		4.04 (2014-03-06)	
\glsdetoklabel	266	\@gls@noidx@nosanitizesort: new ..	24
\showgloplural: added		\@gls@noidx@sanitizesort: new ...	23
\glsdetoklabel	267	\@gls@nosanitizesort: new	23
\showglopluralaccess: added		\@gls@sanitizesort: new	23
\glsdetoklabel	402	\@glo@addchildren: new	205
\showgloshort: added		\@glo@do@sortentries: new	206
\glsdetoklabel	269	\@glo@grabfirst: new	211
\showgloshortaccess: added		\@glo@sortedinsert: new	206
\glsdetoklabel	402	\@glo@sortentries: new	205
\showgloshortpluralaccess: added		\@glo@sorthandler@case: new	207
\glsdetoklabel	403	\@glo@sorthandler@letter: new ...	207
\showglosort: added \glsdetoklabel	269	\@glo@sorthandler@nocase: new ...	207
\showglosymbol: added		\@glo@sorthandler@word: new	207
\glsdetoklabel	269	\@glo@sortmacro@case: new	208
\showglosymbolaccess: added		\@glo@sortmacro@def: new	209
\glsdetoklabel	402	\@glo@sortmacro@def@do: new	209
\showglosymbolplural: added		\@glo@sortmacro@letter: new	208
\glsdetoklabel	269	\@glo@sortmacro@nocase: new	209
\showglosymbolpluralaccess: added		\@glo@sortmacro@standard: new ...	208
\glsdetoklabel	402	\@glo@sortmacro@use: new	210
\showglotext: added \glsdetoklabel	266	\@glo@sortmacro@word: new	208
\showglotextaccess: added		\@gls@noidx@do: new	211
\glsdetoklabel	402	\@gls@noidx@getgroup title: new ..	223
\showglotype: added \glsdetoklabel	267	\@gls@noref@warn: new	188
\showglouser i: added		\@gls@reference: new	214
\glsdetoklabel	267	\@gls@warnonglossdefined: new	21
\showglouser ii: added		\@gls@warnonthe glossdefined: new ..	22
\glsdetoklabel	268	\@no@makeglossaries: new	188
\showglouser iii: added		\@print@glossary: new	203
\glsdetoklabel	268	\@print@noidx@glossary: new	210
\showglouser iv: added		\@print@gloss@setsort: new	201
\glsdetoklabel	268	\@print@glossary: new	201
\showglouser v: added		General: added sort key to printgloss	
\glsdetoklabel	268	group	216
\showglouser vi: added		\compatibleglossentry: changed	
\glsdetoklabel	268	\newcommand to \def as is may or	
dua: fixed bug in \acrfullfmt	242	may not be defined	355
fixed bug in \Acrfullplfmt	242	\compatiblesubglossentry: changed	
fixed bug in \acrfullplfmt	242	\newcommand to \def as is may or	
		may not be defined	356

\defglsdisplayfirst: fixed unwanted space	115	\Acrfullplfmt: fixed no case change bug	232
\glo@grabfirst: new	211	\glsletentryfield: new	157
\gls@defglossaryentry: replaced \ifx with \ifdefvoid	94	4.08 (2014-07-30)	
\glsnoidxdisplayloc: new	213	\@ACRlong: added	
\glsnoidxdisplaylocclsthander: new	213	\do@gls@link@checkfirsthyper	380
\glsnoidxlocclsthander: new	213	\@ACRshort: added	
\glsnoidxstripaccents: new	24	\do@gls@link@checkfirsthyper	379
alltree: moved hangindent and parindent assignments outside level test	332	\@Acrlong: added	
\makeglossaries: Moved definition of \glswrite to \makeglossaries ..	183	\do@gls@link@checkfirsthyper	379
\makenoidxglossaries: new	185	\@Acrshort: added	
\printglossary: changed to use new \@printglossary	201	\do@gls@link@checkfirsthyper	378
\printnoidxglossaries: new	201	\@GLS@: moved \glsifhyper	133
\printnoidxglossary: new	201	moved check for first use to	
\showgloclsthander: new	270	\@gls@link	133
\warn@noprintglossary: Activate warning in \makeglossaries	200	\@GLSpl: moved \glsifhyper	136
\writeist: checked for definition of \glswrite	170, 174	moved check for first use to	
4.06 (2014-03-12)		\@gls@link	136
\@GLS@: added \glsifhyper	133	\@Gls@: moved \glsifhyper	132
\@GLSpl: added \glsifhyper	136	moved check for first use to	
\@Gls@: added \glsifhyper	132	\@gls@link	132
\@GLspl@: added \glsifhyper	135	\@GLspl@: moved \glsifhyper	135
\@gls@: added \glsifhyper	132	moved check for first use to	
\@gls@numbersdef: added hook to set toc title	37	\@gls@link	135
\@gls@symbolsdef: added hook to set toc title	36	\@acrlong: added	
\@glsdisp: added \glsifhyper	136	\do@gls@link@checkfirsthyper	379
\@GLspl@: added \glsifhyper	134	\@acrshort: added	
General: added \glsifhyper	150–157	\do@gls@link@checkfirsthyper	378
acronym: added hook to set toc title	18	\@closegls: new	180
acronyms: added hook to set toc title ...	18	\@gls@: moved \glsifhyper	132
\glsdefmain: added hook to set toc title	17	moved check for first use to	
4.07 (2014-04-04)		\@gls@link	132
\@glossarysection: added optional argument when using unstarred version	48	\@gls@automake: new	180
\@gls@noidx@do: added \global in case it's used in a tabular-like style	211	\@gls@doautomake: new	34
		\@gls@field@link: added assignment of	
		\do@gls@link@checkfirsthyper	137
		\@gls@forbidtexext: new	66
		\@gls@hyp@opt: new	117
		\@gls@link: removed redundancy	119
		renamed \gls@type to \glstype ...	119
		\@gls@link@checkfirsthyper: new .	118
		\@glsdisp: moved \glsifhyper	136
		moved check for first use to	
		\@gls@link	136
		\@GLspl@: moved \glsifhyper	134
		moved check for first use to	
		\@gls@link	134
		\@ignored@glossaries: new	69

General: added entrycounter option to		removed \@sGLStext	137
printgloss family	215	removed \@sGlstext	138
added nopostdot option to		removed \@sglstext	137
printgloss family	215	removed \@sGLSuseriii	147
added subentrycounter option to		removed \@sGlsuseriii	146
printgloss family	215	removed \@sglsuseriii	146
explicitly initialise hyper key	116	removed \@sGLSuserii	146
moved \glsifhyper	150–157	removed \@sGlsuserii	146
removed \@sACRlongpl	156	removed \@sglsuserii	145
removed \@sAcrlongpl	156	removed \@sGLSuseriv	148
removed \@sacrlongpl	155	removed \@sGlsuseriv	147
removed \@sACRlong	154	removed \@sglsuseriv	147
removed \@sAcrlong	154	removed \@sGLSuseri	145
removed \@sacrlong	153	removed \@sGlsuseri	145
removed \@sACRshortpl	152	removed \@sglsuseri	145
removed \@sAcrshortpl	152	removed \@sGLSsuservi	149
removed \@sacrshortpl	151	removed \@sGlsuservi	149
removed \@sACRshort	151	removed \@sglsuservi	149
removed \@sAcrshort	150	removed \@sGLSsuserv	148
removed \@sacrshort	149	removed \@sGlsuserv	148
removed \@sgls@link	118	removed \@sglsuserv	148
removed \@sGLSdescplural	143	removed \@sGLS	133
removed \@sGlsdescplural	142	removed \@sGls	132
removed \@sglsdescplural	142	removed \@sgls	131
removed \@sGLSdesc	142	removed \@thirdofthree (defined in	
removed \@sGlsdesc	142	kernel)	131
removed \@sglsdesc	141	removed sPGLS	278
removed \@sglsdisp	136	removed sPgls	277
removed \@sGLSfirstplural	140	removed spgls	275
removed \@sGlsfirstplural	140	removed sPGLSpl	279
removed \@sglsfirstplural	140	removed sPglspl	277
removed \@sGLSfirst	139	removed spglspl	276
removed \@sGlsfirst	138	\ACRfull: removed \s@ACRfull	231
removed \@sglsfirst	138	switched to using \@gls@hyp@opt ..	231
removed \@sGLSname	141	\Acrfull: removed \s@Acrfull	230
removed \@sGlsname	141	switched to using \@gls@hyp@opt ..	230
removed \@sglsname	140	\acrfull: removed \@sacrfull	229
removed \@sGLSplural	139	switched to using \@gls@hyp@opt ..	229
removed \@sGlsplural	139	\ACRfullpl: removed \s@ACRfullpl ..	232
removed \@sglsplural	139	switched to using \@gls@hyp@opt ..	232
removed \@sGLSpl	135	\Acrfullpl: removed \s@Acrfullpl ..	231
removed \@sGlspl	134	switched to using \@gls@hyp@opt ..	231
removed \@sglspl	134	\acrfullpl: removed \s@acrfullpl ..	231
removed \@sGLSsymbolplural	144	switched to using \@gls@hyp@opt ..	231
removed \@sGlsymbolplural	144	\ACRlong: switched to using	
removed \@sglsymbolplural	144	\@gls@hyp@opt	154
removed \@sGLSsymbol	143	\Acrlong: switched to using	
removed \@sGlsymbol	143	\@gls@hyp@opt	153
removed \@sglsymbol	143		

\acrlong: switched to using \@gls@hyp@opt	153	\glsdohyperlink: new	130
\ACRlongpl: switched to using \@gls@hyp@opt	156	\glsdohypertarget: new	130
\Acrlongpl: switched to using \@gls@hyp@opt	155	\glsenablehyper: added \KV@glslink@hypertrue to definition	131
\acrlongpl: switched to using \@gls@hyp@opt	155	\GLSfirst: switched to using \@gls@hyp@opt	138
\ACRshort: switched to using \@gls@hyp@opt	150	\Glsfirst: switched to using \@gls@hyp@opt	138
\Acrshort: switched to using \@gls@hyp@opt	150	\glsfirst: switched to using \@gls@hyp@opt	138
\acrshort: switched to using \@gls@hyp@opt	149	\GLSfirstplural: switched to using \@gls@hyp@opt	140
\ACRshortpl: switched to using \@gls@hyp@opt	152	\Glsfirstplural: switched to using \@gls@hyp@opt	140
\Acrshortpl: switched to using \@gls@hyp@opt	152	\glsfirstplural: switched to using \@gls@hyp@opt	140
\acrshortpl: switched to using \@gls@hyp@opt	151	\glsifhyper: deprecated	117
\forallacronyms: new	59	\glslink: switched to using \@gls@hyp@opt	118
\GLS: switched to using \@gls@hyp@opt	133	\glslinkcheckfirsthyperhook: new	119
\Gls: switched to using \@gls@hyp@opt	132	\glslinkvar: new	117
\gls: switched to using \@gls@hyp@opt	131	\GLSname: switched to using \@gls@hyp@opt	141
\glsdefglossaryentry: added check for ignored glossary	90	\Glsname: switched to using \@gls@hyp@opt	141
\gls@istfilebase: new	44	\glsname: switched to using \@gls@hyp@opt	140
\glsaddkey: removed \@sGLS@user@<key>	84	\GLSpl: switched to using \@gls@hyp@opt	135
removed \@sGls@user@<key>	84	\Glspl: switched to using \@gls@hyp@opt	134
removed \@sgls@user@<key>	83	\glspl: switched to using \@gls@hyp@opt	134
switched to using \@gls@hyp@opt	83, 84	\GLSplural: switched to using \@gls@hyp@opt	139
\GLSdesc: switched to using \@gls@hyp@opt	142	\Glsplural: switched to using \@gls@hyp@opt	139
\Glsdesc: switched to using \@gls@hyp@opt	141	\glsplural: switched to using \@gls@hyp@opt	139
\glsdesc: switched to using \@gls@hyp@opt	141	\glsspace: new	230
\GLSdescplural: switched to using \@gls@hyp@opt	143	\GLSsymbol: switched to using \@gls@hyp@opt	143
\Glsdescplural: switched to using \@gls@hyp@opt	142	\Glsymbol: switched to using \@gls@hyp@opt	143
\glsdescplural: switched to using \@gls@hyp@opt	142	\glssymbol: switched to using \@gls@hyp@opt	143
\glsdisablehyper: added \KV@glslink@hyperfalse to definition	130	\GLSsymbolplural: switched to using \@gls@hyp@opt	144
\glsdisp: switched to using \@gls@hyp@opt	136		

\Glsymbolplural: switched to using \@gls@hyp@opt	144	altlongragged4col: fixed bug that displayed description instead of symbol	305
\glssymbolplural: switched to using \@gls@hyp@opt	144	\newglossary: added starred version ..	67
\GLStext: switched to using \@gls@hyp@opt	137	\newignoredglossary: new	69
\Glstext: switched to using \@gls@hyp@opt	138	\ns@newglossary: added \@glotype@<name>@log	67
\glstext: switched to using \@gls@hyp@opt	137	new	67
\glstreenamefmt: new	325	\p@gls@hyp@opt: new	117
\GLSuseri: switched to using \@gls@hyp@opt	145	\PGLS: changed to use \@gls@hyp@opt	278
\Glsuseri: switched to using \@gls@hyp@opt	145	\Pgls: changed to use \@gls@hyp@opt	277
\glsuseri: switched to using \@gls@hyp@opt	145	\pgls: changed to use \@gls@hyp@opt	275
\GLSuserii: switched to using \@gls@hyp@opt	146	\PGLSpl: changed to use \@gls@hyp@opt	279
\Glsuserii: switched to using \@gls@hyp@opt	146	\Pglspl: changed to use \@gls@hyp@opt	277
\glsuserii: switched to using \@gls@hyp@opt	145	\pglspl: changed to use \@gls@hyp@opt	276
\GLSuseriii: switched to using \@gls@hyp@opt	147	\s@gls@hyp@opt: new	117
\Glsuseriii: switched to using \@gls@hyp@opt	146	\s@newglossary: new	67
\glsuseriii: switched to using \@gls@hyp@opt	146	automake: new	33
\GLSuseriv: switched to using \@gls@hyp@opt	147	4.09 (2014-08-12)	
\Glsuseriv: switched to using \@gls@hyp@opt	147	\glsaddkey: fixed bug in user commands	83
\glsuseriv: switched to using \@gls@hyp@opt	147	4.10 (2014-08-27)	
\GLSuserv: switched to using \@gls@hyp@opt	148	\@Gls@acentryname: new	158
\Glsuserv: switched to using \@gls@hyp@opt	148	\@Gls@entryname: new	158
\glsuserv: switched to using \@gls@hyp@opt	148	\@gls@glossary: Renamed \@glossary to \@gls@glossary	190
\ifignoredglossary: new	69	\glspercentchar: new	168
		\glstildechar: new	168
		alttree: moved space after symbol	332, 333
		4.11 (2014-09-01)	
		\@do@esc@wrglossary: added hook ..	195
		sanitize: none option	26
		\gls@wrglossary: renamed from \@wrglossary to \gls@wrglossary	191
		\glsaddprotectedpagefmt: new	192
		\glsbackslash: new	168
		4.12 (2014-11-22)	
		\@gls@addpredefinedattributes: Added glsignore attribute	53
		\@gls@adjustmode: new	167
		\@gls@nottranslatorhook: removed ...	27
		\@gls@toc: added \protect to \numberline	49
		\@gls@usetranslator: new	27
		\glsacrpluralsuffix: new	41
		\glsadd: added check for vertical mode	166

\glsaddallunused: replaced @gobble with glsignore	167	\glsunset: switched to \@glsunset ...	98
\glsifusedtranslatordict: new	27	4.15 (2015-03-16)	
\glsignore: new	167	General: bug fix replaced \@glo@type with \glstype	156
\glsupacrpluralsuffix: new	41	4.16 (2015-06-18)	
\ProvidesGlossariesLang: new	41	\glsaddstoragekey: new	82
\RequireGlossariesLang: new	41	4.16 (2015-07-08)	
4.13 (2015-02-03)		\@ACRlong: added \glspostlinkhook	380
\indexspace: new	286, 307, 325	\@ACRshort: added \glspostlinkhook	379
4.14 (2015-02-28)		\@Acrlong: added \glspostlinkhook	380
\@@glslocalreset: new	99	\@Acrshort: added \glspostlinkhook	379
\@@glslocalunset: new	99	\@GLS@: added \glspostlinkhook ...	133
\@@glsreset: new	99	\@GLSpl: added \glspostlinkhook ..	136
\@@glsunset: new	99	\@Gls@: added \glspostlinkhook ...	133
\@@newglossaryentry@defcounters: new	100	\@Glspl@: added \glspostlinkhook .	135
\cGls: new	103	\@acrlong: added \glspostlinkhook	379
\cGls@: new	104	\@acrshort: added \glspostlinkhook	378
\cGlspl@: new	105	\@gls@: added \glspostlinkhook ...	132
\cgls: new	103	\@gls@@link: added \glspostlinkhook	118
\cgls@: new	103	\@gls@field@link: added \glspostlinkhook	137
\cglspl: new	104	\@gls@link: moved definition of \glsifhyperon outside of this macro	120
\cglspl@: new	104	\@glsdisp: added \glspostlinkhook	137
\@gls@entry@count: new	103	\@glspl@: added \glspostlinkhook .	134
\@gls@increment@currcount: new ..	102	General: added \glspostlinkhook	150–157
\@gls@local@increment@currcount: new	102	\glsacspace: new	237
\@gls@write@entrycounts: new	103	\glsadd: changed \@do@wrglossary to \@do@wrglossary	167
\@glslocalreset: new	99	\glsfielddef: new	86
\@glslocalunset: new	98	\glsfieldedef: new	85
\@glsreset: new	99	\glsfieldfetch: new	86
\@glsunset: new	99	\glsfieldgdef: new	85
\@newglossaryentry@defcounters: new	95	\glsfieldxdef: new	85
\cGls: new	103	\glsifhyperon: moved definition of \glsifhyperon	119
\cgls: new	103	\glslinkpostsetkeys: new	119
\cGlsformat: new	104	\glspostlinkhook: new	118
\cglsformat: new	103	\glswriteentry: new	191
\cGlspl: new	104	\ifglsfieldcseq: new	88
\cglspl: new	104	\ifglsfielddefeq: new	87
\cGlsplformat: new	105	\ifglsfieldefeq: new	87
\cglsplformat: new	104	long-sp-short: new	236
\gls@defdocnewglossaryentry: new .	77	\showglofield: new	270
\glsenableentrycount: new	100	4.18 (2015-09-09)	
\glslocalreset: switched to \@glslocalreset	98	General: split mfirstuc into separate bundle	4
\glslocalunset: switched to \@glslocalunset	98		
\glsreset: switched to \@glsreset ...	97		

4.19 (2015-10-31)		\gls@arabicpage: new	192
\glstreenamebox: new	331	\gls@protected@pagefmts: added	
4.19 (2015-11-22)		arabic to list	192
\@gls@link@nocheckfirsthyper: new	137	\glstrytitlecase: new	161
\@gls@preglossaryhook: new	201	\glsfindwidesttoplevelname: new	331
\@printglossary: added		\glslistgroupheaderfmt: new	287
\@gls@preglossaryhook	203	\glslistnavigationitem: new	287
\do@glsglisablehyperinlist: new	119	\glstreegroupheaderfmt: new	325
\doifglossarynoexistsordo: new	62	\glstreenavigationfmt: new	325
\gls@gobbleopt: new	66	\ifglswrallowprimitivemods: new	193
\glsoifexistsordo: new	61	list: fixed missing space before	
4.20 (2015-11-30)		description	288
\@gls@link: added		long: fixed typo in \glossentrydesc	291
\@gls@setdefault@glslink@opts	119	super4col: fixed bug in \glossentry	316
added \glsohhyperlink when		4.23 (2016-04-30)	
hyperlink is suppressed	120	\glscurrentfieldvalue: new	65
\@gls@setdefault@glslink@opts:		\ifglshasfield: added	
new	119	\glscurrentfieldvalue	64, 65
\gls@checkseeallowed@preambleonly:		altlongragged4col: check for	
new	72	nogroupskip changed	305
\glsohhyperlink: new	130	altsuperragged4col: check for	
4.21 (2016-01-24)		nogroupskip changed	324
\@printglossary: warn if no style has		long: check for nogroupskip changed	291
been set	202	long-booktabs: check for nogroupskip	
General: changed checkfirsthyper		changed	297
assignment	150–156	long3col: check for nogroupskip	
\glossarystyle: set default style if not		changed	293
already set	224	long3col-booktabs: check for	
\glsltpenaltycheck: new	300	nogroupskip changed	298
\glspatchLToutput: new	300	long4col: check for nogroupskip	
\glspenaltygroupskip: new	300	changed	294
altlong4col-booktabs: new	299	long4col-booktabs: check for	
altlongragged4col-booktabs: new	300	nogroupskip changed	298
long-booktabs: new	297	longragged: check for nogroupskip	
long3col-booktabs: new	297	changed	302
long4col-booktabs: new	298	longragged3col: check for nogroupskip	
longragged-booktabs: new	299	changed	304
longragged3col-booktabs: new	299	super: check for nogroupskip changed	313
\setglossarystyle: set default style if		super3col: check for nogroupskip	
not already set	224	changed	315
4.22 (2016-04-19)		super4col: check for nogroupskip	
\@do@esc@wrglossary: added check		changed	317
for \@arabic	195	superragged: check for nogroupskip	
added test to allow temporary primitive		changed	320
modifications and added arabic case	195	superragged3col: check for	
mcolalttreespannav: new	312	nogroupskip changed	322
mcolindexspannav: new	308	4.24 (2016-05-27)	
mcoltreenonamespannav: new	310	\@gls@extramakeindexopts: new	179
mcoltreespannav: new	309		

