

# **glossaries-extra.sty v1.19: documented code**

Nicola L.C. Talbot

Dickimaw Books

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

2017-09-08

## Abstract

This is the documented code for the glossaries-extra package. See [glossaries-extra-manual.pdf](#) for the user manual.

This package is experimental and not stable. It's provided for testing purposes only. Future versions may not be compatible with this version. Once it has stabilised I'll add it to CTAN, at which point compatibility with the first stable version will be maintained.

# Contents

|   |            |
|---|------------|
| <b>1 Main Package Code (glossaries-extra.sty)</b>           | <b>5</b>   |
| 1.1 Package Initialisation and Options                      | 5          |
| 1.2 Extra Utilities   | 21         |
| 1.3 Modifications to Commands Provided by glossaries        | 27         |
| 1.3.1 Existence Checks                                      | 31         |
| 1.3.2 Document Definitions                                  | 39         |
| 1.3.3 Existing Glossary Style Modifications                 | 43         |
| 1.3.4 Entry Formatting, Hyperlinks and Indexing             | 47         |
| 1.3.5 Entry Counting  | 81         |
| 1.3.6 Acronym Modifications                                 | 94         |
| 1.3.7 Indexing and Displaying Glossaries                    | 96         |
| 1.4 Integration with glossaries-accsupp                     | 119        |
| 1.5 Categories  | 130        |
| 1.6 Abbreviations   | 153        |
| 1.6.1 Abbreviation Styles Setup                             | 172        |
| 1.6.2 Predefined Styles (Default Font)                      | 175        |
| 1.6.3 Predefined Styles (Small Capitals)                    | 190        |
| 1.6.4 Predefined Styles (Fake Small Capitals)               | 203        |
| 1.6.5 Predefined Styles (Emphasized)                        | 217        |
| 1.6.6 Predefined Styles (User Parentheses Hook)             | 237        |
| 1.6.7 Predefined Styles (Hyphen)                            | 246        |
| 1.6.8 Predefined Styles (No Short on First Use)             | 261        |
| 1.7 Using Entries in Headings                               | 263        |
| 1.8 Multi-Lingual Support                                   | 280        |
| <b>2 Style Adjustments (glossaries-extra-stylemods.sty)</b> | <b>282</b> |
| 2.1 Package Initialisation                                  | 282        |
| 2.2 List-Like Styles  | 283        |
| 2.3 Longtable Styles  | 283        |
| 2.4 Long Ragged Styles                                      | 284        |
| 2.5 Supertabular Styles                                     | 286        |
| 2.6 Super Ragged Styles                                     | 287        |
| 2.7 Inline Style  | 288        |
| 2.8 Tree Styles   | 288        |
| <b>Glossary</b>   | <b>300</b> |
| <b>Change History</b>                                       | <b>301</b> |



# 1 Main Package Code (glossaries-extra.sty)

## 1.1 Package Initialisation and Options

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{glossaries-extra}[2017/09/08 v1.19 (NLCT)]
```

Requires xkeyval to define package options.

```
3 \RequirePackage{xkeyval}
```

Requires etoolbox package.

```
4 \RequirePackage{etoolbox}
```

Has glossaries already been loaded?

```
5 \@ifpackageloaded{glossaries}
```

```
6 {%
```

Already loaded so pass any options to \setupglossaries. This means that the options that can only be set when glossaries is loaded can't be used.

```
7 \newcommand{\glxstr@doption}[1]{\setupglossaries{#1}}%
```

```
8 \let\@glxstr@declareoption\@gls@declareoption
```

```
9 }
```

```
10 {%
```

Not already loaded, so pass options to glossaries.

```
11 \newcommand{\glxstr@doption}[1]{%
```

```
12 \PassOptionsToPackage{#1}{glossaries}%
```

```
13 }%
```

Set the defaults.

```
14 \PassOptionsToPackage{toc}{glossaries}
```

```
15 \PassOptionsToPackage{nopostdot}{glossaries}
```

```
16 \PassOptionsToPackage{noredefwarn}{glossaries}
```

```
17 \@ifpackageloaded{polyglossia}%
```

```
18 {}%
```

```
19 {%
```

```
20 \@ifpackageloaded{babel}%
```

```
21 {\PassOptionsToPackage{translate=babel}{glossaries}}%
```

```
22 {}%
```

```
23 }%
```

```
24 \newcommand*{\@glxstr@declareoption}[2]{%
```

```
25 \DeclareOptionX{#1}{#2}%
```

```
26 \DeclareOption{#1}{#2}%
```

```
27 }
```

```
28 }
```

Declare package options.

`\glstrundefaction` Determines what to do if an entry hasn't been defined. The two arguments are the error or warning message and the help message if an error should be produced.

```

29 \newcommand*{\glstrundefaction}[2]{%
30   \@glstrundeftag\PackageError{glossaries-extra}{#1}{#2}%
31 }

```

`\warnonexistsordo` If user wants `undefaction=warn`, then `glossaries v4.19` is required.

```

32 \newcommand*{\glstr@warnonexistsordo}[1]{%

```

`\glstrundeftag` Text to display when an entry doesn't exist.

```

33 \newcommand*{\glstrundeftag}{??}
34 \newcommand*{\@glstrundeftag}{}

```

This text is switched on at the start of the document to prevent unwanted text inserted into the preamble if any tests are made before the start of the document.

`\warn@undefaction` This is how `\glstrundefaction` should behave if `undefaction=warn` is set.

```

35 \newcommand*{\@glstr@warn@undefaction}[2]{%
36   \@glstrundeftag\GlossariesExtraWarning{#1}%
37 }

```

`\err@undefaction` This is how `\glstrundefaction` should behave if `undefaction=error` is set.

```

38 \newcommand*{\@glstr@err@undefaction}[2]{%
39   \@glstrundeftag\PackageError{glossaries-extra}{#1}{#2}%
40 }

```

`\warn@onexistsordo` This is how `\glstr@warnonexistsordo` should behave if `undefaction=warn` is set.

```

41 \newcommand*{\@glstr@warn@onexistsordo}[1]{%
42   \GlossariesExtraWarning{\string#1\space hasn't been defined, so
43     some errors won't be converted to warnings.
44     (This most likely means your version of
45     glossaries.sty is below version 4.19.)}%
46 }

```

`\f@for@gl@sentries`

```

47 \newcommand*{\@glstr@redef@for@gl@sentries}{}

```

`\f@for@gl@sentries`

```

48 \newcommand*{\@glstr@do@redef@for@gl@sentries}{%
49   \renewcommand*{\for@gl@sentries}[3][\gl@defaulttype]{%
50     \edef\@glo@list{\csname glolist@##1\endcsname}%
51     \ifdefstring{\@glo@list}{,}%
52     {%
53       \GlossariesExtraWarning{No entries defined in glossary '#1'}%
54     }%
55     {%
56       \@for##2:=\@glo@list\do

```

```

57      {%
58      \ifdefempty{##2}{-}{##3}%
59      }%
60      }%
61      }%
62 }%

63 \define@choicekey{glossaries-extra.sty}{undefaction}[\val\nr]%
64 {warn,error}%
65 {%
66   \ifcase\nr\relax
67     \let\glxstrundefaction\@glxstr@warn@undefaction
68     \let\glxstr@warnonexistssordo\@glxstr@warn@onexistssordo
69     \let\@glxstr@redef@for\glxstr@do@redef@for\glxstr@sentries
70   \or
71     \let\glxstrundefaction\@glxstr@err@undefaction
72     \let\glxstr@warnonexistssordo\@gobble
73     \let\@glxstr@redef@for\glxstr@relax
74   \fi
75 }

```

To assist bib2gls, v1.08 introduces the record option, which will write information to the aux file whenever an entry needs to be indexed.

\@glxstr@record Does nothing by default.

```
76 \newcommand*{\@glxstr@record}[3]{}
```

\glxstr@recordsee Does nothing by default.

```
77 \newcommand*{\glxstr@recordsee}[2]{}
```

\glxstr@defaultnumberformat

```
78 \newcommand*{\@glxstr@defaultnumberformat}{\glxstr@numberformat}%
```

\glxstr@setDefaultNumberFormat

```

79 \newcommand*{\GlsXtrSetDefaultNumberFormat}[1]{%
80   \renewcommand*{\@glxstr@defaultnumberformat}{#1}%
81 }%

```

@@glxstr@record This is the actual code that does the recording The first argument is the option list (as passed in the first optional argument to commands like \gls). This allows the noindex setting to be picked up. The third argument is the key family (glslink in most cases, glossadd for \glsadd).

```

82 \newcommand*{\@@glxstr@record}[3]{%
83   \begingroup
84     \let\@glxstr@numberformat\@glxstr@defaultnumberformat
85     \def\@glxstr@thevalue{}%
86     \def\@glxstr@theHvalue{\@glxstr@thevalue}%
87     \let\@glxstr@org@theHvalue\@glxstr@theHvalue
88     \ifcsdef{glo@#2@counter}%

```

```

89  {%
90    \edef\@gls@counter{\csname glo@#2@counter\endcsname}%
91  }%
92  {%

```

Entry hasn't been defined, so we'll have to assume the page number by default.