\@gls@glossary: added check for debug mode	190	\@gls@setupsort@none: new	16
\@gls@see@noindex: new	7	\@xdycrossrefhook: new	55
debug: new	5	\@xdylocationclassorder: bug fix: changed \edef to \def	56
seenoinindex: new	8	\glosortentrieswarning: new	21
\glsnomakeindexwarning: new	50	\gls@set@xr@key: new	72
\GlsSetQuote: new	175	\gls@xr@key: new	72
\GlsSetWriteIstHook: new	175	\GlsAddXdyLocation: bug fix: changed #1 to #2	55
4.25 (2016-06-09)		\glsnoidxstripaccents: added \a ...	24
\@gls@enablesavenonumberlist: new	73	added \TH, \dh and \DH	25
\@gls@initnonumberlist: new	73	4.31 (2017-08-10)	
\@gls@savenonumberlist: new	73	nolist: added check for “list” style	11
4.25 (??)		4.31 (2017-09-10)	
General: changed		style: changed \renewcommand to \def .	9
\DeclareRobustCommand to		4.32 (2017-08-24)	
\newrobustcmd and changed		\@glsnavhypertarget: new	281
\@ifundefined to \ifcsundef ...	368	\@glsshowtarget: new	7
4.26 (2016-10-12)		\glsshowtarget: new	6
\@glossary@default@style: added		4.33 (2017-09-20)	
check for classicthesis	9	\@do@esc@wrglossary: added	
mcolindex: replaced \@idxitem with		\gls@the and \gls@number	195
\glstreeitem	307	renamed from	
mcolindexspannav: replaced \@idxitem		\@do@esc@wrglossary	194
with \glstreeitem	308	\@do@noesc@wrglossary: new	193
\glstreechildpredesc: new	326	\@do@wrglossary: changed to check	
\glstreeitem: new	326	for esclocations	193
\glstreepredesc: new	326	\@gls@missinglang@warn: new	21
\glstreesubitem: new	326	\GlsSetXdyFirstLetterAfterDigits:	
\glstreesubsubitem: new	326	added starred version	168
4.28 (2017-01-07)		\GlsSetXdyNumberGroupOrder: new .	169
\glspatchtabularx: new	97	esclocations: new	10
4.29 (2017-01-19)		4.34 (2017-11-03)	
\@gls@noidx@do: current letter group		mcolalttreespannav: removed spurious	
assignment made global	212	space	312
\@print@noidx@glossary: moved		\glsshowtarget: modified to check for	
definition of		math mode and inner	6
\@gls@currentlettergroup outside		4.35 (2017-11-14)	
of the glossary environment	210	\glsadd: added \@gls@setsort (in case	
General: added check for		of sort=use)	167
\@glsxtr@doaccsupp	355	4.36 (2018-03-07)	
\glsnavhyperlinkname: new	280	\@gls@glossary: removed \index ...	190
4.30 (2017-06-11)		4.37 (2018-04-07)	
\@glo@autosee: new	94	\gls@begindocdefs: new	79
\@glo@autoseehook: new	94	4.38 (2018-05-10)	
\@glo@check@sortallowed: new	14	\@gls@define@glossaryentrycounter:	
\@gls@noidx@do: letter group		added check for existence of	
assignment made global	212	glossaryentry counter	12
\@gls@setupsort@def: added check for		new	12
register	15		

\@gls@define@glossarysubentrycounter: new 13	nonumberlist: changed \val and \nr to \gls@nonumberlist@val and \gls@nonumberlist@nr 72
prepended \currentglossary. to \theHglossarysubentry and removed spurious eol space 13	translate: changed \val and \nr to \gls@translate@val and \gls@translate@nr 28
\glsaccsupp: added braces around actual text argument 365	numberedsection: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr 9
\glsentrycounterlabel: bug fix: move conditional inside command 218	4.42 (2019-01-06)
\GlsEntryCounterLabelPrefix: new 215	\@gls@@automake@immediate: new .. 182
\glsentryitem: bug fix: move conditional inside command 218	\@gls@automake@immediate: new ... 181
\glsrefentry: bug fix: move conditional inside command 218	\gls@automake@nr: new 33
\glsresetsubentrycounter: bug fix: move conditional inside command . 217	\glsfieldedef: changed from \edef to \protected@csedef 85
\glsstepentry: bug fix: move conditional inside command 217	\glsfieldxdef: changed from \edef to \protected@csxdef 85
\glsstepsubentry: bug fix: move conditional inside command 217	\ifglsautomake: now defined explicitly instead of through boolean key 33
\glsesubentrycounterlabel: bug fix: move conditional inside command . 218	noglossaryindex: new 37
\glsesubentryitem: bug fix: move conditional inside command 218	automake: switch from boolean to choice 33
\showglongnameaccess: bug fix: corrected field (was showing text access field) 401	4.42 (??)
4.40 (2018-06-01)	altlong4col-booktabs: removed superfluous \glspatchLtoutput . 299
\istfile: changed \def to \providecommand 189	4.43 (2019-09-28)
\makenoidxglossaries: false 185	\glsnoidxstripaccents: add check for LaTeX version 2019/10/01 25
4.41 (2018-07-23)	4.44 (2019-12-06)
\@gls@override@glossary: new 35	\@gls@prefix@record@hook: new 275
General: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr 215	4.45 (2020-02-13)
debug: changed \val and \nr to \gls@debug@val and \gls@debug@nr 5	\@do@write@glslabels: new 32
seenoindex: changed \val and \nr to \gls@seenoindex@val and \gls@seenoindex@nr 8	\@glsshows@target: new 7
kernelglossredefs: new 35	\@GLSdesc@: added accessibility support 382
\glossary: added warning 35	\@GLSdescplural@: added accessibility support 383
\gls@original@glossary: new 34	\@GLSfirst@: added accessibility support 381
\gls@original@makeglossary: new .. 34	\@GLSfirstplural@: added accessibility support 381
\makeglossaries: removed redefinition of \makeglossary 183	\@GLSname@: added accessibility support 382
\makeglossary: added warning 35	\@GLSplural@: added accessibility support 381
	\@GLSsymbol@: added accessibility support 383
	\@GLSsymbolplural@: added accessibility support 383
	\@GLStext@: added accessibility support 380
	\@GLSuseri@: added accessibility support 384

\@GLSuserii@: added accessibility support	384	\@glsname@: added accessibility support	382
\@GLSuseriii@: added accessibility support	384	\@glsplural@: added accessibility support	381
\@GLSuseriv@: added accessibility support	385	\@glsshowaccsupp: new	7
\@GLSuserv@: added accessibility support	385	\@glssymbol@: added accessibility support	383
\@GLSuservi@: added accessibility support	385	\@glssymbolplural@: added accessibility support	383
\@Gls@acentryname: added check for \glsshortaccessdisplay	159	\@glstext@: added accessibility support	380
\@Glsdesc@: added accessibility support	382	\@glsuseri@: added accessibility support	383
\@Glsdescplural@: added accessibility support	382	\@glsuserii@: added accessibility support	384
\@Glsfirst@: added accessibility support	381	\@glsuseriii@: added accessibility support	384
\@Glsfirstplural@: added accessibility support	381	\@glsuseriv@: added accessibility support	385
\@Glsname@: added accessibility support	382	\@glsuserv@: added accessibility support	385
\@Glsplural@: added accessibility support	381	\@glsuservi@: added accessibility support	385
\@Glsymbol@: added accessibility support	383	General: removed backward compatibility use of symbol key	359
\@Glsymbolplural@: added accessibility support	383	acronyms: changed \renewcommand to \def	18
\@Glstext@: added accessibility support	380	debug: showaccsupp	5
\@Glsuseri@: added accessibility support	384	restoremakegloss: new	32
\@Glsuserii@: added accessibility support	384	\@gls@accessibility: new	356
\@Glsuseriii@: added accessibility support	384	\@gls@accsupp@engine: new	356
\@Glsuseriv@: added accessibility support	385	\@gls@accessibility: new	356
\@Glsuserv@: added accessibility support	385	\@glsdefaultshortaccess: new	396
\@Glsuservi@: added accessibility support	385	\@glsdescriptionaccessdisplay: added check for existence	366
\@do@write@glslabels: new	32	\@glsdescriptionpluralaccessdisplay: added check for existence	366
\@domakeglossaries: new	31	\@glsentrydescpluralaccess: corrected field reference	362
\@gls@fieldaccess@display: new ..	365	\@glsentryfirstpluralaccess: switched to using \@gls@entry@field	362
\@glsdesc@: added accessibility support	382	\@glsentryparent: new	162
\@glsdescplural@: added accessibility support	382	\@glsentryprefix: added \@glsdetoklabel	274
\@glsfirst@: added accessibility support	381	\@glsentryprefix: added \@glsdetoklabel	274
\@glsfirstplural@: added accessibility support	381	\@glsentryprefixfirst: added \@glsdetoklabel	274
		\@glsentryprefixfirst: added \@glsdetoklabel	274

\Glsentryprefixfirstplural: added		\glssymbolpluralaccessdisplay:	
\glstoklabel 274		added check for existence 366	
\Glsentryprefixfirstplural: added		\glstextaccessdisplay: added check	
\glstoklabel 274		for existence 365	
\Glsentryprefixplural: added		\glsseriaccessdisplay: new 367	
\glstoklabel 274		\glsseriiaccessdisplay: new 367	
\Glsentryprefixplural: added		\glsseriiiaccessdisplay: new ... 367	
\glstoklabel 274		\glsserivaccessdisplay: new 368	
\Glsentrytitlecase: added existence		\glsservaccessdisplay: new 368	
check 161		\glsserviaccessdisplay: new 368	
\Glsentryuseriaccess: new 363		\ifglshaschildren: made robust 62	
\Glsentryuseriiaccess: new 363		\ifglshasfield: made robust 64	
\Glsentryuseriiiaccess: new 363		\ifglshaslong: made robust 63	
\Glsentryuserivaccess: new 363		\ifglshasprefix: added	
\Glsentryuservaccess: new 363		\glstoklabel 275	
\Glsentryuserviaccess: new 363		\ifglshasprefixfirst: added	
\glstfieldaccsupp: new 364		\glstoklabel 275	
\glstfirstaccessdisplay: added check		\ifglshasprefixfirstplural: added	
for existence 366		\glstoklabel 275	
\glstfirstpluralaccessdisplay:		\ifglshasprefixplural: added	
added check for existence 366		\glstoklabel 275	
\glstlongaccessdisplay: added check		\ifglshashshort: made robust 63	
for existence 367		\ifglshassymbol: made robust 63	
\glstlongpluralaccessdisplay: added		disablemakegloss: new 31	
check for existence 367		\makeglossaries: let \@makeglossary	
\glstnameaccessdisplay: added check		to \@gobble instead of \relax 183	
for existence 365		writetoklabels: new 33	
\glstpluralaccessdisplay: added		user1access: new 358	
check for existence 366		user2access: new 358	
\glstprefixsep: new 275		user3access: new 358	
\glstsee: switched to \newrobustcmd . 198		user4access: new 359	
\glstseeformat: switched to		user5access: new 359	
\newrobustcmd 198		user6access: new 359	
\glstseeitem: switched to		\xglstfieldaccsupp: new 364	
\newrobustcmd 199	4.46 (2020-03-19)	\@ifglossaryexists: new 60	
\glstseelist: switched to		\@printgloss@checkexists: new ... 200	
\newrobustcmd 198		\@printgloss@checkexists@allowignored:	
\glstshortaccessdisplay: added check		new 200	
for existence 367		\@printgloss@checkexists@noignored:	
\glstshortaccsupp: new 364		new 200	
\glstshortplaccsupp: new 365		\@printglossary: replaced	
\glstshortpluralaccessdisplay:		\ifglossaryexists with	
added check for existence 367		\@printgloss@checkexists 202	
\glstshowaccsupp: new 7		\doifglossarynoexistsordo: switched	
\glstshowtargetfont: new 7		to starred form of	
\glstshowtargetouter: new 6		\ifglossaryexists 62	
\glstshowtargetsymbol: new 6		\ifglossaryexists: added starred form 60	
\glssymbolaccessdisplay: added		\s@ifglossaryexists: new 60	
check for existence 366			

\setglossarypreamble: switched to starred form of \ifglossaryexists	46	\glsdefs@newdocentry: new	80
4.47 (2021-09-20)		\glsdisplaynumberlist: changed	
\@@do@esc@wrglossary: changed \edef to \protected@edef	196	\edef to \protected@edef	165
\@@do@noesc@wrglossary: changed \edef to \protected@edef	193	\glsfielddef: changed \edef to \protected@edef	86
\@@do@write@glslabels: fully expand list	32	\glsfieldedef: changed \edef to \protected@edef	85
\@@gls@@default@acr@descplural: new	249	\glsfieldfetch: changed \edef to \protected@edef	86
\@@gls@@default@acr@firstplural: new	249	\glsfieldgdef: changed \edef to \protected@edef	85
\@@gls@@default@acr@plural: new	248	\glsfieldxdef: changed \edef to \protected@edef	85
\@@gls@noexpand@assign@default@plurals: new	248	\glsIfListOfAcronyms: changed \edef to \protected@edef	19
\@glo@do@sortentries: changed \edef to \protected@edef	206	\glsmoveentry: changed \edef to \protected@edef	95
\@glo@sortentries: changed \edef to \protected@edef	205	\glsnavhyperlink: changed \edef to \protected@edef	280
\@glo@storeentry: changed \edef to \protected@edef	96	\GlsSetQuote: added check for backslash	177
\@gls@checkesc: new	128	\glsstepsubentry: changed \edef to \protected@edef	217
\@gls@deprecated@acr@warning: new	263	\ifglsfieldcseq: changed \edef to \protected@edef	88
\@gls@link: changed \edef to \protected@edef	119	\ifglshaschildren: changed \edef to \protected@edef	62
\@gls@setupsort@use: changed \edef to \protected@edef	16	\ifignoredglossary: changed \edef to \protected@edef	69
\@gls@writedef: added a warning if entry has already been defined	80	\makenoidxglossaries: changed \edef to \protected@edef	185
\@write@glslabelsname: new	33	\new@glossaryentry: allow overwrite if defined in .glsdefs	78
General: changed \x to \@gls@x	227	\newignoredglossary: changed \edef to \protected@edef	69
\do@write@glslabelsname: new	33	\ns@newglossary: changed \edef to \protected@edef	67
\forglsentries: changed \edef to \protected@edef	59	writelabelnames: new	33
\gls@defdocnewglossaryentry: removed see warning	77	\SetDescriptionDUAAcronymStyle: added deprecated warning	253
\gls@defglossaryentry: changed \edef to \protected@edef	89, 90	\SetDescriptionFootnoteAcronymStyle: added deprecated warnings	251
\gls@docdef@addlabel: new	78	\SetDUASyle: added deprecated warnings	263
\gls@docdef@labellist: new	78	\SetFootnoteAcronymStyle: added deprecated warnings	258
\gls@glsdefs@addlabel: new	78	\SetSmallAcronymStyle: added deprecated warnings	261
\gls@glsdefs@labellist: new	78		
\gls@save@numberlist: changed \edef to \protected@edef	200		
\glsadd: changed \edef to \protected@edef	167		
\glsaddall: changed \edef to \protected@edef	167		

4.47 (??)		\glslistexpandedname:new	287
\@makeglossaries@warn@noprintglossary:		\glslistinit:new	287
new	184	\glsunexpandedfieldvalue:new	86
4.48 (2021-10-19)			
\glscapitalisewords:new	161		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	
\!	123, 126, 177, 481
\"	24, 123–126, 128, 129, 417, 478–481, 483
\#	128, 172, 507
\%	168, 174, 339, 340, 504, 508, 509
\&	41, 166, 427, 503
\'	24, 417
\.	12, 24, 409, 417
\=	24, 417
\?	123, 125, 177, 478, 480, 511
\@	79, 450
\@@delimN	226, 542
\@do@wrglossary	185, 193, 196, 517, 522, 523
\@do@esc@wrglossary	193, 522
\@do@noesc@wrglossary	193, 522
\@do@wrglossary	167, 191, 503, 521, 522
\@do@write@glslabels	32, 423
\@glo@assign@sortkey	186, 517, 536
\@glo@list	59, 439
\@glo@sort	24, 416
\@glo@type	201, 526
\@glossarysec	8, 48, 49, 408, 431, 432
\@glossaryseclabel	9, 48, 49, 215, 408, 431, 432, 535
\@glossarysecstar	9, 48, 215, 408, 431, 432, 535
\@gls@@default@acr@descplural	248
\@gls@@default@acr@firstplural	248
\@gls@@default@acr@plural	248
\@gls@checkactual	127, 128, 482, 483
\@gls@checkbar	126, 127, 481, 482
\@gls@checkescactual	125, 480
\@gls@checkescbar	125, 126, 480, 481
\@gls@checkesclevel	126, 481
\@gls@checkescquote	124, 178, 479, 480, 511, 512
\@gls@checklevel	127, 482
\@gls@checkquote	124, 176, 479, 510
\@gls@default@entryfmt	106, 115, 467, 473, 474
\@gls@expand@field	22, 76, 77, 82, 83, 248, 251, 253, 255, 257, 258, 260–263, 397–401, 415, 448, 449, 452, 556, 558, 559, 561, 563, 565, 567
\@gls@extramakeindexopts	176, 183, 510, 515
\@gls@fixbraces	198, 524
\@gls@noexpand@assign@default@plurals	248, 397
\@gls@noexpand@field	22, 75, 76, 415, 448
\@gls@noidx@no@sanitizesort	24, 416
\@gls@noidx@nosanitizesort	188, 416, 519
\@gls@noidx@sanitizesort	416
\@gls@nosanitizesort	23, 188, 416, 519
\@gls@sanitizesort	23, 188, 416, 519
\@gls@xdycheckbackslash	129, 483, 484
\@gls@xdycheckquote	128, 129, 483
\@glslocalreset	99, 101, 463, 465
\@glslocalunset	98, 101, 463, 465
\@glsreset	99, 101, 463–465
\@glsshowtarget	6, 7, 406, 407
\@glsunset	99, 101, 463, 465
\@newglossaryentry@defcounters	100, 464
\@this@glo@	59, 439
\@ACRfull	231, 544
\@ACRfullpl	232, 544
\@ACRlong	154, 231, 497, 544
\@ACRlongpl	156, 232, 498, 544
\@ACRshort	151, 231, 494, 544
\@ACRshortpl	152, 232, 495, 544
\@Acrfull	230, 543, 544
\@Acrfullpl	231, 232, 544
\@Acrlong	154, 230, 496, 544
\@Acrlongpl	156, 232, 497, 544

\@Acrshort	150, 494	\@Glsfirst	138, 488
\@Acrshortpl	152, 495	\@Glsfirst@	138, 488
\@Alpha	192, 195, 521–523	\@Glsfirstplural	140, 489
\@GLS	133, 486	\@Glsfirstplural@	140, 489
\@GLS@	133, 279, 486	\@Glsname	141, 489, 490
\@GLSdesc	142, 490	\@Glsname@	141, 490
\@GLSdesc@	142, 490	\@Glspl	134, 486
\@GLSdescplural	143, 490	\@Glspl@	102, 105, 134, 278, 465, 467, 486, 487
\@GLSdescplural@	143, 490	\@Glsplural	139, 489
\@GLSfirst	138, 139, 489	\@Glsplural@	139, 489
\@GLSfirst@	139, 489	\@Glsymbol	143, 491
\@GLSfirstplural	140, 489	\@Glsymbol@	143, 491
\@GLSfirstplural@	140, 489	\@Glsymbolplural	144, 491
\@GLSname	141, 490	\@Glsymbolplural@	144, 491
\@GLSname@	141, 490	\@Glstext	138, 488
\@GLSpl	135, 487	\@Glstext@	138, 488
\@GLSpl@	135, 279, 487	\@Glsuser@i	384, 385
\@GLSplural	139, 489	\@Glsuseri	145, 491
\@GLSplural@	139, 489	\@Glsuseri@	145, 491
\@GLSsymbol	143, 491	\@Glsuserii	146, 492
\@GLSsymbol@	143, 144, 491	\@Glsuserii@	146, 492
\@GLSsymbolplural	144, 491	\@Glsuseriii	146, 492
\@GLSsymbolplural@	144, 491	\@Glsuseriii@	146, 147, 492
\@GLStext	137, 488	\@Glsuseriv	147, 492
\@GLStext@	137, 138, 488	\@Glsuseriv@	147, 492
\@GLSuseri	145, 491	\@Glsuserv	148, 493
\@GLSuseri@	145, 491, 492	\@Glsuserv@	148, 493
\@GLSuserii	146, 492	\@Glsuservi	149, 493
\@GLSuserii@	146, 492	\@Glsuservi@	149, 493
\@GLSuseriii	147, 492	\@Mi	300
\@GLSuseriii@	147, 492	\@PGLS	278
\@GLSuseriv	147, 148, 493	\@PGLS@	278
\@GLSuseriv@	148, 493	\@PGLSpl	279
\@GLSuserv	148, 493	\@PGLSpl@	279
\@GLSuserv@	148, 493	\@Pgl	277
\@GLSuservi	149, 493	\@Pgl@	277
\@GLSuservi@	149, 493	\@Pglspl	277
\@Gls	132, 485	\@Pglspl@	277
\@Gls@	102, 104, 132, 277, 465, 466, 485	\@Roman	192, 195, 521, 522
\@Gls@acrentryname	233, 499, 545	\@acrfull	229, 543
\@Gls@entry@field	83, 158–164, 453, 498–502	\@acrfullpl	231, 544
\@Gls@entryname	158, 233, 499, 545	\@acrlong	153, 230, 496, 543
\@GlsSetXdyFirstLetterAfterDigits		\@acrlongpl	155, 231, 497, 544
	168, 169, 504	\@acrshort	149, 230, 493, 543, 544
\@GlsSetXdyNumberGroupOrder	169, 504, 505	\@acrshortpl	151, 231, 232, 495, 544
\@Glsdesc	141, 142, 490	\@addtoacronymlists	19, 413, 414
\@Glsdesc@	142, 490	\@afterheading	285, 289, 343, 572
\@Glsdescplural	142, 490	\@alpha	192, 195, 521–523
\@Glsdescplural@	142, 490	\@arabic	192, 195, 521, 522