```

93    \def\@gls@counter{page}%
94  }%
95  \setkeys{#3}{#1}%
96  \ifKV@glslink@noindex
97  \else
98    \glswriteentry{#2}%
99  {%

```

Check if thevalue has been set.

```

100    \ifdefempty{\@glxtr@thevalue}%
101    {%

```

Key thevalue hasn't been set, but check if theHvalue has been set. (Not particularly likely, but allow for it.)

```

102    \ifx\@glxtr@org@theHvalue\@glxtr@theHvalue
103    \else
104    \let\theHglentrycounter\@glxtr@theHvalue
105    \fi

```

Save the entry counter.

```

106    \glxtr@saveentrycounter

```

Temporarily redefine \@do@wrglossary for use with \glxtr@do@wrglossary.

```

107    \let\@do@wrglossary\@glxtr@dorecord
108  }%
109  {%

```

thevalue has been set, so there's no need to defer writing the location value. (If it's dependent on the page counter, the counter key should be set instead.)

```

110    \let\theglentrycounter\@glxtr@thevalue
111    \let\theHglentrycounter\@glxtr@theHvalue
112    \let\@do@wrglossary\@glxtr@dorecordnodefer
113  }%
114  \ifx\@glxtr@record@setting\@glxtr@record@setting@alsoindex
115    \glxtr@do@wrglossary{#2}%
116  \else

```

No need to escape special characters, but need to save the label.

```

117    \edef\@gls@label{\glstetoklabel{#2}}%
118    \@do@wrglossary
119  \fi
120  }%
121  \fi
122 \endgroup
123 }

```



glsxtr@dorecord If record=alsoindex is used, then \@glslocref may have been escaped, but this isn't appropriate here.

```

124 \newcommand*\@glsxtr@dorecord{%
125   \global\let\@glsrecordlocref\theglsentrycounter
126   \let\@glsxtr@orgprefix\@glo@counterprefix
127   \ifx\theglsentrycounter\theHglentrycounter
128     \def\@glo@counterprefix{}%
129   \else
130     \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
131       {\theglsentrycounter}{\theHglentrycounter}}%
132    }%
133     \@do@gls@getcounterprefix
134   \fi
135   \protected@write\@auxout{\let\@glsrecordlocref\relax}{\string\glsxtr@record
136     {\@gls@label}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}}%
137     {\@glsrecordlocref}}%
138   \@glsxtr@counterrecordhook
139   \let\@glo@counterprefix\@glsxtr@orgprefix
140 }

```

dorecordnodefer As above, but don't defer expansion of location. This uses \theglsentrycounter directly for the location rather than \@glslocref since there's no need to guard against premature expansion of the page counter.

```

141 \newcommand*\@glsxtr@dorecordnodefer{%
142   \ifx\theglsentrycounter\theHglentrycounter
143     \protected@write\@auxout{}{\string\glsxtr@record
144       {\@gls@label}{\@gls@counter}{\@glsnumberformat}}%
145       {\theglsentrycounter}}%
146   \else
147     \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
148       {\theglsentrycounter}{\theHglentrycounter}}%
149    }%
150     \@do@gls@getcounterprefix
151     \protected@write\@auxout{}{\string\glsxtr@record
152       {\@gls@label}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}}%
153       {\theglsentrycounter}}%
154   \fi
155   \@glsxtr@counterrecordhook
156 }

```

r@recordcounter

```

157 \newcommand*\@@glsxtr@recordcounter{%
158   \@glsxtr@noop@recordcounter
159 }

```

p@recordcounter

```

160 \newcommand*\@glsxtr@noop@recordcounter[1]{%
161   \PackageError{glossaries-extra}{\string\GlsXtrRecordCounter\space

```

```

162   requires record=only or record=alsoindex package option}{}%
163 }

p@recordcounter
164 \newcommand*{\@glxtr@op@recordcounter}[1]{%
165   \eappto\@glxtr@counterrecordhook{\noexpand\@glxtr@docounterrecord{#1}}%
166 }

lsxtr@recordsee Deal with \glssee in record mode.
167 \newcommand*{\@glxtr@recordsee}[2]{%
168   \def\@gls@xref{#2}%
169   \@onelevel@sanitize\@gls@xref
170   \protected@write\@auxout{}\@string\glxtr@recordsee{#1}{\@gls@xref}}%
171 }

srtglossaryunit
172 \newcommand{\printunsrtglossaryunit}{%
173   \print@noop@unsrtglossaryunit
174 }