\@auxout	66, 67, 103, 183, 185, 186, 188, 200, 204, 281, 442, 443, 466, 515, 517, 519, 525, 527, 528	\@empty	12, 16, 19, 34, 37–39, 50, 52, 55, 58, 59, 91, 96, 120, 121, 131–135, 150–156, 170, 173, 175, 180–182, 189, 191, 197, 223, 249, 251, 253, 254, 256, 258, 259, 261, 263, 265, 337, 339, 341, 378–380, 410, 412, 413, 423, 425–427, 433, 434, 436, 438, 439, 458, 461, 462, 476, 477, 485–487, 494–498, 505, 507, 509, 513, 514, 520, 524, 540, 556, 558, 560–564, 566, 567, 569
\@backslashchar	122, 123, 129, 177, 478, 483, 484	\@end@fixbraces	198, 524
\@bsphack	191, 520	\@endfortrue ...	29, 62, 81, 281, 420, 440, 451
\@cGls	103, 466	\@esphack	191, 521
\@cGls@	102, 104, 465, 466	\@expandtwoargs	37, 426
\@cGlspl	104, 467	\@firstofone	24, 32, 287, 416, 417, 422
\@cGlspl@	102, 104, 465, 467	\@firstofthree	117, 131, 134, 136, 150, 151, 153, 155, 378–380, 475, 485–487, 494–497
\@ccclv	300, 301	\@firstoftwo	27, 28, 80, 81, 117, 134, 135, 151–153, 155, 156, 419, 420, 450, 451, 475, 486, 487, 495–498
\@cgls	103, 466	\@for	29, 37, 39, 51, 52, 59, 80, 81, 122, 170–172, 182– 184, 192, 199, 205, 235, 249, 252, 254, 256, 258, 261, 263, 265, 281, 282, 337, 420, 426, 427, 433, 434, 439, 450, 451, 478, 505, 506, 515, 516, 521, 525, 528, 546, 556, 558, 560, 562, 563, 566, 568, 569
\@cgls@	101, 103, 465, 466	\@glo@desc	93, 460
\@cglspl	104, 467	\@glo@symbol	93, 460
\@cglspl@	101, 104, 465, 467	\@glo@access	357, 360, 365
\@chapter	39, 427	\@glo@addchildren ..	205, 210, 528, 529, 532
\@classoptionslist	37, 426	\@glo@assign@sortkey	184, 186, 216, 516, 517, 536
\@closegls	180, 513	\@glo@autosee	94, 460
\@colht	301	\@glo@autoseehook	94, 461
\@colroom	301	\@glo@check@mkidxrangechar	121, 122, 196, 336, 337, 477, 523
\@currentlabelname	9, 215, 408, 535	\@glo@check@sortallowed	14–17, 184, 188, 411, 412, 516, 519
\@curroptions	37, 426	\@glo@childlist	205, 528
\@declaredoptions	37, 426	\@glo@counter	72, 89, 93, 445, 457, 459
\@delimN	226, 542	\@glo@counterprefix ...	189, 193, 196, 197, 223, 224, 227, 519, 522–524, 540, 542
\@delimR	226, 541, 542	\@glo@default@sorttype	14, 186, 208, 209, 410, 411, 517, 530, 531
\@disable@onlypremakeg	183, 427, 516	\@glo@defaultcounter	92, 93, 459
\@disable@premakecs	39, 427	\@glo@desc	70, 88, 89, 91, 93, 445, 456–458, 460
\@disabled@glsgaddxdycounters	52, 433, 434	\@glo@descaccess	358, 360, 361
\@do@addcounter	51, 433	\@glo@descplural ...	70, 88, 89, 445, 456, 457
\@do@auxoutstuff	204, 527, 528		
\@do@glossentry	219, 355, 356, 538		
\@do@glsg@checkesc	123		
\@do@glsg@getcounterprefix	193, 196, 522, 523		
\@do@glsg@islistofacronyms	19, 414		
\@do@glsgsee	94, 461		
\@do@ifinlist	50, 433		
\@do@newglossaryentry ..	233, 247, 248, 250–255, 257, 258, 260–263, 265, 397– 401, 545, 555–559, 561–563, 565–567, 569		
\@do@seeglossary ...	185, 198, 517, 524, 525		
\@do@subglossentry	221, 356, 539		
\@do@wrglossary	120, 476, 521		
\@do@write@glslabels	33, 423		
\@do@writeaux@info	200, 525		
\@domakeglossaries	31, 32, 182, 185, 422, 515, 516		
\@ehc	300		

\@glo@descpluralaccess	358, 360, 361	\@glo@sort	14, 17, 23, 24, 70, 89, 92,
\@glo@do@sortentries	205, 528, 529		96, 97, 411, 412, 416, 445, 457, 459, 461, 462
\@glo@entry	167, 504	\@glo@sortedinsert	206, 529
\@glo@entryprefix	273	\@glo@sortentries ..	208, 209, 528, 530, 531
\@glo@entryprefixfirst	273	\@glo@sorthandler@case	209, 530, 531
\@glo@entryprefixfirstplural ..	273, 274	\@glo@sorthandler@letter ...	208, 529, 530
\@glo@entryprefixplural	273	\@glo@sorthandler@nocase ...	209, 530, 531
\@glo@esclabel	96, 97, 461, 462	\@glo@sorthandler@word	208, 529, 530
\@glo@etext	107–109, 468, 469	\@glo@sortinghandler ...	205, 206, 528, 529
\@glo@first	71,	\@glo@sortinglist	
	89, 92, 93, 260, 400, 445, 457, 459, 460, 565		205, 206, 209, 528, 529, 531, 532
\@glo@firstaccess	357, 360	\@glo@sortmacro@case	531
\@glo@firstplural		\@glo@sortmacro@def	531
	71, 89, 92, 401, 445, 457, 459	\@glo@sortmacro@def@do	532
\@glo@firstpluralaccess	357, 360	\@glo@sortmacro@letter	530
\@glo@grabfirst	211, 533	\@glo@sortmacro@nocase	531
\@glo@label	74,	\@glo@sortmacro@standard	531
	75, 82, 83, 85–94, 100, 165, 166, 273,	\@glo@sortmacro@use	532
	274, 331, 360, 361, 447, 452–461, 464, 503	\@glo@sortmacro@word	530
\@glo@list	94, 461	\@glo@sorttype	
\@glo@long ..	63, 74, 90, 93, 441, 447, 457, 459		186, 210, 211, 216, 517, 532, 533, 536
\@glo@longaccess	358, 360, 361	\@glo@storeentry	14–16, 411, 412, 461
\@glo@longpl	74, 90, 93, 248–	\@glo@suffix ..	121, 122, 196, 337, 477, 478, 523
	250, 252, 255, 257, 260, 262, 397, 447,	\@glo@symbol	63, 71, 89, 93, 254,
	457, 459, 555–557, 559, 561, 563, 565, 567		259, 260, 441, 445, 457, 460, 560, 564, 565
\@glo@longpluralaccess	358, 360, 361	\@glo@symbolaccess	357, 360, 400
\@glo@name	14,	\@glo@symbolplural ..	71, 89, 94, 445, 457, 460
	70, 75, 89, 91–93, 411, 445, 448, 457–459	\@glo@symbolpluralaccess	357, 360
\@glo@no@assign@sortkey	184, 516, 536	\@glo@text	70, 89,
\@glo@nonumberlist	73, 446		92, 93, 132–136, 157–159, 255, 274, 399,
\@glo@numfmt	197, 337, 523		445, 457, 459, 460, 485–488, 498, 499, 561
\@glo@parent	16, 72, 89, 91, 92,	\@glo@textaccess	357, 360, 397–399
	96, 97, 206, 412, 446, 457–459, 461, 462, 529	\@glo@thislabel	95, 461
\@glo@plural ..	70, 89, 91, 92, 399, 445, 457–459	\@glo@thislettergrp	211, 212, 532–534
\@glo@pluralaccess	357, 360	\@glo@thisvalue	64, 65, 441, 442
\@glo@prefix	10,	\@glo@tmp	82, 83, 197, 452, 453, 524
	72, 73, 89, 96, 97, 121, 122, 196, 336,	\@glo@type	9, 16, 71, 89–94, 166, 167,
	337, 409, 446, 457, 461, 462, 477, 478, 523		183, 189, 201–206, 209–211, 214, 215,
\@glo@range	196, 336, 337, 523		233, 249, 251, 253, 256, 258, 261, 263,
\@glo@see	72, 89, 94, 446, 457, 461		265, 280, 282, 408, 412, 445, 457–460,
\@glo@seeautonumberlist ..	10, 72, 409, 446		503, 504, 515, 520, 526–529, 532, 535,
\@glo@short			545, 556, 558, 560, 561, 563, 566, 567, 569
	64, 74, 90, 93, 400, 441, 447, 457, 459	\@glo@types	32,
\@glo@shortaccess ..	358, 360, 361, 397–400		59, 67, 99, 100, 167, 182–184, 271, 331,
\@glo@shortpl	74, 90, 93, 247–250,		423, 439, 442, 443, 464, 504, 515, 516, 571
	252, 255, 257, 260, 262, 397, 400, 447,	\@glo@useri	73, 90, 93, 447, 457, 459
	457, 459, 555, 557, 559, 561, 563, 565, 567	\@glo@useriaccess	358, 360, 361
\@glo@shortpluralaccess	358, 360, 361	\@glo@userii	73, 90, 93, 447, 457, 459
		\@glo@useriaccess	358, 360, 361

<code>\@glo@useriii</code>	74, 90, 93, 447, 457, 459	<code>\@gls@classI</code>	170, 171, 505, 506
<code>\@glo@useriiiaccess</code>	358, 360, 361	<code>\@gls@classII</code>	171, 506
<code>\@glo@useriv</code>	74, 90, 93, 447, 457, 459	<code>\@gls@codepage</code>	204, 438, 528
<code>\@glo@userivaccess</code>	359–361	<code>\@gls@counter</code>	
<code>\@glo@userv</code>	74, 90, 93, 447, 457, 459	116, 119–121, 166, 167, 189, 196,
<code>\@glo@uservaccess</code>	359–361		197, 337, 474, 476, 477, 503, 519, 523, 524
<code>\@glo@uservi</code>	74, 90, 93, 447, 457, 459	<code>\@gls@counterwithin</code>	12, 13, 410
<code>\@glo@userviaccess</code>	359–361	<code>\@gls@ctr</code>	51, 433
<code>\@glodesc</code>	93, 460	<code>\@gls@currentlettergroup</code>	210, 212, 532–534
<code>\@glolist@</code>	91, 458	<code>\@gls@debugfalse</code>	5, 406
<code>\@gloname</code>	93, 459	<code>\@gls@debugtrue</code>	5, 6, 406
<code>\@glossary@default@style</code>	9, 11,	<code>\@gls@declareoption</code>	
	202, 224, 266, 408, 409, 526, 540, 541, 569	10, 11, 17, 18, 21, 22, 27, 30–
<code>\@glossaryentryfield</code>	96, 461, 462		33, 36, 37, 409, 413, 415, 419, 421–423, 425
<code>\@glossarysection</code>	47, 430, 431	<code>\@gls@default</code>	105, 467
<code>\@glossarystyle</code> ..	202, 203, 214, 526, 527, 535	<code>\@gls@default@value</code>	
<code>\@glossarysubentryfield</code> ..	96, 97, 461, 462	63–65, 75, 76, 89, 91–93, 259,
<code>\@gls</code>	131, 485		273, 441, 442, 445, 448, 449, 457–460, 564
<code>\@gls@</code> ...	101, 103, 131, 276, 277, 465, 466, 485	<code>\@gls@deffile</code>	78–80, 450, 451
<code>\@gls@@automake@immediate</code>	183, 515	<code>\@gls@define@glossaryentrycounter</code> ..	
<code>\@gls@@link</code>	118, 475	13, 38, 215, 217, 410, 426, 536
<code>\@gls@Hcounter</code>	120, 121, 476, 477	<code>\@gls@define@glossarysubentrycounter</code>	
<code>\@gls@ReturnAfterFi</code>	227, 542	38, 216, 217, 410, 426, 536
<code>\@gls@actualchar</code>	96,	<code>\@gls@defsort</code>	14–16, 93, 411, 412, 460
	97, 125, 127, 174, 340, 462, 480, 482, 508	<code>\@gls@defsortcount</code> ..	14–16, 68, 411, 412, 443
<code>\@gls@addpredefinedattributes</code>		<code>\@gls@deprecated@acr@warning</code>	
.....	170, 179, 434, 435, 505, 512	251, 253, 255, 258, 261, 263
<code>\@gls@adjustmode</code>	166, 503, 504	<code>\@gls@do@acronymsdef</code>	18, 38, 69, 413, 426, 444
<code>\@gls@after</code>	19, 414	<code>\@gls@do@indexdef</code> ..	37–39, 69, 425–427, 444
<code>\@gls@automake</code>	184, 513, 516	<code>\@gls@do@numbersdef</code>	36–38, 69, 425, 426, 444
<code>\@gls@automake@immediate</code> ...	182, 514, 515	<code>\@gls@do@symbolsdef</code>	36, 69, 425, 444
<code>\@gls@before</code>	19, 414	<code>\@gls@do@symbolssdef</code>	38, 39, 426, 427
<code>\@gls@between</code>	282	<code>\@gls@doautomake</code>	
<code>\@gls@body</code>	158, 159, 499	33, 34, 182, 184, 423, 515, 516
<code>\@gls@checkactual</code>	123, 177, 479, 482, 483, 511	<code>\@gls@docheckquotedef</code> ..	176–178, 510–512
<code>\@gls@checkbar</code> ..	123, 177, 479, 481, 482, 511	<code>\@gls@docloadedfalse</code>	5, 405
<code>\@gls@checkedmkidx</code>		<code>\@gls@docloadedtrue</code>	4, 5, 405
.....	122–129, 176–178, 478–484, 510–512	<code>\@gls@dodeflistparser</code>	184, 516
<code>\@gls@checkesc</code>	123, 177	<code>\@gls@doentrycounterdef</code> ..	38, 39, 426, 427
<code>\@gls@checkescactual</code>	123, 177, 478, 480, 511	<code>\@gls@doentrydef</code>	115, 116, 473, 474
<code>\@gls@checkescbar</code> ..	123, 177, 479–481, 511	<code>\@gls@dolast</code>	198, 199, 525
<code>\@gls@checkesclevel</code>	123, 177, 481	<code>\@gls@donext</code>	199, 525
<code>\@gls@checkescquote</code>		<code>\@gls@donext@def</code>	165, 503
.....	123, 177, 178, 478–480, 510–512	<code>\@gls@dosubentrycounterdef</code>	38, 39, 426, 427
<code>\@gls@checklevel</code> ...	123, 177, 479, 482, 511	<code>\@gls@dothiswrite</code>	180–182, 513–515
<code>\@gls@checkmkidxchars</code>		<code>\@gls@elem</code>	281
.....	96, 121, 177, 185, 196, 198,	<code>\@gls@enablesavenonumberlist</code>	79, 446, 450
	336, 337, 461, 477, 478, 510, 517, 523, 524	<code>\@gls@encapchar</code>	125, 126, 174,
<code>\@gls@checkquote</code>	123, 176, 177, 478, 479, 510		197, 198, 337, 340, 480–482, 508, 523, 524

<code>\@gls@entry@count</code>	102, 103, 466	<code>\@gls@link@nocheckfirsthyper</code>	
<code>\@gls@entry@field</code>	82, 83, 101,		137, 150–156, 488, 494–498
	158–165, 361–363, 452, 453, 464, 498–502	<code>\@gls@link@opts</code> .	119, 250, 257, 476, 557, 562
<code>\@gls@escbsdq</code>	123, 175, 341, 478, 509	<code>\@gls@list</code>	281, 282
<code>\@gls@expand@fields</code>	76, 77, 448, 449	<code>\@gls@listsuffix</code>	50, 433
<code>\@gls@expandonce</code>	77, 449	<code>\@gls@loadlist</code>	11, 266, 409, 569
<code>\@gls@extramakeindexopts</code> ...	183, 512, 515	<code>\@gls@loadlong</code>	11, 266, 409, 569
<code>\@gls@fetchfield</code>	64, 442, 451	<code>\@gls@loadsuper</code>	11, 266, 409, 569
<code>\@gls@field@link</code>		<code>\@gls@loadtree</code>	11, 266, 409, 569
	. 84, 137–149, 380–385, 453, 454, 488–493	<code>\@gls@local@increment@currcount</code> ...	
<code>\@gls@fieldaccess@display</code>	365–368		101, 465, 466
<code>\@gls@firsttok</code>	211, 532, 533	<code>\@gls@loclist</code>	
<code>\@gls@fixbraces</code>	94, 461, 524		186, 187, 211, 212, 517, 518, 533, 534
<code>\@gls@forbidtexext</code>	67, 443	<code>\@gls@map</code>	80, 81, 450, 451
<code>\@gls@fulllist</code>	32	<code>\@gls@missinglang@warn</code> ...	21, 42, 415, 428
<code>\@gls@get@counterprefix</code>	197, 524	<code>\@gls@missingnumberlist</code>	93, 448, 460
<code>\@gls@getbody</code>	158, 499	<code>\@gls@noaccess</code>	365
<code>\@gls@getcounter</code>	444	<code>\@gls@noexpand@fields</code>	77, 448, 449
<code>\@gls@getcounterprefix</code> .	193, 196, 522–524	<code>\@gls@nohyperlist</code> .	20, 69, 119, 414, 444, 476
<code>\@gls@getgrouptitle</code>		<code>\@gls@noidx@do</code>	210, 532, 533
	185, 222, 282, 517, 539, 540	<code>\@gls@noidx@getgrouptitle</code> ..	185, 517, 540
<code>\@gls@glossary</code>	190, 191, 520	<code>\@gls@noidx@sanitizesort</code> .	23, 188, 416, 519
<code>\@gls@gobbleopt</code>	66, 443	<code>\@gls@noidx@setsanitizesort</code>	
<code>\@gls@grptitle</code>	222, 280, 282, 539		26, 188, 418, 519
<code>\@gls@hyp@opt</code>	84, 103,	<code>\@gls@noidx@loclist@finalsep</code> ...	187, 518
	104, 118, 131–156, 229–232, 275–279,	<code>\@gls@noidx@loclist@prev</code> .	187, 213, 518, 534
	453, 454, 466, 467, 475, 485–498, 543, 544	<code>\@gls@noidx@loclist@sep</code> .	187, 213, 518, 534
<code>\@gls@hyp@opt@cs</code>	117, 475	<code>\@gls@noref@warn</code> ...	185, 211, 517, 519, 532
<code>\@gls@hypergroup</code>	281	<code>\@gls@numberlink</code>	226, 227, 542
<code>\@gls@ifinlist</code>	51, 433	<code>\@gls@numbersdef</code>	36, 425
<code>\@gls@ifnotmeasuring</code>	97, 462	<code>\@gls@numlist@lastsep</code>	165, 166, 503
<code>\@gls@igtype</code>	69, 445	<code>\@gls@numlist@nextsep</code>	165, 503
<code>\@gls@increment@currcount</code> ..	101, 465, 466	<code>\@gls@numlist@sep</code>	165, 166, 503
<code>\@gls@indexdef</code>	37, 425	<code>\@gls@old@chapter</code>	39, 427
<code>\@gls@initnonumberlist</code> ...	73, 90, 446, 457	<code>\@gls@oldnewglossaryentryposthook</code> .	360
<code>\@gls@islistofacronyms</code>	19, 414	<code>\@gls@oldnewglossaryentryprehook</code> ..	
<code>\@gls@keylist</code>	396		359, 360
<code>\@gls@keymap</code> .	73, 80–83, 273, 359, 447, 450–452	<code>\@gls@onlypremakeg</code>	39, 427
<code>\@gls@label</code>		<code>\@gls@order</code>	180–182, 513, 514
	185, 189, 193, 196, 197, 517, 519, 522, 523	<code>\@gls@org@LT@output</code>	300
<code>\@gls@labelsfile</code>	32, 33, 423	<code>\@gls@org@glsnoidx@displayloc</code> .	187, 518, 519
<code>\@gls@langmod</code>	180, 181, 513, 514	<code>\@gls@org@glssseeformat</code>	187, 518, 519
<code>\@gls@levelchar</code>		<code>\@gls@override@glossary</code> ..	35, 36, 424, 425
	. 97, 126, 127, 174, 340, 462, 481, 482, 508	<code>\@gls@patchtabularx</code>	97, 462
<code>\@gls@link</code>	118, 132–137, 150–	<code>\@gls@preglossaryhook</code>	203, 526, 527
	157, 378–380, 475, 476, 485–488, 494–498	<code>\@gls@prevlevel</code> .	311, 312, 331–334, 349, 350
<code>\@gls@link@checkfirsthyper</code>		<code>\@gls@provide@newglossary</code> ...	67, 442, 443
	131–136, 475, 485–487	<code>\@gls@quotechar</code>	124–
<code>\@gls@link@label</code> .	119, 250, 257, 476, 557, 562		127, 174, 176, 178, 340, 479–482, 508–512

<code>\@gls@reference</code>	. 185, 186, 189, 517, 519, 534	<code>\@gls@usetranslator</code>	.. 28, 41, 419, 420, 428
<code>\@gls@removespaces</code> 227, 542	<code>\@gls@value</code> 76, 161, 448, 449, 500
<code>\@gls@renewglossary</code> 179, 512, 520	<code>\@gls@warnonglossdefined</code>	22, 200, 415, 525
<code>\@gls@replacementtext</code> 364, 365	<code>\@gls@warnontheGLOSSdefined</code>
<code>\@gls@rest</code> 158, 499	22, 219, 415, 538
<code>\@gls@restoreat</code> 79, 450	<code>\@gls@write@entrycounts</code> 102, 466
<code>\@gls@roman</code> 54, 337, 338, 435	<code>\@gls@writedef</code> 79, 450
<code>\@gls@sanitized@tmp</code> 122, 478	<code>\@gls@writeisthook</code> 174, 175, 508, 509
<code>\@gls@sanitizedesc</code> 29, 415, 421	<code>\@gls@x</code> 227
<code>\@gls@sanitizename</code> 416	<code>\@gls@xdy@locationlist</code>
<code>\@gls@sanitizesort</code> 14, 411, 416	170, 171, 435, 505, 506
<code>\@gls@sanitizesymbol</code> 29, 30, 416, 421	<code>\@gls@xdycheckbackslash</code>	122, 478, 483, 484
<code>\@gls@saveentrycounter</code>	. 120, 167, 476, 503	<code>\@gls@xdycheckquote</code> 122, 478, 483
<code>\@gls@savenonumberlist</code> 73, 446	<code>\@gls@xref</code> 198, 524
<code>\@gls@see@noindex</code> 8, 72, 407, 408, 446	<code>\@gls@Alphacompositor</code>	. 45, 54, 338, 430, 436
<code>\@gls@setacstyle</code> 29, 30, 38, 421, 426	<code>\@gls@Hlocref</code> 193, 196, 522, 523
<code>\@gls@setcounter</code> 68, 443, 444	<code>\@gls@acronymlists</code>
<code>\@gls@setdefault@glslink@opts</code>	. 119, 476	19, 20, 59, 233–235, 249, 251–254, 256,
<code>\@gls@setsort</code>	258, 261, 263, 265, 270, 413, 414, 439,
.....	14–17, 120, 167, 411, 412, 476, 503	545, 546, 556, 558, 560–563, 566–569, 571
<code>\@gls@setupshortcuts</code> 38, 426	<code>\@gls@addkey</code> 83, 452
<code>\@gls@setupsort@def</code> 411	<code>\@gls@addstoragekey</code> 82, 452
<code>\@gls@setupsort@none</code> 412	<code>\@gls@addxdyattribute</code> 51, 52, 433, 434
<code>\@gls@setupsort@standard</code> 411	<code>\@gls@defaultplural</code> 447
<code>\@gls@setupsort@use</code> 412	<code>\@gls@defaultsort</code> 14, 411, 448
<code>\@gls@sort</code> 212, 533	<code>\@gls@desc</code> 141, 490
<code>\@gls@sort@A</code> 207, 529, 530	<code>\@gls@desc@</code> 141, 490
<code>\@gls@sort@B</code> 207, 529, 530	<code>\@gls@descplural</code> 142, 490
<code>\@gls@startswithexpandonce</code> 76, 449	<code>\@gls@descplural@</code> 142, 490
<code>\@gls@storenonumberlist</code>	.. 73, 93, 446, 460	<code>\@gls@disp</code> 136, 487
<code>\@gls@symbolsdef</code> 36, 425	<code>\@gls@entry</code>	. 32, 33, 99, 100, 103, 423, 464, 466
<code>\@gls@this</code> 192, 521	<code>\@gls@entrytitlecase</code> 161, 500, 501
<code>\@gls@thisHloc</code> 197, 524	<code>\@gls@first</code> 138, 488
<code>\@gls@thisentryname</code> 32, 33	<code>\@gls@first@</code> 138, 488
<code>\@gls@thisfield</code> 64, 65, 442	<code>\@gls@firstletter</code> 169, 504
<code>\@gls@thislabel</code>	.. 62, 199, 209, 440, 525, 531	<code>\@gls@firstplural</code> 140, 489
<code>\@gls@thislist</code> 165, 166, 503	<code>\@gls@firstplural@</code> 140, 381, 489
<code>\@gls@thisloc</code> 197, 524	<code>\@gls@hypernumber</code> 226, 541, 542
<code>\@gls@thisval</code> 81, 451	<code>\@gls@isacronymlistfalse</code> 20, 414
<code>\@gls@title</code> 47, 430	<code>\@gls@isacronymlisttrue</code> 20, 414
<code>\@gls@tmp</code>	... 7, 16, 42, 54, 55, 77, 122, 191,	<code>\@gls@link</code>
.....	281, 282, 407, 412, 428, 436, 449, 478, 520	120, 130, 131, 166, 280, 476, 484, 485, 503
<code>\@gls@tmpb</code>	124–129, 176, 178, 479–484, 510–512	<code>\@gls@localreset</code> 98, 101, 463, 465
<code>\@gls@toc</code> 48, 431, 432	<code>\@gls@localunset</code> 98, 101, 463, 465
<code>\@gls@type</code>	... 182, 184, 235, 249, 252, 254,	<code>\@gls@locref</code>
.....	256, 258, 261, 263, 265, 331, 515, 516,	189, 193, 196, 197, 336, 337, 519, 522, 523
.....	546, 556, 558, 560, 562, 563, 566, 568, 569	<code>\@gls@minrange</code> 169, 171, 338, 505, 506
<code>\@gls@updatechecked</code>	<code>\@gls@name</code> 140, 489
.....	122, 123, 177, 478, 479, 510, 511	<code>\@gls@name@</code> 140, 141, 489

\@glsnavhypertarget	280	\@ifglossaryexists	60, 200, 439, 525
\@glsnextpages	202, 527, 536	\@ifl@t@r	25, 417
\@glsnodelsc	89, 91, 93, 447, 457, 458, 460	\@ifnextchar	68, 117, 443, 475
\@glsnoname	89, 91, 93, 447, 457–459	\@ifpackageloaded	...
\@glsnonextpages	202, 527, 536	.. 4, 9, 27–29, 41, 58, 97, 165, 176, 355,	
\@glsnumberformat	116, 119, 166, 189,	405, 408, 419, 420, 428, 438, 462, 502, 509	
196, 197, 336, 337, 474, 476, 503, 519, 523		\@ifstar	60, 67, 82, 83, 117,
\@glsopenfile	179, 189, 512, 513, 520	168, 169, 228, 439, 443, 452, 475, 504, 543	
\@glsorder	183, 421, 515	\@ifundefined	41, 281, 290,
\@glspl	134, 486	291, 301, 312, 313, 319, 332, 333, 349, 428	
\@glspl@	102, 104, 134, 276, 278, 465, 467, 486	\@ignored@glossaries	32, 69, 423, 444, 445
\@glsplural	139, 489	\@input@	203, 527
\@glsplural@	139, 489	\@istfilename	183, 430, 515
\@glsprefix@record@hook	276–279	\@makecol	300, 301
\@glsreset	98, 101, 463, 465	\@makeglossaries@warn@noprintglossary	184
\@glssee	94, 198, 461, 525	
\@glsshowaccsupp	6, 357, 406, 407	\@makeglossary	31, 32, 183, 422, 512, 515, 516
\@glsshowtarget	5, 6, 130, 406, 407, 484	\@minus	285, 286, 307, 325, 572
\@glsymbol	143, 490	\@mkboth	47, 48, 431
\@glsymbol@	143, 491	\@newglossary	66, 67, 442–444
\@glsymbolplural	144, 491	\@newglossaryentry@defcounters	94, 100, 460, 461, 464
\@glsymbolplural@	144, 491	
\@glstarget	130, 131, 219, 281, 484, 485, 538	\@newglossaryentryposthook	82, 83, 94, 273, 360, 452, 460, 461
\@glstext	137, 488	
\@glstext@	137, 488	\@newglossaryentryprehook	82,
\@glsunset	98, 101, 463, 465	83, 88, 90, 273, 359, 452, 456, 457, 461	
\@glsuseri	145, 491	\@nil	19, 94,
\@glsuseri@	145, 491	121–123, 158, 177, 196, 198, 211, 212,	
\@glsuserii	145, 492	226, 227, 336, 337, 414, 461, 477–479,	
\@glsuseriii	145, 492	499, 510, 511, 523, 524, 532, 533, 541, 542	
\@glsuseriii@	146, 492	\@nnil	19, 199, 414, 525
\@glsuseriv	147, 492	\@no@makeglossaries	31, 32, 184, 186, 422, 516, 517, 519
\@glsuseriv@	147, 492	
\@glsuserv	148, 493	\@no@post@desc	342, 429
\@glsuserv@	148, 493	\@nopostdesc	202, 429, 527
\@glsuservi	149, 493	\@onelevel@sanitize	7, 23, 33, 54, 80, 96, 122, 169, 173,
\@glsuservi@	149, 493	198, 200, 211, 338, 339, 407, 416, 435,	
\@glswidestname	331–333, 349	450, 462, 478, 504, 505, 507, 524, 525, 532	
\@glswritefiles	34, 423, 520	\@onlypreamble	68, 78, 88, 89, 102, 105, 184,
\@glstr@doaccsupp	355	188, 444, 450, 456, 457, 466, 467, 516, 519	
\@glstr@record	275	\@onlypremakeg	44, 45, 51, 52, 56, 68,
\@gobble	5, 6,	175, 427, 429, 430, 433, 434, 437, 444, 509	
14–16, 31–33, 80, 97, 122, 168, 172, 183,		\@org@glossaryentrynumbers	202, 203, 526, 527
185, 287, 335, 339, 340, 406, 411, 412,		
422, 450, 451, 462, 478, 504, 507, 516, 517		\@org@gl@assign@descplural	248, 257, 258,
\@gobblethree	6, 406	
\@idxitem	326	260–263, 397, 400, 401, 556, 563, 565–567	
\@ifclassloaded	5, 12, 47, 405, 410, 431		

<code>\@org@gl@assign@firstpl</code>	<code>\@tmp</code>	338, 435
.... 248, 251, 253, 255, 257, 258, 260–	<code>\@use@ption</code>	37, 426
263, 397–401, 556–559, 561, 563, 565, 567	<code>\@warn@nomakeglossaries</code>	182, 204, 515, 528
<code>\@org@gl@assign@plural</code>	<code>\@wrglossary@pageformat</code>	192, 521, 522
. 248, 251, 253, 255, 257, 258, 260–263,	<code>\@wrglossarynumberhook</code> .	192, 195, 521–523
397–401, 556, 558, 559, 561, 563, 565–567	<code>\@write@gl@labelsname</code>	33
<code>\@org@gl@assign@symbolplural</code>	<code>\@xdy@main@language</code>	
..... 248, 251, 253, 255, 260–263, 30, 180, 181, 204, 421, 513, 514, 527	
397–399, 401, 556, 558, 559, 561, 565–567	<code>\@xdy@attributelist</code> ..	52, 172, 433, 434, 506
<code>\@org@gl@numberformat</code>	<code>\@xdy@attributes</code>	
165, 503 51, 170, 335, 337, 433, 434, 505	
<code>\@org@newglossaryentryprehook</code> ..	<code>\@xdy@counters</code>	50–52, 172, 433, 434, 506
88, 456	<code>\@xdy@crossrefhook</code>	171, 436, 506
<code>\@outputpage</code>	<code>\@xdy@language</code>	204, 438, 527, 528
300, 301	<code>\@xdy@lettergroups</code>	58, 173, 340, 438, 439, 508
<code>\@p@glossarysection</code>	<code>\@xdy@locationclassorder</code>	
47, 430, 431 56, 172, 339, 437, 506	
<code>\@pgls</code>	<code>\@xdy@locref</code> ..	51, 173, 335, 339, 433, 434, 507
275	<code>\@xdynumbergrouporder</code>	58, 169, 438, 504, 505
<code>\@pgls@</code>	<code>\@xdy@requiredstyles</code>	
275, 276 57, 170, 337, 437, 438, 505	
<code>\@pglspl</code>	<code>\@xdysortrules</code>	56, 174, 340, 437, 508
276	<code>\@xdystyle</code>	170, 337, 505
<code>\@pglspl@</code>	<code>\@xdy@useralphabets</code> ..	53, 170, 337, 435, 505
276	<code>\@xdy@userlocationdefs</code>	
<code>\@plus</code> 55, 171, 336, 338, 436, 506	
285, 286, 307, 325, 572	<code>\@xdy@userlocationnames</code>	55, 56, 336, 436, 437
<code>\@print@glossary</code>	<code>\@xfor@nextelement</code>	199, 525
201, 526, 527	<code>\@</code>	95, 122, 128, 168,
<code>\@print@noidx@glossary</code>	174, 175, 226, 227, 340, 341, 343–345,	
201, 526, 532	353, 354, 461, 478, 504, 508, 509, 541, 542	
<code>\@printgloss@checkexists</code> ...	<code>\{</code>	80, 168, 175, 335, 340, 341, 450, 504, 509
201, 202, 526	<code>\}</code>	80, 168, 175, 335, 341, 450, 451, 504, 509
<code>\@printgloss@checkexists@allowignored</code>	<code>\~</code>	24, 33, 417
..... 526	<code>\‘</code>	24, 417
<code>\@printgloss@checkexists@noignored</code>	<code>\ </code>	123, 125, 128, 177, 479–481, 511
..... 200, 201, 525, 526	<code>\~</code>	24, 417
<code>\@printgloss@setsort</code>		
..... 184, 186, 202, 516, 517, 526		
<code>\@printglossary</code>		
201, 526		
<code>\@roman</code>		
54, 337, 435		
<code>\@secondofthree</code> .		
117, 131, 132, 135, 150,		
152, 154, 156, 378, 475, 485, 487, 494–497		
<code>\@secondoftwo</code>		
24, 27, 28,		
42, 80, 81, 130–133, 136, 150, 151, 153,		
154, 287, 378–380, 404, 416, 417, 419,		
420, 428, 450, 451, 484–487, 494, 496, 497		
<code>\@set@glo@numformat</code>		
197, 337, 477, 523		
<code>\@sglsaddkey</code>		
83, 452		
<code>\@sglsaddstoragekey</code>		
82, 452		
<code>\@tabacckludge</code>		
24, 417		
<code>\@text@composite@x</code>		
24, 417		
<code>\@thirdofthree</code> ..		
117, 133, 135, 151, 153,		
154, 156, 379, 475, 486, 487, 494, 496–498		
<code>\@this@attr</code>		
172, 506, 507		
<code>\@this@childlabel</code>		
205, 206, 528, 529		
<code>\@this@counter</code>		
52, 434		
<code>\@this@ctr</code>		
172, 506, 507		
<code>\@this@key</code>		
81, 451		
<code>\@this@label</code>		
205, 528		
<code>\@this@cs</code>		
39, 427		

A

<code>\a</code>	24, 417
<code>\AA</code>	25, 417
<code>\aa</code>	25, 417
<code>\Ac</code>	555
<code>\ac</code>	555
<code>accsupp package</code>	355, 356
<code>\accsuppglossaryentryfield</code>	355
<code>\accsuppglossarysubentryfield</code>	356
<code>\Acf</code>	555
<code>\acf</code>	555

<code>\Acfp</code>	555	<code>\acronymtype</code>	18, 233, 235, 247, 249–253, 255–258, 260–263, 265, 397– 400, 413, 545, 546, 555–563, 565–567, 569
<code>\acfp</code>	555	<code>\acrpluralsuffix</code>	233, 236–238, 242–245, 248–250, 252, 255–258, 260– 262, 265, 388, 393–395, 397–401, 543, 545, 547–549, 552–554, 556–559, 561–569
<code>\Acl</code>	555	<code>\ACRshort</code>	494
<code>\acl</code>	555	<code>\Acrshort</code>	246, 494, 555
<code>\Aclp</code>	555	<code>\acrshort</code>	246, 493, 555
<code>\aclp</code>	555	<code>\ACRshorttpl</code>	495
<code>\Acp</code>	555	<code>\Acrshorttpl</code>	246, 495, 555
<code>\acp</code>	555	<code>\acrshorttpl</code>	246, 495, 555
<code>\acrfootnote</code>	250, 257, 556, 557, 562	<code>\Acs</code>	555
<code>\ACRfull</code>	544	<code>\acs</code>	555
<code>\Acrfull</code>	246, 543, 555	<code>\Acsp</code>	555
<code>\acrfull</code>	246, 543, 555	<code>\acsp</code>	555
<code>\ACRfullfmt</code>	231, 234, 242, 244, 392, 394, 544, 545, 552, 553	<code>\add@accent@</code>	24, 416
<code>\Acrfullfmt</code>	230, 234, 242, 244, 392, 394, 544, 545, 552, 553	<code>\addcontentsline</code>	49, 432
<code>\acrfullfmt</code>	230, 234, 242, 243, 392, 394, 543, 545, 552, 553	<code>\addglossarytocaptions</code>	42, 428
<code>\acrfullformat</code>	164, 230, 247, 248, 264, 502, 543, 555, 568	<code>\addtolength</code>	333, 349
<code>\ACRfullpl</code>	544	<code>\advance</code>	15, 16, 91, 120, 300, 411, 412, 458, 477
<code>\Acrfullpl</code>	247, 544, 555	<code>\AE</code>	24, 417
<code>\acrfullpl</code>	246, 544, 555	<code>\ae</code>	24, 417
<code>\ACRfullplfmt</code>	232, 234, 242, 244, 392, 394, 544, 546, 552, 553	<code>\altnewglossary</code>	444
<code>\Acrfullplfmt</code>	232, 234, 242, 244, 392, 394, 544, 546, 552, 553	<code>amsgen package</code>	<u>4</u> , <u>116</u>
<code>\acrfullplfmt</code>	231, 234, 242, 244, 392, 394, 544, 545, 552, 553	<code>amsmath package</code>	<u>97</u>
<code>\acrlinkfootnote</code>	249, 556	<code>\andname</code>	199, 427, 525
<code>\acrlinkfullformat</code>	230–232, 543, 544	<code>\AnyTrackedLanguages</code>	42, 404, 428
<code>\ACRlong</code>	496	<code>\appto</code>	20, 25, 73, 82, 83, 273, 359, 414, 417, 447, 452
<code>\Acrlong</code>	246, 496, 555	<code>array package</code>	<u>297</u> , <u>301</u> , <u>319</u>
<code>\acrlong</code>	246, 496, 555	<code>article class</code>	<u>197</u>
<code>\ACRlongpl</code>	498	<code>\AtBeginDocument</code>	18, 58, 79, 97, 167, 186, 413, 438, 450, 462, 504, 517
<code>\Acrlongpl</code>	246, 497, 555	<code>\AtEndDocument</code>	32, 34, 79, 102, 185, 189, 204, 281, 423, 450, 466, 516, 519, 527, 528
<code>\acrlongpl</code>	246, 497, 555		
<code>\acrnameformat</code>	255, 398, 545, 561		
<code>\acrno linkfootnote</code>	557		
<code>\acronymentry</code>	233, 236, 237, 239, 240, 242–245, 387–390, 393–396, 545–555		
<code>\acronymfont</code>	112, 113, 150–153, 158, 164, 232, 234, 236–240, 242–245, 250, 252, 254, 256–261, 375, 376, 378–380, 388– 390, 392–396, 398–400, 472, 494–496, 499, 502, 545–555, 557, 558, 560, 562–566		
<code>\acronymname</code>	18, 43, 413, 427, 429		
<code>\acronymsort</code>	233, 236, 237, 239, 240, 242, 243, 245, 388–390, 393–396, 545–555		

B

<code>\b</code>	24, 417
<code>babel package</code>	<u>27</u> , <u>40</u> , <u>42</u> , <u>57</u>
<code>\begin</code>	172, 210, 285, 287, 291– 296, 299, 300, 302–325, 339, 507, 532, 572
<code>\BeginAccSupp</code>	356
<code>\begingroup</code>	5, 7, 33, 191, 195, 227, 405, 407, 520, 522, 542
<code>\bfseries</code>	292– 295, 297, 298, 303–306, 314–319, 321–325
<code>\bgroup</code>	24, 88, 128, 165, 202, 205, 416, 456, 502, 526, 528

bib2gls	24, 62, 194		
booktabs package	297–300		
\boolean	264, 568		
\boolfalse	34, 423, 424		
\booltrue	34, 424		
\bottomrule	297, 298		
\box	301		
C			
\c	24, 417		
\c@equation	120, 477		
\c@glossaryentry	12, 410		
\c@glossarysubentry	13, 410		
\c@page	192, 193, 195, 521–523		
\capitalisewords	161		
\catcode	33, 79, 128, 450		
\cGls	104, 466		
\cglS	103, 466		
\cGlsformat	102, 465, 466		
\cglSformat	101, 465, 466		
\cGlspl	105, 467		
\cglSpl	104, 467		
\cGlsplformat	102, 465, 467		
\cglSplformat	101, 465, 467		
\char	223, 540		
classicthesis package	9		
\cleardoublepage	49, 432		
\clearpage	49, 432		
\closeout	33, 79, 174,		
	175, 180, 190, 423, 450, 508, 509, 513, 520		
\compatglossarystyle	342–354		
\compatibleglossentry	221, 538, 539		
\compatiblesubglossentry	221, 539		
\copy	300, 301		
\count@	211, 533		
\csdef	22, 82–84, 94, 95, 100,		
	101, 205, 206, 225, 235, 341, 415, 452–		
	454, 460, 461, 464, 465, 528, 529, 541, 546		
\csedef	102, 192, 466, 521		
\csgdef	46, 66, 69, 101, 102,		
	200, 214, 430, 443, 444, 465, 466, 525, 535		
\cslet	73, 88, 95, 209, 446, 456, 461, 532		
\csname	14–17, 24, 38, 40, 42, 43, 48, 51, 52,		
	54, 55, 57, 59, 62, 67, 68, 75, 76, 82–84,		
	86, 87, 91, 92, 94–97, 99, 115, 119–121,		
	132–136, 150–157, 165, 167, 170–172,		
	177–180, 185, 189, 191, 192, 196–198,		
	201–204, 206, 214, 219, 221, 224, 225,		
	228, 248, 266–274, 281, 282, 331–333,		
	335–337, 349, 355, 356, 360, 361, 364,		
	368, 378–380, 402, 403, 411, 412, 417,		
	426–429, 431, 432, 434–436, 438–440,		
	443, 444, 448, 452–456, 458–464, 473,		
	474, 476, 477, 485–488, 494–498, 503,		
	506, 507, 510–513, 517, 519–521, 523,		
	524, 526–529, 535, 538–541, 543, 569–571		
\csshow	270, 571		
\csuse	43, 46, 66,		
	76, 83, 84, 115, 180–182, 206, 208, 210,		
	212, 213, 215, 216, 224, 235, 274, 342–		
	354, 429, 430, 443, 448, 449, 453, 454,		
	474, 513–515, 529, 531–534, 536, 541, 547		
\csxdef	93, 102, 248, 460, 466		
\currentglossary	12, 13, 46, 202, 410, 430, 527		
\currentglssubentry	13, 217, 410, 537		
\CurrentOption	37, 38, 273, 355, 426		
\CurrentTrackedLanguage	42, 404, 405, 428		
\CurrentTrackedTag	42, 404, 405, 428		
\CustomAcronymFields	265, 568, 569		
\CustomNewAcronymDef	265, 568, 569		
D			
\d	24, 417		
datatool package	207		
datatool-base package	4		
\day	170, 174, 337, 340, 505, 508		
\DeclareAcronymList			
 18, 20, 233, 235, 249, 251, 253,		
	256, 258, 261, 263, 265, 413, 414, 545,		
	546, 556, 558, 560, 561, 563, 566, 567, 569		
\DeclareCurrentRelease	4, 286		
\DeclareListParser	184, 516		
\DeclareOption	10, 273, 355, 409		
\DeclareOptionX	10, 409		
\DeclareRelease	4, 286		
\DeclareRobustCommand	43, 259, 429, 564		
\def	9, 10, 12–19, 23–25, 30–32, 35,		
	36, 38–40, 43, 44, 47, 50, 53–58, 62, 66–		
	68, 70–74, 77, 86, 88–93, 95, 97, 101–		
	105, 115, 116, 119–129, 131–157, 165,		
	166, 170, 174, 176–178, 180–182, 186,		
	187, 189, 192, 193, 195–198, 201, 202,		
	205, 209–211, 213, 214, 216, 223–228,		
	230–233, 248, 249, 251, 253–263, 265,		
	273, 276–279, 282–284, 311, 312, 331–		
	334, 336, 337, 340, 342, 349, 350, 355–		
	360, 378–385, 397–401, 409–414, 416,		