tr@setup@record Initialise.
175 \newcommand*{\glxtr@setup@record}{}

saveentrycounter Only store the entry counter information if the indexing is on.
176 \newcommand*{\glxtr@indexonly@saveentrycounter}{%
177   \ifKV@glslink@noindex
178   \else
179     \glxtr@saveentrycounter
180   \fi
181 }

addloclistfield
182 \newcommand*{\glxtr@addloclistfield}{%
183   \key@ifundefined{glossentry}{loclist}%
184   {%
185     \define@key{glossentry}{loclist}{\def\@glo@loclist{##1}}%
186     \appto\@gls@keymap{,{loclist}{loclist}}%
187     \appto\@newglossaryentryprehook{\def\@glo@loclist{}}%
188     \appto\@newglossaryentryposthook{%
189       \gls@assign@field{\@glo@label}{loclist}{\@glo@loclist}%
190     }%
191     \glssetnoexpandfield{loclist}%
192   }%
193   {%
194     \key@ifundefined{glossentry}{location}%
195     {%
196       \define@key{glossentry}{location}{\def\@glo@location{##1}}%

```

```

197 \appto\@gls@keymap{,{location}{location}}%
198 \appto\@newglossaryentryprehook{\def\@glo@location{}}%
199 \appto\@newglossaryentryposthook{%
200 \gls@assign@field{\@glo@label}{location}{\@glo@location}%
201 }%
202 \glssetnoexpandfield{location}%
203 }%
204 {}%

```

Add a key to store the group heading.

```

205 \key@ifundefined{glossentry}{group}%
206 {%
207 \define@key{glossentry}{group}{\def\@glo@group{##1}}%
208 \appto\@gls@keymap{,{group}{group}}%
209 \appto\@newglossaryentryprehook{\def\@glo@group{}}%
210 \appto\@newglossaryentryposthook{%
211 \gls@assign@field{\@glo@label}{group}{\@glo@group}%
212 }%
213 \glssetnoexpandfield{group}%
214 }%
215 {}%
216 }

```

@record@setting Keep track of the record package option.

```

217 \newcommand*{\@glsxtr@record@setting}{off}

```

ttting@alsoindex

```

218 \newcommand*{\@glsxtr@record@setting@alsoindex}{alsoindex}

```

Now define the record package option.

```

219 \define@choicekey{glossaries-extra.sty}{record}[\val\nr]%
220 {off,only,alsoindex}%
221 [only]%
222 {%
223 \let\@glsxtr@record@setting\val
224 \ifcase\nr\relax

```

Don't record.

```

225 \def\glsxtr@setup@record{%
226 \renewcommand*{\@do@seeglossary}{\@glsxtr@org@doseeglossary}%
227 \renewcommand*{\@glsxtr@record}[3]{}%
228 \let\@do@wrglossary\glsxtr@do@wrglossary
229 \let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
230 \let\glsxtrundefaction\@glsxtr@err@undefaction
231 \let\glsxtr@warnonexistsordo\@gobble
232 \let\@glsxtr@recordcounter\@glsxtr@noop@recordcounter
233 \def\printunsrtglossaryunit{\print@noop@unsrtglossaryunit}%
234 \undef\glsxtrsetaliasnoindex
235 }%
236 \or

```

Only record (don't index).

```
237 \def\glxtr@setup@record{%
238   \@glxtr@autoseeindexfalse
239   \let\@do@seeglossary\@glxtr@recordsee
240   \let\@glxtr@record\@glxtr@record
241   \let\@do@wrglossary\@gobble
242   \let\@glxtr@saveentrycounter\relax
243   \let\glxtrundefaction\@glxtr@warn@undefaction
244   \let\glxtr@warnonexistssordo\@glxtr@warn@onexistssordo
245   \glxtr@addloclistfield
246   \renewcommand*{\@glxtr@autoindexcrossrefs}{}%
247   \let\@glxtr@recordcounter\@glxtr@op@recordcounter
248   \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
```

Switch off the index suppression for aliased entries. (bib2gls will deal with them.)

```
249 \def\glxtrsetaliasnoindex{}%
```

`\@glxtr@setupsort@none` was only introduced to glossaries v4.30, so it may not be available. If it's defined, use it to remove the unnecessary overhead of escaping and sanitizing the sort value.

```
250 \ifdef\@glxtr@setupsort@none{\@glxtr@setupsort@none}{}%
251 }%
252 \or
```

Record and index.