<code>\empty</code>	227, 355, 542	178, 192, 207, 219, 221, 233, 247–250,
<code>\encodingdefault</code>	24, 417	252, 255, 257, 260, 262, 355, 356, 448,
<code>\end</code>	172, 210, 285, 287, 291–	449, 478, 510–512, 521, 529, 530, 538,
	296, 299, 300, 302–325, 339, 507, 532, 572	539, 545, 555–557, 559, 561, 563, 565, 567
<code>\end@doifinlist</code>	50, 433	
<code>\end@getprefix</code>	197, 524	
<code>\end@glis@islistofacronyms</code>	19, 414	
<code>\EndAccSupp</code>	356	
<code>\endcsname</code>	14–	
	17, 24, 38, 40, 42, 43, 48, 51, 52, 54,	
	55, 57, 59, 62, 67, 68, 75, 76, 82–84,	
	86, 87, 91, 92, 94–97, 99, 115, 119–121,	
	132–136, 150–157, 165, 167, 170–172,	
	177–180, 185, 189, 191, 192, 196–198,	
	201–204, 206, 214, 219, 221, 224, 225,	
	228, 248, 266–274, 281, 282, 331–333,	
	335–337, 349, 355, 356, 360, 361, 364,	
	368, 378–380, 402, 403, 411, 412, 417,	
	426–429, 431, 432, 434–436, 438–440,	
	443, 444, 448, 452–456, 458–464, 473,	
	474, 476, 477, 485–488, 494–498, 503,	
	506, 507, 510–513, 517, 519–521, 523,	
	524, 526–529, 535, 538–541, 543, 569–571	
<code>\endfoot</code>	292–295, 297, 298, 302–306	
<code>\endgroup</code>	5,	
	7, 33, 191, 196, 227, 406, 407, 521, 523, 542	
<code>\endhead</code>	292–295, 297, 298, 302–306	
<code>\endtheglossary</code>	5, 405, 406	
<code>\entryname</code>	43, 292–295, 297, 298,	
	303, 304, 306, 314–319, 321–325, 427, 429	
<code>\equal</code>	26, 39, 49, 120,	
	183, 223, 281, 419, 427, 432, 476, 515, 540	
<code>equation (counter)</code>	120, 121	
<code>etoolbox package</code>	<u>4</u>	
<code>\expandafter</code>	7, 14–	
	17, 23, 24, 38, 39, 42, 51, 52, 54–59, 62,	
	67–69, 75, 76, 80–84, 86, 87, 91–94, 96,	
	97, 99, 115, 119–129, 158, 159, 161, 166,	
	168, 172, 176–179, 189, 191–193, 195,	
	196, 199, 202, 206, 211, 212, 219, 221,	
	225, 227, 228, 248, 250, 257, 266–271,	
	273, 274, 281, 282, 331, 335–337, 339,	
	340, 355, 356, 360, 361, 364, 365, 396,	
	402, 403, 407, 411, 412, 416, 417, 426–	
	428, 434–440, 443–445, 448–456, 458–	
	464, 473, 474, 476–483, 499, 503, 504,	
	507, 510–512, 520–523, 525, 526, 529,	
	533, 538, 539, 541–543, 557, 562, 569–571	
<code>\expandonce</code>	76, 77, 87, 122, 123, 177,	
		178, 192, 207, 219, 221, 233, 247–250,
		252, 255, 257, 260, 262, 355, 356, 448,
		449, 478, 510–512, 521, 529, 530, 538,
		539, 545, 555–557, 559, 561, 563, 565, 567
F		
<code>\fi</code>	5, 6, 8, 9, 12–19, 21, 23, 25, 26, 28,	
	33–39, 43–45, 47–59, 68, 73, 75, 87, 91–	
	93, 95–97, 101, 102, 118–129, 132–137,	
	159, 167–170, 173, 175, 176, 178, 179,	
	181, 182, 184, 189–191, 193, 195–200,	
	202, 204, 211, 215–218, 224, 226, 227,	
	237, 247–249, 251–253, 256, 258, 259,	
	261–266, 272, 281, 285, 288, 291, 293,	
	294, 297, 298, 300–302, 304, 305, 313,	
	315, 317, 320, 322, 324, 327, 328, 330–	
	333, 335–339, 341, 342, 347–350, 355,	
	365, 406–416, 418–420, 423–427, 429–	
	439, 444, 446, 448, 458–462, 465, 475–	
	488, 499, 504, 505, 507, 509, 510, 512,	
	514–516, 520–528, 533, 535–538, 540–	
	542, 548, 555, 556, 558–564, 566–569, 572	
file types		
<code>.aux</code>	204	
<code>.glo</code>	95	
<code>.ist</code>	168, 179	
<code>.toc</code>	49	
<code>.xdy</code>	44	
<code>glo</code>	271	
<code>\findrootlanguage</code>	438	
<code>\firstacronymfont</code>		
 114, 236–238, 243, 250, 254, 257,	
	260, 377, 378, 387, 388, 393, 394, 473,	
	545, 547, 548, 553, 557, 560, 562, 564, 565	
<code>\fmtversion</code>	25, 417	
<code>\footnote</code> ...	243, 249, 393, 394, 553, 556, 557	
<code>\FootnoteNewAcronymDef</code>	258, 562, 563	
<code>\footnotesize</code>	7, 407	
<code>\forallacronyms</code>	439	
<code>\forallglossaries</code>		
 59, 189, 201, 331, 439, 520, 526	
<code>\forallglstentries</code>	32,	
	99, 100, 103, 167, 423, 439, 464, 466, 504	
<code>\ForEachTrackedDialect</code> ..	42, 404, 405, 428	
<code>\forallglstentries</code>		
	... 59, 62, 95, 209, 331, 439, 440, 461, 531	
<code>\foralllistcsloop</code>	205, 210, 528, 532	
<code>\foralllistloop</code>	187, 213, 518, 534	

G

- garamondx package [229](#)
- \gdef ... [16](#), [33](#), [52](#), [67](#), [86](#), [91](#), [92](#), [191](#), [216](#),
[281](#), [412](#), [434](#), [443](#), [455](#), [458](#), [459](#), [520](#), [536](#)
- \Genacrfullformat ... [113](#), [234](#), [236](#), [237](#),
[243](#), [377](#), [387](#), [388](#), [393](#), [473](#), [545–548](#), [553](#)
- \genacrfullformat [113](#), [114](#), [234](#), [236](#), [237](#),
[243](#), [377](#), [387](#), [388](#), [393](#), [473](#), [545–548](#), [553](#)
- \GenericAcronymFields
..... [233](#), [236](#), [237](#), [239](#), [240](#), [242](#), [243](#),
[245](#), [387–390](#), [392](#), [393](#), [395](#), [396](#), [545–555](#)
- \Genplacrfullformat
..... [113](#), [234](#), [236](#), [237](#), [243](#),
[377](#), [387](#), [388](#), [393](#), [472](#), [473](#), [546–548](#), [553](#)
- \genplacrfullformat
..... [113](#), [114](#), [234](#), [236](#), [237](#), [243](#), [376](#),
[377](#), [387](#), [388](#), [393](#), [472](#), [473](#), [546–548](#), [553](#)
- getttitlestring package [287](#)
- \GetTitleStringDisableCommands ... [287](#)
- \GetTitleStringSetup [287](#)
- \glo@desc [342](#)
- \glo@do@compare [207](#), [208](#), [529](#), [530](#)
- \glo@grabfirst [212](#), [532](#), [533](#)
- \glo@label [62](#), [95](#), [440](#), [461](#)
- \glo@list [95](#), [461](#)
- \glo@name [219](#), [538](#)
- \glo@parent [62](#), [440](#)
- \glo@type [95](#), [461](#)
- \glo@value [80](#), [450](#)
- \global . [15–17](#), [75](#), [78](#), [88](#), [94](#), [99](#), [191](#), [203](#),
[211](#), [212](#), [217](#), [301](#), [411](#), [412](#), [448](#), [450](#),
[456](#), [460](#), [463](#), [464](#), [520](#), [527](#), [533](#), [534](#), [536](#)
- \glo@linkprefix .. [120](#), [166](#), [219](#), [476](#), [503](#), [538](#)
- \glosortentrieswarning .. [21](#), [205](#), [415](#), [528](#)
- glossaries package
[35](#), [37](#), [57](#), [58](#), [169](#), [266](#), [273](#), [287](#), [335](#), [355](#)
- glossaries-accsupp package [95](#), [355](#), [356](#)
- glossaries-extra package
..... [79](#), [171](#), [179](#), [201](#), [275](#), [355](#), [364](#)
- \GlossariesWarning [5](#), [6](#), [8](#), [21](#),
[22](#), [25](#), [26](#), [31](#), [32](#), [35](#), [36](#), [46](#), [50](#), [61](#), [65](#),
[72](#), [75](#), [78–80](#), [103–105](#), [115](#), [117](#), [165](#),
[180–182](#), [185–188](#), [191](#), [197](#), [200](#), [202](#),
[221](#), [224](#), [263](#), [335](#), [355](#), [406](#), [408](#), [414](#),
[415](#), [417–419](#), [422](#), [424](#), [430](#), [432](#), [440](#),
[442](#), [446](#), [448](#), [466](#), [467](#), [473–475](#), [502](#),
[513–515](#), [517–519](#), [521](#), [524–526](#), [539](#), [541](#)
- \GlossariesWarningNoLine
..... [5](#), [6](#), [21](#), [184](#), [186](#),
[189](#), [204](#), [281](#), [406](#), [415](#), [516](#), [517](#), [520](#), [528](#)
- \glossary [34–36](#), [336](#), [337](#), [424](#), [425](#)
- glossary package [1](#), [35](#), [228](#)
- glossary styles:
 - altlist [288](#), [289](#), [343](#)
 - altlistgroup [289](#), [343](#)
 - altlisthypergroup [289](#), [343](#)
 - altlong4col [295](#), [296](#), [305](#), [345](#)
 - altlong4col-booktabs [299](#), [300](#)
 - altlong4colborder [296](#), [345](#)
 - altlong4colheader [296](#), [299](#), [345](#)
 - altlong4colheaderborder [296](#), [345](#)
 - altlongragged4col ... [300](#), [305](#), [306](#), [346](#)
 - altlongragged4col-booktabs [300](#)
 - altlongragged4colborder [306](#), [346](#)
 - altlongragged4colheader [305](#), [346](#)
 - altlongragged4colheaderborder [306](#), [347](#)
 - altsuper4col [318](#), [319](#), [323](#), [354](#)
 - altsuper4colborder [318](#), [354](#)
 - altsuper4colheader [318](#), [354](#)
 - altsuper4colheaderborder ... [319](#), [354](#)
 - altsuperragged4col [323](#), [324](#), [352](#)
 - altsuperragged4colborder ... [324](#), [352](#)
 - altsuperragged4colheader ... [324](#), [352](#)
 - altsuperragged4colheaderborder .
..... [324](#), [352](#)
 - alttree [311](#), [326](#), [331](#), [333](#), [349](#)
 - alttreegroup [334](#), [350](#)
 - alttreehypergroup [334](#), [350](#)
 - index [9](#), [307](#), [325–328](#), [347](#)
 - indexgroup [327](#), [347](#)
 - indexhypergroup [327](#), [347](#)
 - inline [342](#)
 - list [9](#), [11](#), [287–289](#), [342](#)
 - listdotted [289](#), [290](#), [343](#)
 - listgroup [288](#), [342](#)
 - listhypergroup [288](#), [343](#)
 - long [291](#), [292](#), [297](#), [301](#), [343](#), [345](#)
 - long-booktabs [297](#), [299](#)
 - long3col [292](#), [293](#), [298](#), [344](#)
 - long3col-booktabs [297](#), [299](#)
 - long3colborder [293](#), [344](#)
 - long3colheader [293](#), [297](#), [344](#)
 - long3colheaderborder [293](#), [344](#)
 - long4col [294](#), [295](#), [298](#), [344](#)
 - long4col-booktabs [298](#), [299](#)
 - long4colborder [295](#), [345](#)
 - long4colheader [295](#), [298](#), [345](#)

long4colheaderborder	295, 345	treehypergroup	329, 348
longborder	291, 344	treenoname	310, 326, 329, 330, 348
longheader	292, 297, 344	treenonamegroup	330, 349
longheaderborder	292, 344	treenonamehypergroup	330, 349
longragged	299, 301, 303	glossary-hypernav package	168
longragged-booktabs	299	glossary-list package	9, 11, 286
longragged3col	299, 303, 304, 346	glossary-long package	11, 290, 305, 312, 313
longragged3col-booktabs	299	glossary-longragged package	301
longragged3colborder	304, 346	glossary-mcols package	307
longragged3colheader	304, 346	glossary-super package	11, 290, 312, 319, 323
longragged3colheaderborder	304, 346	glossary-superragged package	319
longraggedborder	302, 345	glossary-tree package	11, 325
longraggedheader	303, 346	\glossaryentry	197, 198, 337, 523, 524
longraggedheaderborder	303, 346	glossaryentry (counter)	12, 13, 217, 218
mcolalmtree	311, 351	\glossaryentryfield	219, 342–349, 351–354, 387, 538, 539
mcolalmtreegroup	311, 351	\glossaryentrynumbers	10, 173, 202, 203, 212, 216, 217, 339, 409, 507, 526, 527, 533, 534, 536
mcolalmtreehypergroup	311, 312, 351	\glossaryheader	172, 210, 283, 285–289, 291–295, 297, 298, 302–306, 308–311, 313, 314, 316, 320, 321, 323, 326–331, 334, 339, 507, 532, 538, 572, 573
mcolindex	307, 350	\glossarymark	47, 431
mcolindexgroup	307, 350	\glossaryname	17, 42, 43, 412, 427–429
mcolindexhypergroup	307, 308, 350	\glossarypostamble	172, 210, 339, 430, 507, 532
mcoltree	309, 350	\glossary preamble	172, 210, 339, 430, 506, 532
mcoltreegroup	350	\glossarysection	172, 210, 339, 430, 506, 532
mcoltreehypergroup	309, 350	\glossarystyle	540, 541
mcoltreenoname	310, 351	glossarysubentry (counter)	13, 217, 218
mcoltreenonamegroup	310, 351	\glossarysubentryfield	221, 342–349, 351–354, 387, 539
mcoltreenonamehypergroup	310, 351	\glossarytitle	172, 201, 202, 210, 214, 339, 506, 526, 527, 532, 535
sublistdotted	343	\glossarytoctitle	9, 17, 18, 36, 37, 40, 43, 47, 172, 202, 210, 214, 215, 339, 408, 412, 413, 425, 427, 429, 431, 506, 526, 532, 535
super	313, 314, 321, 353	\glossentry	95, 203, 212, 221, 283, 285, 286, 288–292, 294, 302, 303, 305, 313, 315, 316, 320, 322, 323, 326, 328, 329, 332, 461, 527, 534, 539, 572, 573
super3col	314–316, 353	\Glossentrydesc	386, 538
super3colborder	315, 353	\glossentrydesc	283–286, 288–294, 302–305, 313, 315, 316, 320, 322–324, 327, 328, 330, 332, 333, 386, 538, 572, 573
super3colheader	315, 353	\Glossentryname	538
super3colheaderborder	316, 353		
super4col	316–318, 354		
super4colborder	317, 354		
super4colheader	317, 354		
super4colheaderborder	317, 354		
superborder	313, 353		
superheader	314, 353		
superheaderborder	314, 353		
superragged	320, 321, 351		
superragged3col	321–323, 352		
superragged3colborder	322, 352		
superragged3colheader	322, 352		
superragged3colheaderborder	323, 352		
superraggedborder	320, 351		
superraggedheader	321, 351		
superraggedheaderborder	321, 352		
tree	308, 328, 329, 331, 347		
treegroup	309, 329, 348		

<code>\glossentryname</code>	283–292, 294, 302, 303, 305, 313, 315, 316, 320, 322, 323, 327, 328, 330, 332, 333, 386, 538, 572, 573	<code>\gls@docdef@addlabel</code>	79
<code>\Glossentrysymbol</code>	386, 539	<code>\gls@docdef@labellist</code>	78
<code>\glossentrysymbol</code>	283, 284, 294, 305, 316, 317, 323, 324, 327, 328, 330, 332, 333, 386, 539	<code>\gls@docclearpage</code>	49, 432
<code>\GLS</code>	486	<code>\gls@doentryfmt</code>	443
<code>\Gls</code>	104, 228, 247, 466, 485, 543, 555	<code>\gls@dosubst</code>	122, 478
<code>\gls</code>	35, 103, 185, 218, 228, 247, 424, 466, 485, 517, 537, 543, 555	<code>\gls@dotocitle</code> .	202, 203, 214, 526, 527, 535
<code>\gls@accessibility</code>	357	<code>\gls@end@sanitizesort</code>	23, 24, 416
<code>\gls@accsupp@engine</code>	356	<code>\gls@endcheck</code>	76, 77, 449
<code>\gls@Alphpage</code>	192, 195, 521, 523	<code>\gls@glossary</code>	35, 36, 196–198, 424, 425, 520, 523, 524
<code>\gls@alphpage</code>	192, 195, 521, 523	<code>\gls@glsdefs@addlabel</code>	80
<code>\gls@arabicpage</code>	192, 195, 521, 522	<code>\gls@glsdefs@labellist</code>	78, 79
<code>\gls@assign@desc</code>	88, 93, 456, 460	<code>\gls@gobbleopt</code>	68, 443, 444
<code>\gls@assign@descplural</code> .	248, 257, 258, 260–263, 397, 400, 401, 556, 563, 565–567	<code>\gls@grplabel</code>	280
<code>\gls@assign@field</code>	77, 82, 83, 88, 90, 92–94, 273, 274, 449, 452, 453, 456, 457, 459, 460	<code>\gls@hypergroup prerun</code>	281
<code>\gls@assign@firstpl</code>	248, 251, 253, 255, 257, 258, 260– 263, 397–401, 556–559, 561, 563, 565, 567	<code>\gls@ifnotmeasuring</code>	98, 462, 463
<code>\gls@assign@plural</code>	248, 251, 253, 255, 257, 258, 260–263, 397–401, 556, 558, 559, 561, 563, 565–567	<code>\gls@inlinepostchild</code>	283, 284, 342
<code>\gls@assign@symbolplural</code>	248, 251, 253, 255, 260–263, 397–399, 401, 556, 558, 559, 561, 565–567	<code>\gls@inlinesep</code>	283, 342
<code>\gls@automake@nr</code>	33, 182, 423, 515	<code>\gls@inlinesubsep</code>	283, 284, 342
<code>\gls@automake@val</code>	33, 423	<code>\gls@islistofacronyms</code>	19, 414
<code>\gls@begindocdefs</code>	79, 450	<code>\gls@istfilebase</code>	43, 44, 180, 181, 429, 513, 514
<code>\gls@checkisacronymlist</code>	118, 414, 475	<code>\gls@label</code>	228, 543
<code>\gls@checkseeallowed</code>	72, 183, 185, 446, 449, 516, 517	<code>\gls@level</code>	91, 92, 212, 448, 458, 459, 533
<code>\gls@checkseeallowed@preambleonly</code> .	446, 449	<code>\gls@noidxglossary</code>	185, 517, 519
<code>\gls@codepage</code>	58, 180, 181, 204, 421, 438, 513, 514, 528	<code>\gls@nonumberlist@nr</code>	72, 446
<code>\gls@debug@nr</code>	5, 36, 406, 424	<code>\gls@nonumberlist@val</code>	72, 446
<code>\gls@debug@val</code>	5, 36, 406, 424	<code>\gls@nosetquote</code>	89, 174, 176, 178, 457, 508, 509, 512
<code>\gls@defdocnewglossaryentry</code>	79, 100, 449, 450, 464	<code>\gls@number</code>	195, 522
<code>\gls@defglossaryentry</code>	77–80, 88, 449, 450, 456, 457	<code>\gls@numberedsection@nr</code> ..	9, 215, 408, 535
<code>\gls@disablepagerefexpansion</code>	191, 195, 521, 523	<code>\gls@numberedsection@val</code> .	9, 215, 408, 535
<code>\gls@do@addxdyattribute</code>	52, 434	<code>\gls@numberpage</code>	192, 195, 521, 522
		<code>\gls@org@glossaryentryfield</code> ...	203, 527
		<code>\gls@org@glossarysubentryfield</code>	203, 527
		<code>\gls@org@insert</code>	254, 256, 257, 259, 560, 562, 564
		<code>\gls@orgAlph</code>	195, 522, 523
		<code>\gls@orgalph</code>	195, 522, 523
		<code>\gls@orgarabic</code>	195, 522
		<code>\gls@orgnumber</code>	195, 522
		<code>\gls@orgRoman</code>	195, 522, 523
		<code>\gls@orgromannumeral</code>	195, 522
		<code>\gls@orgthe</code>	195, 522
		<code>\gls@original@glossary</code>	36, 424, 425
		<code>\gls@original@makeglossary</code> ..	36, 424, 425
		<code>\gls@protected@pagefmts</code>	122, 192, 478, 521
		<code>\gls@Romanpage</code>	192, 195, 521, 523
		<code>\gls@romanpage</code>	192, 195, 521, 522

<code>\gls@save@numberlist</code>	10, 409, 525	<code>\GlsAddXdyStyle</code>	437, 438
<code>\gls@seen@index@nr</code>	8, 407	<code>\glsautomakefalse</code>	33, 34, 182, 423, 515
<code>\gls@seen@index@val</code>	8, 407	<code>\glsautomaketrue</code>	33, 423
<code>\gls@set@xr@key</code>	72, 446	<code>\glsautoprefix</code>	9, 215, 408, 535
<code>\gls@suffixF</code>		<code>\glsbackslash</code>	504
.....	45, 173, 175, 339, 341, 430, 507, 509	<code>\glscapitalisewords</code>	161
<code>\gls@suffixFF</code>		<code>\glscapspace</code>	106,
.....	45, 173, 175, 339, 341, 430, 507, 509		108, 110–113, 131–136, 150–156, 241,
<code>\gls@text</code>	114, 473		254, 260, 368, 370, 373–380, 391, 467,
<code>\gls@the</code>	195, 522		469–473, 485–487, 494–498, 551, 560, 564
<code>\gls@thissty</code>	29, 420	<code>\glscategory</code>	364
<code>\gls@tmp</code>	189, 259, 520, 564	<code>\glsclearpage</code>	48, 431, 432
<code>\gls@tmplen</code>	130, 331–333, 349, 350, 484	<code>\glsclsebrace</code>	
<code>\gls@tr@set@acronym@toctitle</code> ...	18, 413	55, 173, 339, 340, 436, 504, 507, 508
<code>\gls@tr@set@main@toctitle</code>	17, 412	<code>\glscompositor</code>	
<code>\gls@tr@set@numbers@toctitle</code> ...	37, 425	...	44, 45, 55, 175, 338, 341, 430, 436, 509
<code>\gls@tr@set@symbols@toctitle</code> ...	36, 425	<code>\glscounter</code>	20, 39, 50,
<code>\gls@translate@nr</code>	28, 420		68, 92, 120, 335, 414, 427, 433, 443, 459, 476
<code>\gls@translate@val</code>	28, 420	<code>\glscurrententrylabel</code> ..	200, 203, 525, 527
<code>\gls@wrglossary</code>	191, 520	<code>\glscurrentfieldvalue</code>	64, 65, 442
<code>\gls@xdystring</code>	122, 478	<code>\glscustomtext</code> ..	106, 109, 110, 112, 114,
<code>\gls@xindy@glsnumbersfalse</code>	31, 422		131–136, 150–157, 241, 242, 250, 254,
<code>\gls@xindy@glsnumberstrue</code>	30, 421		256, 257, 259, 260, 368, 372, 375, 377–
<code>\gls@xr@key</code>	7, 8, 72, 407, 408, 446		380, 390, 392, 467, 470, 472, 473, 485–
<code>\glsaccessibility</code>	364, 365		487, 494–498, 551, 557, 560, 562, 564, 565
<code>\glsaccsupp</code>	364, 365	<code>\GlsDeclareNoHyperList</code>	20, 414
<code>\glsacronymtrue</code>	18, 413	<code>\glsdefaultshortaccess</code>	396–400
<code>\glsacrpluralsuffix</code> ..	41, 229, 238, 242,	<code>\glsdefaulttype</code>	17, 46, 57, 59,
	243, 245, 249, 427, 543, 549, 552–554, 556		66, 67, 90, 105, 115, 189, 201, 413, 430,
<code>\glsacrshortcutsfalse</code>	38, 426		438, 439, 443, 457, 467, 473, 474, 520, 526
<code>\glsacrshortcutstrue</code>	38, 426	<code>\glsdefmain</code>	17, 69, 412, 413, 444
<code>\glsacspace</code>	236, 237, 239, 547, 548, 550	<code>\glsdefs@newdocentry</code>	80
<code>\glsadd</code>	35, 167, 424, 503, 504	<code>\GLSdesc</code>	490
<code>\glsadd options</code>		<code>\Glsdesc</code>	490
counter	166	<code>\glsdesc</code>	490
format	166, 225	<code>\GLSdescplural</code>	490
<code>\glsaddall</code>	504	<code>\Glsdescplural</code>	490
<code>\glsaddall options</code>		<code>\glsdescplural</code>	490
types	166, 167	<code>\glsdescriptionaccessdisplay</code>	
<code>\glsaddallunused</code>	504		370–372, 382, 386, 387
<code>\glsaddkey</code>	452	<code>\glsdescriptionpluralaccessdisplay</code>	
<code>\GlsAddLetterGroup</code>	439	369, 370, 382, 383
<code>\glsaddprotectedpagefmt</code>	521	<code>\glsdescwidth</code>	291–293,
<code>\GlsAddSortRule</code>	437		296, 299, 300, 302–306, 313–316, 318–325
<code>\glsaddstoragekey</code>	452	<code>\glsdetoklabel</code> .	32, 60–65, 73, 80, 85–89,
<code>\GlsAddXdyAlphabet</code>	435		95, 99, 101, 102, 119, 157, 158, 165–167,
<code>\GlsAddXdyAttribute</code> ..	51–53, 335, 336, 433–435		185–187, 193, 196, 203, 206, 207, 211,
<code>\GlsAddXdyCounters</code> .	51, 52, 68, 433, 434, 444		212, 214, 217–219, 266–270, 274, 275,
<code>\GlsAddXdyLocation</code>	436, 437		

331, 355, 356, 365–368, 402, 403, 439–
 442, 446, 450, 454–457, 461, 463–466,
 476, 498, 499, 503, 517, 518, 522, 523,
 527, 529, 530, 533, 535, 537, 538, 569–571
 \glsdisablehyper 484
 \glsdisp 487
 \glsdisplay 106, 115, 467, 473, 474
 \glsdisplayfirst ... 106, 115, 467, 473, 474
 \glsdisplaynumberlist 187, 502, 518
 \glsdohyperlink 130, 131, 484, 485
 \glsdohypertarget 130, 131, 484, 485
 \glsdoifexists 62–64, 85–88, 98, 131–137,
 149–156, 161, 165, 166, 186, 187, 276–
 279, 378–380, 386, 439–441, 454–456,
 463, 485–488, 494–498, 500, 502, 503, 518
 \glsdoifexistsordo . 118, 157, 440, 475, 498
 \glsdoifexistsorwarn
 214, 219, 220, 440, 534, 538, 539
 \glsdoifnoexists 77, 88, 440, 449, 456
 \glsdonohyperlink 120, 130, 476, 484
 \glsdoparenifnotempty 564
 \glsdosanitizesort 14, 411
 \glsenableentrycount 464, 466
 \glsenablehyper 485
 \glsentryaccess 365
 \glsentrycounter ... 224, 227, 476, 540, 542
 \glsentrycounterfalse 13, 410
 \glsentrycounterlabel 218, 537
 \GlsEntryCounterLabelPrefix
 217, 218, 536, 537
 \glsentrycountertrue 13, 410
 \glsentrycurrcount 101–103, 464, 466
 \Glsentrydesc 142, 220, 382, 386, 490, 499, 538
 \glsentrydesc ... 108, 109, 141, 142, 220,
 370–372, 382, 386, 469, 470, 490, 499, 538
 \glsentrydescaccess 366
 \Glsentrydescplural 142, 382, 490, 499
 \glsentrydescplural 106, 107, 142,
 143, 369, 370, 382, 383, 467, 468, 490, 499
 \glsentrydescpluralaccess 367
 \Glsentryfirst 104, 109, 111,
 138, 371, 374, 381, 466, 469, 471, 488, 500
 \glsentryfirst
 . 103, 108, 109, 111, 138, 139, 371, 372,
 374, 375, 381, 466, 469–471, 488, 489, 500
 \glsentryfirstaccess 366
 \Glsentryfirstplural ... 105, 107, 110,
 140, 369, 373, 381, 467, 468, 471, 489, 500
 \glsentryfirstplural
 . 104, 106, 107, 110, 111, 140, 369, 370,
 373, 374, 381, 467, 468, 470, 471, 489, 500
 \glsentryfirstpluralaccess 366
 \glsentryfmt 67, 69, 443, 444, 467
 \Glsentryfull
 234, 242, 244, 392, 394, 502, 546, 552, 554
 \glsentryfull
 234, 242, 244, 392, 394, 502, 546, 552, 553
 \Glsentryfullpl
 234, 242, 244, 392, 395, 502, 546, 552, 554
 \glsentryfullpl
 234, 242, 244, 392, 394, 502, 546, 552, 554
 \glsentryitem
 283, 285–292, 294, 302, 303, 305,
 313, 315, 316, 320, 322, 323, 327, 328,
 330, 332, 342–349, 351–354, 537, 572, 573
 \Glsentrylong 104, 154,
 158, 164, 236, 241, 242, 377, 380, 387,
 391, 392, 466, 496, 499, 502, 547, 551, 552
 \glsentrylong
 . 103, 114, 153, 155, 158, 164, 236, 237,
 239–245, 257, 377, 379, 380, 387–396,
 466, 473, 496, 497, 499, 502, 547–555, 562
 \glsentrylongaccess 367
 \Glsentrylongpl 105,
 156, 164, 236, 237, 241, 242, 378, 387,
 391, 392, 467, 498, 502, 547, 548, 551, 552
 \glsentrylongpl 104, 114,
 155, 157, 164, 236, 237, 241–244, 257,
 264, 378, 387, 388, 391–395, 467, 473,
 497, 498, 502, 547, 548, 551–554, 562, 568
 \glsentrylongpluralaccess 367
 \Glsentryname 141, 220, 382, 386, 490, 499, 538
 \glsentryname 141, 331, 382, 386, 489, 490, 499
 \glsentrynumberlist 165, 186, 502, 517
 \glsentryparent 501
 \Glsentryplural 107,
 110, 139, 369, 373, 381, 468, 471, 489, 500
 \glsentryplural . 106, 107, 110, 139, 369,
 370, 373, 381, 467, 468, 470, 471, 489, 500
 \glsentrypluralaccess 366
 \Glsentryprefix 277
 \glsentryprefix 276, 278
 \Glsentryprefixfirst 277
 \glsentryprefixfirst 276, 279
 \Glsentryprefixfirstplural 278
 \glsentryprefixfirstplural 276, 279
 \Glsentryprefixplural 278

<code>\glsentryprefixplural</code>	276, 279	<code>\glsentryuserivaccess</code>	368
<code>\glsentryprevcount</code>	101, 102, 464, 465	<code>\Glsentryuseriv</code>	148, 385, 493, 501
<code>\Glsentryshort</code>	113, 150, 158, 159, 237, 243, 244, 376–378, 388, 393, 394, 472, 494, 499, 502, 548, 553, 554	<code>\glsentryuseriv</code>	148, 385, 493, 501
<code>\glsentryshort</code>	112–114, 150, 151, 158, 159, 164, 234, 236, 237, 239, 240, 242–245, 376–379, 387–390, 392– 396, 472, 473, 494, 499, 501, 502, 546–555	<code>\glsentryuserivaccess</code>	368
<code>\glsentryshortaccess</code>	367	<code>\Glsentryuserivi</code>	149, 385, 493, 501
<code>\Glsentryshorttpl</code>	112, 152, 237, 243, 244, 375, 388, 394, 395, 472, 495, 502, 548, 553, 554	<code>\glsentryuserivi</code>	149, 385, 493, 501
<code>\glsentryshorttpl</code>	112, 114, 151, 153, 164, 236, 237, 242–244, 264, 375, 376, 378, 387, 388, 392–395, 472, 473, 495, 496, 502, 547, 548, 552–554, 568	<code>\glsentryuseriviaccess</code>	368
<code>\glsentryshorttplpluralaccess</code>	367	<code>\glsesclocationsfalse</code>	185, 517
<code>\glsentrysort</code>	501	<code>\glsesclocationstrue</code>	10, 409
<code>\Glsentrysymbol</code>	143, 220, 383, 386, 491, 500, 539	<code>\glsexpandfields</code>	449
<code>\glsentrysymbol</code>	108, 109, 143, 144, 220, 250, 254, 259, 370–372, 383, 386, 469, 470, 491, 500, 539, 557, 560, 564	<code>\glsfieldaccsupp</code>	364, 365
<code>\glsentrysymbolaccess</code>	366	<code>\glsfielddef</code>	455
<code>\Glsentrysymbolplural</code> ..	144, 383, 491, 500	<code>\glsfieldedef</code>	454
<code>\glsentrysymbolplural</code>	106, 107, 144, 250, 254, 259, 369, 370, 383, 467, 468, 491, 500, 557, 560, 564	<code>\glsfieldfetch</code>	161, 455, 500
<code>\glsentrysymbolpluralaccess</code>	366	<code>\glsfieldgdef</code>	454
<code>\Glsentrytext</code>	108, 111, 138, 371, 374, 380, 469, 471, 488, 500	<code>\glsfieldxdef</code>	454
<code>\glsentrytext</code>	108, 109, 111, 137, 138, 166, 199, 370, 371, 374, 375, 380, 469–471, 488, 500, 503, 525	<code>\GLSfirst</code>	489
<code>\glsentrytextaccess</code>	366	<code>\Glsfirst</code>	488
<code>\glsentrytitlecase</code>	500, 501	<code>\glsfirst</code>	488
<code>\glsentrytype</code>	90, 457, 501	<code>\glsfirstaccessdisplay</code>	371, 372, 374, 375, 381
<code>\Glsentryuseri</code>	145, 384, 491, 501	<code>\GLSfirstplural</code>	489
<code>\glsentryuseri</code> ..	145, 383, 384, 491, 492, 501	<code>\Glsfirstplural</code>	489
<code>\glsentryuseriaccess</code>	367	<code>\glsfirstplural</code>	489
<code>\Glsentryuserii</code>	146, 384, 492, 501	<code>\glsfirstpluralaccessdisplay</code>	369, 370, 373, 381
<code>\glsentryuserii</code>	145, 146, 384, 492, 501	<code>\glsfirstpluralacesdisplay</code>	374
<code>\glsentryuseriiaccess</code>	367	<code>\glsngenacfmt</code>	236, 237, 243, 387, 388, 393, 472, 547, 548, 553
<code>\Glsentryuseriii</code>	147, 384, 492, 501	<code>\glsngenentryfmt</code> ..	236, 237, 242, 243, 247, 250, 252, 254, 256, 257, 259, 262, 264, 387, 388, 391, 393, 470, 547, 548, 551, 553, 555, 557, 558, 560, 562, 564, 566, 568
<code>\glsentryuseriii</code> ...	146, 147, 384, 492, 501	<code>\glsgetgrouplabel</code>	540
<code>\glsentryuseriiiaccess</code>	368	<code>\glsgetgrouptitle</code>	282, 285– 289, 307–312, 327–330, 334, 539, 572, 573
<code>\Glsentryuseriv</code>	147, 385, 493, 501	<code>\gls glossarymark</code>	47, 431
<code>\glsentryuseriv</code> .	147, 148, 385, 492, 493, 501	<code>\gls groupheading</code>	173, 212, 283, 285–289, 291, 292, 294, 302, 303, 305, 307–313, 315, 316, 320, 321, 323, 326– 331, 333, 334, 340, 508, 534, 539, 572, 573

<code>\glshypernumber</code> .. 46, 227, 228, 430, 541–543	<code>\glslistdottedwidth</code> 286, 290, 343, 573
<code>\glsifhyper</code> 475	<code>\glslistexpandedname</code> 287
<code>\glsifhyperon</code> 117, 475, 476	<code>\glslistgroupheaderfmt</code> . 285–289, 572, 573
<code>\glsIfListOfAcronyms</code> 19, 20, 414	<code>\glslistinit</code> 287
<code>\glsifplural</code> 106, 110, 112, 113, 131–136, 150– 156, 241, 250, 254, 257, 259, 368, 372, 375, 376, 378–380, 390, 467, 470, 472, 485–487, 494–498, 551, 557, 560, 562, 564	<code>\glslistnavigationitem</code> 285, 286, 288, 289, 572, 573
<code>\glsifusedtranslatorordict</code> 419	<code>\glslocalreset</code> 99, 463, 464
<code>\glsifusetranslator</code> 27, 28, 42, 404, 419, 420, 428, 429	<code>\glslocalresetall</code> 464
<code>\glsignore</code> 504	<code>\glslocalunset</code> 100, 132–137, 463, 464, 485–488
<code>\glsindexonlyfirstfalse</code> 29, 421	<code>\glslocalunsetall</code> 464
<code>\glsinlinedescformat</code> 283, 342	<code>\glslongaccessdisplay</code> 377, 379, 380, 387–397
<code>\glsinlinedopostchild</code> 283, 342	<code>\glslongkey</code> 401, 543
<code>\glsinlineemptydescformat</code> 283, 342	<code>\glslongpluralaccessdisplay</code> 378, 387, 388, 391–395, 397
<code>\glsinlinenameformat</code> 283, 342	<code>\glslongpluralkey</code> 401, 543
<code>\glsinlineparentchildseparator</code> 283, 342	<code>\glslongtok</code> 233, 234, 236, 237, 242, 243, 247–253, 255– 258, 260–265, 387, 388, 392, 393, 396– 401, 545–548, 552, 553, 555–563, 565–569
<code>\glsinlinepostchild</code> 283, 342	<code>\glsLTpenaltycheck</code> 301
<code>\glsinlineseparator</code> 283, 342	<code>\glsmccols</code> 307–312
<code>\glsinlinesubdescformat</code> 284, 342	<code>\glsmoveentry</code> 461
<code>\glsinlinesubnameformat</code> 284, 342	<code>\GLSname</code> 490
<code>\glsinlinesubseparator</code> 284, 342	<code>\Glsname</code> 489
<code>\glsinsert</code> 106–114, 131–136, 150– 156, 241, 242, 250, 254, 256, 257, 259, 260, 369–380, 391, 392, 467–473, 485– 487, 494–498, 551, 557, 560, 562, 564, 565	<code>\glsname</code> 489
<code>\glskeylisttok</code> 233, 248–251, 253, 255–258, 260–263, 265, 396–401, 545, 556–561, 563, 565–567, 569	<code>\glsnameaccessdisplay</code> 382, 386, 387
<code>\glslabel</code> 89, 106–114, 118–120, 150–156, 236, 237, 241, 243, 250, 254, 256, 257, 259, 369–377, 387, 388, 390, 391, 393, 457, 467–473, 475, 476, 494– 498, 547, 548, 551, 553, 557, 560, 562, 564	<code>\glsnamefont</code> 219–221, 355, 356, 386, 538, 539, 541
<code>\glslabeltok</code> 233, 247–254, 256–258, 260–265, 397–400, 545, 555–563, 565–569	<code>\glsnavhyperlink</code> 282
<code>\glsletentryfield</code> 498	<code>\glsnavhyperlinkname</code> 280, 281
<code>\glslink</code> 234, 242–244, 249, 392, 394, 475, 545, 546, 552, 553, 556	<code>\glsnavhypertarget</code> 285–289, 308–312, 328–330, 334, 572, 573
<code>\glslink options</code>	<code>\glsnavigation</code> 285, 286, 288, 289, 308–312, 327, 329, 330, 334, 572, 573
counter 116, 131, 272	<code>\glsnextpages</code> . 10, 73, 202, 409, 446, 527, 536
format 116, 131, 225	<code>\glsnoexpandfields</code> 449
hyper 116, 118, 119, 131	<code>\glsnogroupskipfalse</code> 12, 410
local 116	<code>\glsnoidxdisplayloc</code> 187, 189, 518, 519, 534
<code>\glslinkcheckfirsthyperhook</code> ... 119, 476	<code>\glsnoidxdisplayloclisthandler</code> 187, 518, 534
<code>\glslinkpostsetkeys</code> 120, 476	<code>\glsnoidxloclist</code> ... 186, 212, 518, 533, 534
<code>\glslinkvar</code> 117, 475	<code>\glsnoidxloclisthandler</code> 213, 534
	<code>\glsnoidxnumberlistloophandler</code> 187, 518, 519
	<code>\glsnoidxstripaccents</code> 24, 416, 417
	<code>\glsnomakeindexwarning</code> 176, 432, 509
	<code>\glsnonextpages</code> 72, 202, 446, 527, 536
	<code>\glsnopostdotfalse</code> 12, 409

<code>\glsnoxindywarning</code>	45, 51–53, 56–58, 169, 430, 432–435, 437, 438, 504, 505
<code>\glsnumberformat</code>	165, 430, 503
<code>\glsnumberlistloop</code>	187, 518, 519
<code>\glsnumbersgroupname</code>	37, 43, 223, 425, 427, 429, 540
<code>\glsnumlistlastsep</code>	165, 187, 503, 518
<code>\glsnumlistparser</code>	166, 184, 503, 516
<code>\glsnumlistsep</code>	165, 187, 503, 518
<code>\glsopenbrace</code>	55, 173, 339, 340, 436, 504, 507, 508
<code>\glsorder</code>	30, 180–183, 208, 421, 513–515, 531
<code>\glsorg@endtheglossary</code>	5, 405, 406
<code>\glsorg@PrintChanges</code>	5, 405, 406
<code>\glsorg@theglossary</code>	5, 405
<code>\glspagelistwidth</code> ...	292, 293, 296, 299, 300, 303–306, 314–316, 318, 319, 321–325
<code>\glspar</code>	429
<code>\glspatchLToutput</code>	297–300
<code>\glspatchtabularx</code>	462
<code>\glspenaltygroupskip</code>	297, 298
<code>\glsperscentchar</code>	80, 172, 174, 450, 451, 504, 507, 508
<code>\GLSpl</code>	487
<code>\Glspl</code>	105, 247, 467, 486, 555
<code>\glspl</code>	104, 247, 467, 486, 555
<code>\GLSplural</code>	489
<code>\Glsplural</code>	489
<code>\glsplural</code>	489
<code>\glspluralaccessdisplay</code>	369, 370, 373, 381
<code>\glspluralsuffix</code>	41, 92, 236–238, 388, 393–395, 427, 459, 547, 548
<code>\glspostdescription</code>	43, 284, 285, 288, 289, 291, 302, 313, 320, 327, 328, 330, 332, 333, 342–345, 347–351, 353, 409, 429, 572
<code>\glspostinline</code>	283
<code>\glspostlinkhook</code>	118, 132–137, 150–157, 378–380, 475, 485–488, 494–498
<code>\glsprefixsep</code>	276–279
<code>\glsprestandardsort</code>	14, 411
<code>\glsquote</code>	504
<code>\glsrefentry</code>	537
<code>\glsreset</code>	99, 463, 464
<code>\glsresetall</code>	464
<code>\glsresetentrycounter</code>	217, 537
<code>\glsresetentrylist</code>	172, 210, 216, 339, 507, 532, 536
<code>\glsresetsubentrycounter</code>	218, 283, 342, 536, 537
<code>\glssanitizesortfalse</code>	26, 418
<code>\glssanitizesorttrue</code>	26, 418
<code>\glssavenumberlistfalse</code>	10, 409
<code>\glssavewritesfalse</code>	34, 423
<code>\glssee</code>	525
<code>\glsseeformat</code>	171, 185, 187, 339, 506, 517–519, 525
<code>\glsseeitem</code>	199, 525
<code>\glsseeitemformat</code>	199, 525
<code>\glsseelastsep</code>	199, 525
<code>\glsseelist</code>	198, 525
<code>\glsseesep</code>	199, 525
<code>\glsSetAlphaCompositor</code>	430
<code>\glsSetCompositor</code>	430
<code>\glssetexpandfield</code>	22, 25–27, 415, 416, 418, 419
<code>\glssetnoexpandfield</code>	22, 23, 25, 26, 415–418
<code>\GlsSetQuote</code>	89, 174, 457, 508, 509, 512
<code>\glsSetSuffixF</code>	430
<code>\glsSetSuffixFF</code>	430
<code>\glssettoctitle</code>	42, 202, 427, 429, 526
<code>\GlsSetWriteIstHook</code>	509
<code>\GlsSetXdyCodePage</code>	438
<code>\GlsSetXdyFirstLetterAfterDigits</code> ..	504
<code>\GlsSetXdyLanguage</code>	438
<code>\GlsSetXdyLocationClassOrder</code>	437
<code>\GlsSetXdyMinRangeLength</code>	505
<code>\GlsSetXdyNumberGroupOrder</code>	504, 505
<code>\GlsSetXdyStyles</code>	438
<code>\glsshortaccessdisplay</code>	159, 376–379, 387–390, 392–397, 499
<code>\glsshortaccsupp</code>	365
<code>\glsshortkey</code>	401, 543
<code>\glsshortpluralaccessdisplay</code>	375, 376, 378, 387, 388, 392–395, 397
<code>\glsshortpluralkey</code>	401, 543
<code>\glsshorttok</code>	233, 247–253, 255–258, 260–265, 396–401, 545, 555–561, 563, 565–569
<code>\glsshowaccsupp</code>	6, 406, 407
<code>\glsshowtarget</code>	7, 407
<code>\glsshowtargetfont</code>	6, 407
<code>\glsshowtargetouter</code>	6, 407
<code>\glsshowtargetsymbols</code>	6, 407
<code>\glssortnumberfmt</code>	15, 16, 411, 412
<code>\glsspace</code>	230, 248, 249, 543
<code>\glsstepentry</code>	218, 537
<code>\glsstepsubentry</code>	218, 537, 538

<code>\glssubentrycounterfalse</code>	13, 410	<code>\glsunsetall</code>	464
<code>\glssubentrycounterlabel</code> ...	218, 537, 538	<code>\glsupacrpluralsuffix</code> 238, 244, 252, 256,	
<code>\glssubentryitem</code>	284–	258, 261, 427, 549, 554, 558, 562, 564, 566	
286, 288–291, 293, 294, 302, 304, 305,		<code>\GlsUseAcrEntryDispStyle</code>	
313, 315, 316, 320, 322, 324, 327, 328,		235, 238–240, 243–245,
330, 332, 342–349, 351–354, 537, 572, 573		389, 390, 393, 395, 396, 546–552, 554, 555	
<code>\GLSsymbol</code>	491	<code>\GlsUseAcrStyleDefs</code>	
<code>\Glsymbol</code>	491	235, 238–240, 243–245, 389, 390,
<code>\glssymbol</code>	490	393, 395, 396, 546, 547, 549–552, 554, 555	
<code>\glssymbolaccessdisplay</code>		<code>\GLSuseri</code>	491
.....	370–372, 383, 386, 387	<code>\Glsuseri</code>	491
<code>\GLSsymbolplural</code>	491	<code>\gluseri</code>	491
<code>\Glsymbolplural</code>	491	<code>\gluseriaccessdisplay</code>	383, 384
<code>\glsymbolplural</code>	491	<code>\GLSuserii</code>	492
<code>\glsymbolpluralaccessdisplay</code>		<code>\Glsuserii</code>	492
.....	369, 370, 383	<code>\gluserii</code>	492
<code>\glssymbolsgroupname</code>		<code>\gluseriiaccessdisplay</code>	384
.....	36, 43, 223, 425, 427, 429, 540	<code>\GLSuseriii</code>	492
<code>\glstarget</code>	221, 284–294, 302–305,	<code>\Glsuseriii</code>	492
313, 315, 316, 320, 322–324, 327, 328,		<code>\gluseriii</code>	492
330, 332, 333, 342–354, 538, 539, 572, 573		<code>\gluseriiiaccessdisplay</code>	384
<code>\GLStext</code>	488	<code>\GLSuseriv</code>	493
<code>\Glstext</code>	488	<code>\Glsuseriv</code>	492
<code>\glstext</code>	488	<code>\gluseriv</code>	492
<code>\glstextaccessdisplay</code> 370, 371, 374, 375, 380		<code>\gluserivaccessdisplay</code>	385
<code>\glstextformat</code>	118, 120, 467, 475, 476	<code>\GLSuserv</code>	493
<code>\glstextup</code>	41, 395, 427, 543	<code>\Glsuserv</code>	493
<code>\glstildechar</code> 51, 172, 173, 434, 504, 507, 508		<code>\gluserv</code>	493
<code>\glstranslatefalse</code>	28, 419, 420	<code>\gluservaccessdisplay</code>	385
<code>\glstranslatetrue</code>	28, 29, 420	<code>\GLSuservi</code>	493
<code>\glstreechildpredesc</code>	327, 328	<code>\Glsuservi</code>	493
<code>\glstreegroupheaderfmt</code>		<code>\gluservi</code>	493
.....	307–312, 327–330, 334	<code>\gluserviaccessdisplay</code>	385
<code>\glstreeindent</code> ..	328, 330, 332, 333, 348–350	<code>\glswrallowprimitivemodstrue</code> ..	194, 522
<code>\glstreeitem</code>	307, 308, 326	<code>\glswrite</code>	170–175, 183,
<code>\glstreenamebox</code>	332, 333	189, 190, 337–341, 505–509, 515, 519, 520	
<code>\glstreenamefmt</code>	325, 327, 328, 330–333	<code>\glswritedefhook</code>	80, 451, 456
<code>\glstreenavigationfmt</code>		<code>\glswriteentry</code>	191, 521
.....	308–312, 327, 329, 330, 334	<code>\glswritefiles</code>	34, 189, 423, 519
<code>\glstreepredesc</code>	327, 328, 330	<code>\glsxindyfalse</code>	30, 421
<code>\glstreesubitem</code>	307, 326	<code>\glsxindytrue</code>	31, 422
<code>\glstreesubsubitem</code>	307, 326		
<code>\glstype</code>	118, 119, 132–136, 150–		
157, 378–380, 475, 476, 485–488, 494–498			
<code>\glsucmarkfalse</code>	12, 410		
<code>\glsucmarktrue</code>	12, 410		
<code>\glsunexpandedfieldvalue</code>	287		
<code>\glsunset</code>			
. 97, 100–102, 132–137, 462–465, 485–488			

H

<code>\H</code>	24, 417
<code>\hangindent</code>	
311, 312, 326, 328, 330, 332–334, 347–350	
<code>\hbox</code>	97, 286, 290, 343, 462, 573
<code>\hfill</code>	286, 290, 343, 573
<code>\hline</code>	292–295, 302–306, 314–325

<code>\hspace</code>	286, 290, 291, 301, 312, 313, 319, 573	<code>\ifdef</code>	64,
<code>\hss</code>	286, 290, 343, 573		65, 73, 79, 80, 97, 116, 157, 161, 186,
<code>\ht</code>	300		187, 229, 275, 287, 326, 364, 441, 442,
<code>\hyperbf</code>	542		446, 450, 462, 475, 498, 500, 517, 518, 543
<code>\hyperdef</code>	39, 427	<code>\ifdefempty</code>	20, 32, 48, 59, 63–
<code>\hyperemph</code>	543		65, 69, 106, 110, 112, 182, 184, 211, 212,
<code>\hyperit</code>	543		233, 234, 241, 250, 254, 256, 259, 368,
<code>\hyperlink</code>	116, 130, 227, 475, 484, 542		372, 375, 390, 414, 423, 431, 432, 439,
<code>\hypermd</code>	542		441, 442, 444, 467, 470, 472, 515, 516,
<code>hyperref</code> package	197, 200, 225, 226, 272		532, 533, 545, 546, 551, 557, 560, 562, 564
<code>\hyperrm</code>	542	<code>\ifdefequal</code>	31,
<code>\hypersc</code>	543		32, 62–65, 75–77, 81, 91, 95, 212, 365,
<code>\hypersf</code>	542		422, 440–442, 448, 449, 451, 458, 461, 533
<code>\hypersl</code>	543	<code>\ifdefstrequal</code>	87, 456
<code>\hypertarget</code>	130, 484	<code>\ifdefstring</code>	11, 42, 66, 180–
<code>\hypertt</code>	542		182, 184, 208, 209, 211, 213, 356, 409,
<code>\hyperup</code>	543		428, 443, 513, 514, 516, 530, 531, 533, 534
		<code>\ifdefvoid</code> .	23, 24, 94, 212, 416, 461, 533, 534
		<code>\ifdim</code>	237, 300, 331, 548
		<code>\iffalse</code>	94, 99, 460, 463
		<code>\IfFileExists</code>	11,
			27, 28, 181, 182, 203, 409, 419, 420, 514, 527
		<code>\ifglossaryexists</code>	
			46, 57, 62, 179–182, 430, 438–440, 512–514
		<code>\ifgl@sanitize@description</code>	25, 417
		<code>\ifgl@sanitize@name</code>	25, 418
		<code>\ifgl@sanitize@symbol</code>	25, 418
		<code>\ifgl@xindy@gl@numbers</code>	58, 438
		<code>\ifgl@acr@description</code>	264, 568
		<code>\ifgl@acr@dua</code>	
		 252, 259, 262, 264, 558, 564, 566, 568
		<code>\ifgl@acr@footnote</code>	118, 264, 475, 568
		<code>\ifgl@acr@synonym</code>	18, 413
		<code>\ifgl@acr@shortcuts</code>	38, 247, 426, 555
		<code>\ifgl@acr@smallcaps</code>	251–253, 255,
			256, 258, 261, 262, 558, 559, 562, 564, 566
		<code>\ifgl@acr@smaller</code>	
			251–253, 255, 256, 258, 558, 559, 562, 564
		<code>\ifgl@automake</code>	33, 184, 423, 516
		<code>\ifgl@desc@suppressed</code>	283, 441
		<code>\ifgl@sentry@counter</code>	
		 12, 13, 38, 217, 218, 410, 426, 537
		<code>\ifgl@sentry@exists</code>	
			. 61, 78, 80, 89, 91, 439, 440, 450, 456, 458
		<code>\ifgl@esc@locations</code>	193, 522
		<code>\ifgl@field@dcseq</code>	456
		<code>\ifgl@field@defeq</code>	455
		<code>\ifgl@field@eq</code>	455
		<code>\ifgl@has@children</code>	283, 342, 440

I

<code>\IeC</code>	24, 416
<code>\if</code>	122, 196, 336, 477, 478, 523
<code>\if@endfor</code>	281
<code>\if@gl@debug</code>	5, 21, 190, 406, 415, 520
<code>\if@gl@docloaded</code> ...	5, 17, 35, 405, 412, 424
<code>\if@gl@is@acronymlist</code>	118, 414, 475
<code>\if@openright</code>	49, 432
<code>\ifbool</code>	18, 29,
	34, 61, 107, 108, 413, 421, 424, 439, 468, 469
<code>\ifbool@expr</code>	42, 66, 223, 428, 443, 540
<code>\ifcase</code>	5, 8, 9, 28, 36, 72,
	215, 327, 347, 406–408, 420, 424, 446, 535
<code>\ifcsdef</code>	27, 43, 49, 75, 76,
	83–88, 115, 191, 205, 206, 208, 210, 224,
	235, 364, 419, 429, 432, 448, 453–456,
	473, 474, 520, 528, 529, 531, 532, 541, 546
<code>\ifcsempy</code>	63, 275, 441
<code>\ifcsequal</code>	63, 441
<code>\ifcsname</code>	87
<code>\ifcsstrequal</code>	88, 456
<code>\ifcsstring</code>	87, 455
<code>\ifcsundef</code>	8, 15, 31, 39, 42, 45, 47,
	49, 60, 67, 69, 71, 90, 92, 93, 101, 116,
	120, 121, 130, 180, 181, 189, 200, 204,
	206, 214, 218, 223–226, 228, 234, 282,
	341, 365–368, 408, 411, 421, 427, 428,
	430–432, 439, 443–445, 457, 459, 460,
	464, 474, 476, 477, 484, 513, 514, 520,
	525, 527, 529, 535, 538, 540, 541, 543, 546

<code>\ifglshasdesc</code>	283, 441	<code>\ifnum</code> .	15, 33, 101, 102, 182, 211, 300, 328, 330, 332, 348, 349, 411, 423, 465, 515, 533
<code>\ifglshasfield</code>	441	<code>\ifstrempy</code>	7, 342, 407
<code>\ifglshaslong</code>	103–105, 158, 236, 237, 241, 243, 257, 387, 388, 390, 393, 441, 466, 467, 499, 547, 548, 551, 553, 562	<code>\ifstrequal</code>	223, 540
<code>\ifglshasparent</code> .	206, 212, 331, 440, 529, 533	<code>\ifthenelse</code>	26, 39, 49, 120, 183, 223, 264, 281, 419, 427, 432, 476, 515, 540, 568
<code>\ifglshasprefix</code>	276–278	<code>\IfTrackedLanguage</code>	176, 509
<code>\ifglshasprefixfirst</code>	276, 277, 279	<code>\IfTrackedLanguageFileExists</code>	42, 404, 405, 428
<code>\ifglshasprefixfirstplural</code> .	276, 278, 279	<code>\iftrue</code>	94, 99, 460, 463
<code>\ifglshasprefixplural</code>	276, 278, 279	<code>\ifundef</code>	12, 13, 67, 78, 90, 170, 174, 183, 410, 443, 450, 457, 505, 508, 515
<code>\ifglshasshort</code>	441	<code>\ifvmode</code>	167, 504
<code>\ifglshassymbol</code>	250, 254, 259, 327, 328, 330, 332, 333, 441, 557, 560, 564	<code>\ifvoid</code>	301
<code>\ifglshyperfirst</code>	118, 475	<code>\ifx</code>	12, 14, 16, 19, 32, 37, 39, 50, 52, 54, 55, 58, 59, 91– 93, 96, 120, 121, 124–129, 158, 159, 170, 173, 175, 176, 178, 189, 193, 195–199, 202, 223, 224, 226, 227, 248, 249, 251, 253, 256, 258, 259, 261, 263, 265, 266, 337–339, 341, 342, 347–350, 355, 365, 410–414, 425–427, 433–436, 438, 439, 458–462, 476, 477, 479–484, 499, 505, 507, 509–512, 520, 522–526, 540–542, 556, 558, 560, 561, 563, 564, 566, 567, 569
<code>\ifglsindeonlyfirst</code>	191, 521	<code>\immediate</code>	32, 33, 78, 80, 103, 180–182, 189, 190, 204, 423, 450, 451, 466, 513, 514, 520, 527, 528
<code>\ifglsnogroupskip</code> ...	285, 288, 291, 293, 294, 297, 298, 302, 304, 305, 313, 315, 317, 320, 322, 324, 327, 328, 330, 333, 572	<code>\in@</code>	37, 426
<code>\ifglsnonumberlist</code>	216, 536	<code>\indexname</code>	37, 425
<code>\ifglsnopostdot</code>	12, 409	<code>\indexspace</code>	285, 288, 307–312, 327–330, 333, 334, 572
<code>\ifglsnnumberline</code>	49, 432	<code>\input</code>	41, 105, 428, 467
<code>\ifglssanitizesort</code>	23, 26, 416, 418	<code>inputenc package</code>	211
<code>\ifglssavenumberlist</code>	75, 184, 199, 448, 516, 525	<code>\inputencodingname</code>	31, 421
<code>\ifglssavewrites</code> .	34, 179, 191, 423, 512, 520	<code>\InputIfFileExists</code>	79, 450
<code>\ifglssubentrycounter</code>	13, 38, 217, 218, 410, 426, 536, 537	<code>\istfile</code>	519
<code>\ifglstoc</code>	49, 432	<code>\istfilename</code>	43, 170, 174, 181–183, 337, 340, 429, 505, 508, 513, 515
<code>\ifglstranslate</code>	41, 419, 428	<code>\item</code>	285–290, 307, 308, 326–328, 342, 343, 347, 572, 573
<code>\ifglsucmark</code>	47, 48, 431		
<code>\ifglssused</code>	103, 106–112, 118, 167, 191, 250, 254, 256, 259, 276–279, 369–375, 439, 466–472, 475, 504, 521, 557, 560, 562, 564		
<code>\ifglswrallowprimitivemods</code>	195, 522		
<code>\ifglsxindy</code>	43– 45, 50–53, 55–58, 68, 95, 96, 123, 168– 170, 176, 180, 181, 196, 198, 204, 335– 337, 421, 429, 430, 433–438, 444, 461, 478, 504, 505, 509, 513, 514, 523, 524, 527		
<code>\ifignoredglossary</code>	90, 94, 191, 200, 444, 458, 460, 521, 525		
<code>\ifin@</code>	37, 426		
<code>\ifinlistcs</code>	209, 214, 532, 535		
<code>\ifinner</code>	6, 407		
<code>\ifKV@glslink@hyper</code>	119, 120, 476		
<code>\ifKV@glslink@local</code>	132–137, 485–488		
<code>\ifmeasuring@</code>	97, 462		
<code>\ifmmode</code>	6, 407		

J

<code>\jobname</code>	32, 44, 78–80, 170, 174, 180–182, 203, 337, 340, 423, 429, 450, 505, 508, 513–515, 527
-----------------------------	--

K

<code>\key@ifundefined</code>	82, 83, 452
-------------------------------------	-------------

\KV@glslink@hyperfalse		\makebox . 286, 290, 331–333, 343, 349, 350, 573
..... 116, 118, 119, 130, 475, 476, 484		makeglossaries 30, 44, 57, 58, 67, 176, 183, 204
\KV@glslink@hypertrue .. 116, 131, 475, 485		\makeglossaries
		7, 8, 31–36,
		40, 72, 179, 184, 186, 188, 204, 407, 408,
		422–425, 427, 446, 512, 515–517, 519, 528
		\makeglossary
		34, 36, 424, 425
		makeindex
		574
		makeindex 10, 14, 30, 33, 40, 44, 46, 50,
		66–68, 70, 96, 121, 125, 168, 171, 174–
		176, 179, 190, 194–197, 200, 222, 336, 337
		delim_n
		46
		delim_r
		46
		page_compositor
		44
		special characters
		123, 124, 168
		\makenoidxglossaries ... 7, 8, 31, 32, 72,
		184, 188, 407, 408, 422, 446, 516, 517, 519
		\MakeTextUppercase
		4, 405
		\MakeUppercase
		370–372, 379, 380
		\marginpar
		6, 407
		\markboth
		47, 431
		\mbox 167, 285, 289, 311, 312, 331, 343, 504, 572
		memoir class
		190
		\memUHead
		47, 431
		\MessageBreak
		21,
		35, 66, 202, 355, 404, 405, 415, 424, 443, 526
		mfirstuc package
		1, 158
		\mfirstucMakeUppercase
		4, 47,
		48, 84, 107, 109–114, 138–149, 151, 153,
		155, 157, 234, 241, 242, 244, 254, 260,
		278, 279, 373–377, 380–385, 391, 392,
		394, 405, 431, 454, 468, 470–473, 488–
		494, 496–498, 545, 546, 551–553, 560, 565
		\midrule
		297, 298
		\month
		170, 174, 337, 340, 505, 508
		multicol package
		307
		N
		\n
		174, 175, 340, 508, 509
		\NeedsTeXFormat
	 4, 273, 286, 335, 341, 355, 404, 405
		\new@glossaryentry .. 78, 186, 449, 450, 517
		\new@ifnextchar 66, 83, 84, 103, 104,
		131–135, 137–156, 229–232, 275–279,
		443, 453, 454, 466, 467, 485–498, 543, 544
		\newacronym
		228, 233, 249,
		251, 253, 256, 258, 261, 263, 265, 543,
		545, 556, 558, 559, 561, 563, 566, 567, 569
		\newacronymhook
		233, 249,
		251, 253, 256, 258, 261, 263, 265, 396,
		545, 556, 558, 560, 562, 563, 566, 567, 569
		L
\L	25, 417	
\l	25, 417	
\label	9, 215, 217, 408, 535, 537	
\language	30, 421	
\leaders	286, 290, 343, 573	
\leavevmode	88, 119, 456, 476	
\let .. 5, 6, 11, 14–18, 24, 25, 27, 28, 32–34,		
36–39, 42, 43, 52, 53, 64–66, 75, 77, 78,		
88, 89, 91, 93, 94, 97, 99, 100, 102, 105,		
117–120, 122, 130–137, 150–156, 158,		
159, 165, 166, 174, 175, 179–188, 191,		
192, 195, 198–203, 212, 214, 217, 221,		
233, 246–248, 251, 253–263, 273, 281,		
282, 287, 300, 307, 308, 326, 341, 359,		
360, 378–380, 397–401, 404–406, 409,		
411–413, 416, 417, 419, 420, 422–429,		
434, 435, 442, 448, 449, 456–460, 462–		
464, 466, 467, 475, 476, 478, 484–488,		
494–499, 503, 508, 509, 512–514, 516–		
522, 525–527, 533–536, 539, 545, 555–567		
\letcs	32, 62–65, 80, 82,	
83, 86, 92, 93, 157, 158, 180, 181, 186,		
187, 205, 207, 211, 212, 219, 223, 331,		
440–442, 450, 452, 455, 459, 498, 499,		
513, 514, 517, 518, 528–530, 533, 538, 540		
link text	106	
\listcsadd	209, 532	
\listcsgadd	214, 535	
\listcsxadd	205, 528, 529	
\listeadd	209, 531	
\listxadd	78	
\loadglsentries	105, 467	
\long	88, 227, 456, 542	
\longnewglossaryentry	89, 456, 457	
\longprovideglossaryentry	456, 457	
longtable package	290, 297, 301	
\LT@end@open	300	
\LT@err	300	
\LT@foot	300, 301	
\LT@head	301	
\LT@lastfoot	300, 301	
\LT@output	300	
		M
\makeatletter	79, 203, 450, 527	

<code>\newacronymstyle</code>	236–241, 243–245, 546–555	<code>see</code>	7, 10, 72, 77, 183, 185
<code>\newcommand</code>	...	<code>short</code>	112, 163, 358, 364
...	6–24, 26, 27, 29, 30, 32–41, 43–53,	<code>shortaccess</code>	362, 367, 396
...	55–63, 65–69, 71–89, 94–106, 110, 112,	<code>shortpl</code>	364
...	114, 115, 117–120, 122, 123, 130–169,	<code>shortplural</code>	163, 358
...	175, 176, 178–182, 184, 185, 188–193,	<code>shortpluralaccess</code>	362, 367
...	195–201, 203, 205–211, 213–225, 227–	<code>sort</code>	70, 162, 190, 222
...	235, 237, 245, 247–271, 274–282, 284,	<code>symbol</code>	70, 71, 143,
...	285, 287, 300, 307, 325, 326, 331, 341,	...	251–253, 255, 261, 294, 316, 357, 359, 364
...	357, 361–365, 387, 396, 401–403, 407–	<code>symbolaccess</code>	362, 366
...	416, 418, 419, 421, 423–457, 460–464,	<code>symbolplural</code>	144, 357
...	466, 467, 470, 472–476, 478, 484–505,	<code>symbolpluralaccess</code>	362, 366
...	509, 512–516, 519–534, 536–548, 555–572	<code>text</code>	70, 71, 131, 137, 159, 251, 255, 357
<code>\newcount</code>	15, 16, 75, 411, 412, 448	<code>textaccess</code>	361, 365
<code>\newcounter</code>	12, 13, 410	<code>type</code>	17, 71, 105, 161
<code>\newenvironment</code>	218, 538	<code>user1</code>	144, 162, 358
<code>\newglossary</code>	17, 18, 36,	<code>user1access</code>	363, 367
...	37, 68, 183, 412, 413, 425, 443, 444, 515, 516	<code>user2</code>	145, 162, 358
<code>\newglossaryentry</code>	7, 37, 74, 75, 78, 100,	<code>user2access</code>	363, 367
...	233, 247, 250, 252, 254, 257, 260, 262,	<code>user3</code>	146, 162, 358
...	265, 397–400, 407, 425, 447, 449, 464,	<code>user3access</code>	363, 367
...	545, 555, 557, 559, 561, 562, 565, 566, 569	<code>user4</code>	147, 163, 359
<code>\newglossaryentry options</code>		<code>user4access</code>	363, 368
<code>access</code>	359, 361	<code>user5</code>	148, 163, 359
<code>counter</code>	71	<code>user5access</code>	363, 368
<code>description</code>	...	<code>user6</code>	148, 163, 359
	29, 70, 75, 77, 89, 141, 159, 229, 258, 358	<code>user6access</code>	363, 368
<code>descriptionaccess</code>	362, 366	<code>\newglossarystyle</code>	282, 285–300,
<code>descriptionplural</code>	142, 358	...	302–324, 326–331, 333, 334, 541, 572, 573
<code>descriptionpluralaccess</code>	362, 366	<code>\newif</code>	4, 5, 19, 27,
<code>entrycounter</code>	215	...	30, 33, 194, 405, 406, 414, 419, 421, 423, 522
<code>first</code>	71, 92, 131, 138, 160, 255, 261, 357	<code>\newignoredglossary</code>	444
<code>firstaccess</code>	362, 366	<code>\newlength</code>	130, 286,
<code>firstplural</code>	71, 139, 161, 357	...	290, 291, 301, 312, 313, 319, 329, 484, 573
<code>firstpluralaccess</code>	362, 366	<code>\newrobustcmd</code>	7, 62–64, 77, 78, 84,
<code>format</code>	170	...	103, 104, 118, 131–156, 158–164, 166,
<code>long</code>	112, 164, 358	...	167, 198, 199, 229–232, 274–279, 331,
<code>longaccess</code>	363, 367	...	365–368, 407, 440, 441, 449, 450, 453,
<code>longplural</code>	164, 358	...	454, 466, 467, 475, 485–504, 525, 543, 544
<code>longpluralaccess</code>	363, 367	<code>\newterm</code>	37, 425
<code>name</code>	70, 74, 77, 89, 140, 158, 199, 357, 359	<code>\newtoks</code>	123, 179, 232, 233, 479, 512, 545
<code>nonumberlist</code>	72, 73	<code>\newwrite</code>	32, 78, 170,
<code>parent</code>	72, 77	...	174, 179, 183, 423, 450, 505, 508, 512, 515
<code>plural</code>	70, 92, 139, 357	<code>\nfss@text</code>	6, 407
<code>pluralaccess</code>	362, 366	<code>ngerman package</code>	175
<code>prefix</code>	273	<code>\noalign</code>	300
<code>prefixfirst</code>	273	<code>\nobreak</code>	285, 289, 301, 343, 572
<code>prefixfirstplural</code>	274	<code>\noexpand</code>	19,
<code>prefixplural</code>	274	...	39, 51, 52, 79, 93, 94, 115, 120–123, 129,

165, 176–178, 180–182, 184, 192, 193, 196, 200, 204, 207, 219, 221, 228, 233, 247, 248, 250, 252, 254, 255, 257, 260, 262, 264, 265, 335, 355, 356, 397–401, 414, 427, 433, 434, 450, 460, 461, 473, 474, 476–478, 483, 484, 503, 510–514, 516, 521–523, 525, 527–530, 538, 539, 543, 545, 555–557, 559, 561–563, 565–569	\or 5, 6, 8, 9, 28, 36, 215, 327, 347, 406, 408, 420, 425, 535
\nohyperpage 226, 541	\org@glossaryentrynumbers 202, 217, 527, 536
\noindent 221, 308–312, 329, 330, 539	\org@glossarytitle 202, 526
\noist 340, 341, 512	\org@glspostdescription 43, 429
\nopostdesc 37, 43, 88, 202, 342, 425, 429, 456, 527	\org@ifKV@glslink@hyper 119, 120, 476
\normalbaselineskip 300	\outputpenalty 300
\ns@ACRfull 231, 544	P
\ns@Acrfull 230, 543	\p@ 285, 286, 307, 325, 326, 572
\ns@acrfull 229, 543	\p@glshyp@opt 117, 475
\ns@ACRfullpl 232, 544	package options:
\ns@Acrfullpl 231, 544	acronym 17, 18, 40, 201, 229
\ns@acrfullpl 231, 544	true 18
\ns@ACRlong 154, 496, 497	counter 20
\ns@Acrlong 153, 154, 496	debug
\ns@acrlong 153, 496	showaccsupp 7
\ns@ACRlongpl 156, 498	showtargets 6, 7
\ns@Acrlongpl 155, 156, 497	description 255, 256
\ns@acrlongpl 155, 497	disablemakegloss 32
\ns@ACRshort 150, 151, 494	dua 254–256
\ns@Acrshort 150, 494	entrycounter 12, 215, 217, 287
\ns@acrshort 149, 493	true 13
\ns@ACRshortpl 152, 495	esclocations 597
\ns@Acrshortpl 152, 495	false 10
\ns@acrshortpl 151, 495	footnote 132–136, 252, 254, 255, 258
\ns@newglossary 67, 443	hyperfirst
\null 122– 129, 176–178, 203, 478–484, 510–512, 527	false 132–136
\number 15, 79, 92, 102, 192, 195, 221, 356, 411, 450, 459, 466, 521, 522, 539	index 37
\numberline 49, 432	indexonlyfirst 581
\numexpr 102, 466	kernelglossredefs
O	nowarn 35
\O 25, 417	makeindex 172, 272
\o 25, 417	nogroupskip 291, 293, 294, 297, 298, 302, 304, 305, 313, 315, 317, 320, 322, 324
\OE 24, 417	nolist 266
\oe 25, 417	nolong 266, 290
\oldacronym 543	nomain 17
\openout 32, 78, 170, 174, 180, 337, 340, 423, 450, 505, 508, 513	nonumberlist 10
\OR 264, 568	nosuper 266
	notree 266
	nowarn 5
	numberline 8
	record 275
	sanitize 25, 70, 158, 159
	sanitizesort 22
	savewrites 34, 578
	false 179

true	183, 189	nopostdot	215
section	8, 48	numberedsection	215
sort		style	214
def	14, 15	subentrycounter	215
none	14	title	214
standard	14	toctitle	214
use	14, 15, 597	type	17, 199, 214
style	9, 266	\printindex	37, 425
subentrycounter	13, 215, 217	\printnoidxglossaries	186, 517, 526
toc	8	\printnoidxglossary	185, 186, 188, 201,
true	8	208, 209, 216, 517, 519, 526, 530, 531, 536	
translate	28	\printnoidxglossary options	
false	27	sort	216
translator	27	\printnumbers	37, 425
xindy	30, 31, 172, 272	\printsymbols	36, 425
\PackageError	7, 8, 17,	\ProcessOptions	273, 355
33, 40, 51, 57, 61, 62, 66, 71, 74, 75, 82–		\ProcessOptionsX	38, 426
88, 90–92, 100, 116, 157, 178, 179, 183,		\protect	49, 114, 236, 237, 243,
186, 188, 208–210, 214, 216, 224, 225,		250, 254, 257, 377, 378, 387, 388, 393,	
234, 235, 252, 253, 259, 262, 341, 368,		394, 432, 473, 547, 548, 553, 557, 560, 562	
407, 408, 412, 423, 427, 433, 438–440,		\protected@csdef	85, 454
443, 445, 447, 448, 452–458, 464, 474,		\protected@csxdef	85, 454
498, 512, 516, 517, 519, 530–532, 535,		\protected@eappto	32, 69, 95, 192
536, 540, 541, 546, 547, 558, 559, 564, 566		\protected@edef	7,
\PackageInfo		9, 16, 19, 32, 52, 54, 56, 58, 59, 62, 67,	
..... 5, 6, 32, 180, 190, 406, 422, 513, 520		69, 85–91, 94–96, 107, 108, 114, 119,	
\PackageWarning	5, 6, 21, 406, 415	121, 165, 167, 185, 191, 193, 196, 200,	
\PackageWarningNoLine	5, 6, 21, 404–406, 415	205, 206, 215, 217, 219, 221, 224, 233,	
\pagegoal	300	259, 265, 274, 280, 336, 337, 355, 356,	
\pagelistname	43, 293–295,	364, 365, 396, 407, 408, 434, 435, 437,	
298, 304–306, 315–319, 322–325, 427, 429		439, 458, 461, 468, 469, 473, 477, 503,	
\par	43, 221, 285, 286, 289, 307, 309–312, 325,	520, 522, 523, 535, 538–541, 545, 564, 569	
326, 328–334, 343, 348–350, 429, 539, 572		\protected@write	66, 67, 170–
\parindent		172, 183, 185, 188, 191, 200, 281, 337,	
307–312, 326, 328–330, 332–334, 348–350		442, 443, 505, 506, 515, 517, 519, 521, 525	
\parskip	307–310, 326, 328, 329	\protected@xdef	14–
\PassOptionsToPackage	273, 355	16, 19, 24, 76, 91, 94, 96, 97, 196, 203,	
\penalty	300	360, 361, 411–413, 416, 448, 461, 462, 523	
\phantomsection	48, 432	\providecommand	18, 31, 40,
polyglossia package	27, 41	41, 48, 66, 103, 131, 172, 183, 186, 189,	
\printacronyms	413	204, 219, 220, 275, 285, 286, 307, 325,	
\PrintChanges	405	356, 413, 422, 427, 431, 442, 466, 485,	
\printglossaries	184, 516, 526	506, 515, 517, 519, 527, 528, 538, 539, 572	
\printglossary	18, 21, 36, 37, 184,	\provideglossaryentry	449, 450
201, 216, 413, 415, 425, 516, 525, 526, 536		\ProvidesFile	41, 428
\printglossary options		\ProvidesGlossariesLang	428
entrycounter	215	\ProvidesPackage	4, 273,
nogroupskip	215	280, 282, 285, 286, 290, 297, 301, 307,	
nonumberlist	216	312, 319, 325, 335, 341, 355, 404, 405, 572	

R		\s@GlsSetXdyFirstLetterAfterDigits	
\r	24, 417	168, 169, 504
\raggedright	299, 300, 302–306, 320–325	\s@GlsSetXdyNumberGroupOrder	169, 504, 505
\raisebox	130, 484	\s@ifglossaryexists	60, 200, 439, 526
\ref	218, 537	\s@newglossary	67, 443
\refstepcounter	217, 537	\savecounters@	120, 477
\relax 5, 9, 11, 15–18, 28, 32, 33, 36–38, 53, 66, 71, 72, 76, 79, 91, 93, 97, 101, 102, 117, 118, 120, 122, 124–129, 158, 159, 172–176, 178, 179, 182, 183, 185–187, 192, 196, 198, 199, 202, 203, 211, 212, 214, 215, 223, 224, 266, 281, 285, 286, 300, 307, 311, 312, 325, 327, 328, 330– 334, 336, 339–341, 347–350, 359, 360, 378–380, 397–400, 406, 408, 409, 411– 413, 419, 420, 423–426, 435, 442, 445, 446, 449, 450, 458, 460, 462, 465, 466, 475, 477–483, 499, 507–512, 515–518, 521, 523–527, 533, 535, 540, 541, 569, 572		\seename	198, 427, 525
\renewacronymstyle	387–390, 393, 395, 396, 546	\setabbreviationstyle	263
\renewcommand	4–6, 8–11, 13, 14, 17, 18, 20–22, 26, 28–31, 33, 34, 36, 38, 42–45, 58, 69, 72, 73, 88, 100– 102, 165, 167, 169, 175, 177, 182–187, 203, 204, 215, 233, 234, 236–240, 242– 245, 249, 251–253, 256, 258, 261, 263, 265, 283–295, 297, 298, 300, 302–317, 320–324, 326–336, 341–349, 351–354, 356, 359, 360, 368, 372, 375, 377, 378, 386–390, 392–396, 398–400, 405–415, 418, 420–423, 425, 426, 428–430, 438, 444, 446, 456, 464–466, 503–505, 509, 510, 515–518, 527, 528, 535, 545–556, 558, 559, 561–564, 566, 567, 569, 572, 573	\SetAcronymLists	414
\renewenvironment	219, 283, 285, 287, 291–296, 299, 300, 302–326, 328, 329, 331, 538, 572	\SetAcronymStyle	29, 30, 421, 568
\renewglossarystyle	541	\setacronymstyle	263, 546
\RequireGlossariesLang 42, 404, 405, 427, 428		\setbool	26, 418
\RequirePackage	4, 10, 11, 27, 28, 38, 41, 266, 272, 273, 290, 297, 301, 307, 312, 319, 356, 404, 405, 409, 419, 420, 426, 428, 569, 571	\setbox	300, 301
\restorecounters@	121, 477	\setcounter	217, 536, 537
\romannumeral 192, 195, 331–333, 349, 521, 522		\SetCustomDisplayStyle	265, 568, 569
S		\SetCustomStyle	569
\s@glshyp@opt	117, 475	\SetDefaultAcronymDisplayStyle	249, 555, 556
		\SetDefaultAcronymStyle	264, 556, 568
		\SetDescriptionAcronymDisplayStyle	256, 560–562
		\SetDescriptionAcronymStyle 264, 561, 568	
		\SetDescriptionDUAAcronymDisplayStyle	253, 254, 558, 560
		\SetDescriptionDUAAcronymStyle	264, 559, 568
		\SetDescriptionFootnoteAcronymDisplayStyle	251, 252, 557, 558
		\SetDescriptionFootnoteAcronymStyle	264, 558, 568
		\SetDUADisplayStyle	263, 566–568
		\SetDUASyle	264, 567, 568
		\setentrycounter	52, 172, 213, 335, 434, 507, 534, 540
		\SetFootnoteAcronymDisplayStyle ...	258, 562, 563
		\SetFootnoteAcronymStyle ...	264, 563, 568
		\SetGenericNewAcronym	235, 545, 546
		\setglossarypreamble	430
		\setglossarysection	431
		\setglossarystyle 202, 224, 266, 285, 286, 288–293, 295–300, 302–325, 327, 329, 330, 333, 334, 526, 540, 541, 569, 572, 573	
		\setglossentrycompatibility	214, 224, 535, 539, 541
		\setkeys	27, 31, 38, 48, 90, 119, 167, 202, 233, 249, 251, 253, 256, 258, 261, 263, 265, 419,

422, 426, 431, 457, 476, 503, 504, 526, 545, 556, 558, 560, 561, 563, 566, 567, 569	178, 180–184, 186, 188, 199, 202, 204, 218, 224, 230, 236, 237, 239, 240, 242– 245, 254, 259, 282, 284, 285, 288, 289, 291, 302, 313, 320, 326–328, 330, 332, 333, 335, 336, 338–340, 342–345, 347– 351, 353, 377, 378, 387–390, 392, 394– 397, 407, 408, 422–424, 427, 433, 436– 439, 446, 447, 464, 466, 467, 473–475, 502, 505–508, 512–517, 519, 525, 526, 528, 537, 541, 543, 547–555, 560, 564, 572
\setlength 286, 290, 291, 301, 307–310, 312, 313, 319, 326, 328, 329, 333, 349, 350, 573	\spacefactor 12, 409
\SetSmallAcronymDisplayStyle 261, 564, 566	\SS 25, 417
\SetSmallAcronymStyle 264, 566, 568	\ss 25, 417
\setstylefile 429	\string 7, 8, 17, 21, 31– 36, 40, 50, 51, 54–56, 58, 66, 67, 72, 74, 75, 80, 83, 84, 95, 96, 100, 101, 103–105, 115, 117, 121, 122, 124–126, 128, 129, 165, 168–176, 178, 179, 183–186, 188, 189, 196–198, 202, 204, 208, 209, 216, 221, 224, 263, 281, 335–341, 407, 408, 412, 415, 422–424, 427, 432–439, 442, 443, 446, 447, 450, 451, 453, 461, 462, 464, 466, 467, 473–475, 477–481, 483, 502, 504–509, 511, 512, 515–517, 519, 523, 524, 526–528, 530, 531, 536, 539, 541
\settoheight 130, 484	\strut 221, 285, 286, 288–291, 293, 294, 302, 304, 305, 313, 315, 316, 320, 322, 324, 330, 342–346, 348, 351–354, 539, 572, 573
\settowidth 237, 331–333, 349, 548	\subglossentry 95, 203, 212, 221, 284–286, 288–291, 293, 294, 302, 303, 305, 313, 315, 316, 320, 322, 323, 327, 328, 330, 332, 461, 527, 533, 539, 572, 573
\setupglossaries 426	\subitem 307, 326, 327, 347
\sfcode 12, 409	\subsubitem 307, 326, 327, 347
\show 266–271, 402, 403, 569–571	supertabular package 11, 266, 312, 319
\showacronymlists 571	\symbolname 43, 295, 298, 306, 317–319, 324, 325, 427, 429
\showglocounter 570	
\showglodesc 570	T
\showglodescplural 570	\t 24, 417
\showglofield 571	\tablehead 313–325
\showglofirst 570	\tabletail 313–325
\showglofirstpl 570	\tabularnewline 291–295, 297, 298, 302–306, 313–325, 345, 346, 351, 352
\showgloflag 571	\texorpdfstring 161, 500
\showgloindex 571	\textbar 282
\showglolevel 570	\textbf 221, 227, 325, 347–350, 539, 542
\showgloloclist 571	textcase package 4
\showglolong 571	\textit 227, 543
\showglongname 570	\textmd 227, 542
\showgloparent 569	
\showgloplural 570	
\showgloshort 571	
\showglosort 570	
\showglossaries 571	
\showglossarycounter 571	
\showglossaryentries 571	
\showglossaryin 571	
\showglossaryout 571	
\showglossarytitle 571	
\showglosymbol 571	
\showglosymbolplural 571	
\showglotext 570	
\showglotype 570	
\showglouseri 570	
\showglouserii 570	
\showglouseriii 570	
\showglouseriv 570	
\showglouserv 570	
\showglouservi 570	
\SmallNewAcronymDef 261, 565, 566	
\space 7, 8, 31–36, 40, 51, 55, 56, 58, 72, 74, 75, 100, 101, 103–105, 114, 115, 117, 165, 170–174,	

<code>\textrm</code>	227, 542		
<code>\textsc</code>	228, 238, 244, 252, 256, 258, 261, 395, 543, 549, 554, 558, 562, 564, 566		
<code>\textsf</code>	227, 542		
<code>\textsl</code>	228, 543		
<code>\textsmaller</code>	238, 244, 252, 256, 258, 261, 395, 549, 554, 558, 562, 564, 566		
<code>\texttt</code>	227, 542		
<code>\textulc</code>	229, 543		
<code>\textup</code>	228, 229, 543		
<code>\TH</code>	25, 417		
<code>\th</code>	25, 417		
<code>\the</code>	39, 42, 51, 56, 58, 67, 124–129, 170, 174, 176, 178, 189, 191, 195, 200, 211, 219, 221, 227, 233, 234, 236, 237, 242, 243, 247, 248, 250, 252–255, 257, 260, 262, 264, 265, 335, 337, 340, 356, 387, 388, 392, 393, 396–401, 427, 428, 434, 437, 439, 443, 479–484, 505, 508, 510–512, 520, 522, 525, 533, 538, 539, 542, 545–548, 552, 553, 555–557, 559, 561–563, 565–569		
<code>\the@numberlist</code>	165, 166, 503		
<code>\theglossary</code>	5, 405		
<code>\theglossaryentry</code>	12, 218, 410, 537		
<code>\theglossarysubentry</code>	13, 218, 410, 537		
<code>\theglentrycounter</code>	120, 121, 193, 196, 336, 337, 476, 477, 522, 523		
<code>\theH</code>	197, 524		
<code>\theHglossaryentry</code>	12, 410		
<code>\theHglossarysubentry</code>	13, 410		
<code>\theHglentrycounter</code>	121, 193, 196, 477, 522, 523		
<code>\thesection</code>	39, 427		
<code>\this@dialect</code>	42, 404, 405, 428		
<code>\tiny</code>	6, 407		
<code>\toks@</code>	39, 42, 51, 56, 58, 67, 124– 129, 176, 178, 199, 200, 219–221, 227, 335, 355, 356, 427, 428, 434, 437, 439, 443, 479–484, 510–512, 525, 538, 539, 542		
<code>\toprule</code>	297, 298		
<code>tracklang</code> package	<u>41</u> , <u>404</u>		
<code>\trans@languages</code>	42, 428		
<code>\translate</code>	42, 43, 428, 429		
<code>\translatelet</code> ...	17, 18, 36, 37, 412, 413, 425		
<code>translator</code> package .	<u>17</u> , <u>18</u> , <u>27</u> , <u>36</u> , <u>37</u> , <u>41</u> , <u>42</u> , <u>200</u>		
<code>\triangleright</code>	6, 407		
<code>\ttfamily</code>	7, 407		
<code>\TX@trial</code>	97, 462		
<code>\typeout</code>	21, 415		
U			
<code>\u</code>	24, 417		
<code>\uccode</code>	211, 533		
<code>\undef</code>	73, 79, 200, 446, 450, 525		
<code>\unskip</code>	88, 286, 290, 343, 456, 573		
<code>\unvbox</code>	300, 301		
<code>\usedictionary</code>	42, 428		
<code>\usepackage</code>	208, 209, 530, 531		
<code>\UTFviii@two@octets</code>	25, 417		
<code>\UTFviii@two@octets@combine</code>	25, 417		
V			
<code>\v</code>	24, 417		
<code>\vbox</code>	300, 301		
<code>\vsize</code>	301		
<code>\vskip</code>	285, 286, 300, 307, 325, 572		
<code>\vss</code>	300, 301		
W			
<code>\warn@nomakeglossaries</code>	183, 185, 186, 515–517		
<code>\warn@noprintglossary</code>	184–186, 203, 516, 517, 525, 527		
<code>\write</code>	33, 80, 103, 170–175, 180–182, 186, 189, 204, 337–341, 423, 450, 451, 466, 505–509, 513, 514, 517, 520, 527, 528		
<code>\writeist</code>	179, 341, 505, 508, 509, 512		
X			
<code>\x</code>	542		
<code>\xatlevel@</code>	120, 476		
<code>\xcapitalisewords</code>	500		
<code>\xdef</code>	92, 281, 458–460, 527		
<code>\xglaccsupp</code>	365		
<code>\xifinlist</code>	78, 79		
<code>\xifinlistcs</code>	205, 206, 209, 528, 529, 531		
<code>xindy</code>	574		
<code>xindy</code>	10, 14, 24, 30, 31, 33, 44, 45, 50, 53, 55, 57, 58, 96, 128, 129, 168– 171, 190, 194, 196, 200, 204, 222, 272, 336		
<code>\xmakefirstuc</code>	107–109, 114, 157, 159, 274, 468, 469, 473, 498, 499		
<code>\xspace</code>	228, 543		
<code>xspace</code> package	<u>4</u> , <u>228</u>		
Y			
<code>\year</code>	170, 174, 337, 340, 505, 508		
Z			
<code>\z@</code>	300, 301		