```
253 \def\glxtr@setup@record{%
254   \renewcommand*{\@do@seeglossary}{\@glxtr@org@doseeglossary}%
255   \let\@glxtr@record\@glxtr@record
256   \let\@do@wrglossary\glxtr@do@wrglossary
257   \let\@glxtr@saveentrycounter\glxtr@indexonly@saveentrycounter
258   \let\glxtrundefaction\@glxtr@warn@undefaction
259   \let\glxtr@warnonexistssordo\@glxtr@warn@onexistssordo
260   \glxtr@addloclistfield
261   \let\@glxtr@recordcounter\@glxtr@op@recordcounter
262   \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
263   \undef\glxtrsetaliasnoindex
264 }%
265 \fi
266 }
```

Version 1.06 changes the `docdef` option to a choice rather than boolean setting. The available values are: false, true or restricted. The restricted option permits document definitions as long as they occur before the first glossary is displayed.

`\glxtr@docdefval` The `docdef` value is stored as an integer: 0 (false), 1 (true) and 2 (restricted).

```
267 \newcount\@glxtr@docdefval
```

Need to provide conditional commands that are backward compatible:

`\if@glxtrdocdef`

```
268 \newcommand*{\if@glxtrdocdef}{\ifnum\@glxtr@docdefval>0 }
```

lsxtrdocdeftrue

```
269 \newcommand*{\@glxtrdocdeftrue}{\@glxtr@docdefval=1 }
```

lsxtrdocdeffalse

```
270 \newcommand*{\@glxtrdocdeffalse}{\@glxtr@docdefval=0 }
```

By default don't allow entries to be defined in the document to encourage the user to define them in the preamble, but if the user is really determined to define them in the document allow them to request this.

```
271 \define@choicekey{glossaries-extra.sty}{docdef}{\val\nr}%  
272 {false,true,restricted}[true]%  
273 {%  
274   \@glxtr@docdefval=\nr\relax  
275   \ifnum\@glxtr@docdefval=2\relax  
276     \renewcommand*{\@glsdofexistsorwarn}{\glsdofexists}%  
277   \fi  
278 }
```

ocdefrestricted

```
279 \newcommand*{\if@glxtrdocdefrestricted}{\ifnum\@glxtr@docdefval=2 }
```

oifexistsorwarn

Need an error to notify user if an undefined entry is being referenced in the glossary for the docdef=restricted option. This is used by \glossentryname (but not by \glossentrydesc etc as one error per entry is sufficient).

```
280 \newcommand*{\@glsdofexistsorwarn}{\glsdofexistsorwarn}
```

indexcrossrefs

Automatically index cross references at the end of the document

```
281 \define@boolkey{glossaries-extra.sty}[@glxtr]{indexcrossrefs}[true]{%  
282   \if@glxtrindexcrossrefs  
283   \else  
284     \renewcommand*{\@glxtr@autoindexcrossrefs}{}%  
285   \fi  
286 }
```

Switch off since this can increase the build time.

```
287 \@glxtrindexcrossrefsfalse
```

But allow see key to switch it on automatically.

oindexcrossrefs

```
288 \newcommand*{\@glxtr@autoindexcrossrefs}{\@glxtrindexcrossrefstrue}
```

autoseeindex

Provide a boolean option to allow the user to prevent the automatic indexing of the cross-referencing keys see, seealso and alias.

```
289 \define@boolkey{glossaries-extra.sty}[@glxtr@]{autoseeindex}[true]{%  
290 }  
291 \@glxtr@autoseeindextrue
```

iesExtraWarning Allow users to suppress warnings.

```

292 \newcommand*{\GlossariesExtraWarning}[1]{\PackageWarning{glossaries-extra}{#1}}

```

raWarningNoLine Allow users to suppress warnings.

```

293 \newcommand*{\GlossariesExtraWarningNoLine}[1]{%
294   \PackageWarningNoLine{glossaries-extra}{#1}}

295 \@glsxtr@declareoption{nowarn}{%
296   \let\GlossariesExtraWarning\@gobble
297   \let\GlossariesExtraWarningNoLine\@gobble
298   \glsxtr@doption{nowarn}%
299 }

postdot Shortcut for nopostdot=false
300 \@glsxtr@declareoption{postdot}{%
301   \glsxtr@doption{nopostdot=false}%
302 }

glsxtrabbrvtype Glossary type for abbreviations.
303 \newcommand*{\glsxtrabbrvtype}{\glsdefaulttype}

abbreviationsdef Set by abbreviations option.
304 \newcommand*{\@glsxtr@abbreviationsdef}{}

abbreviationsdef
305 \newcommand*{\@glsxtr@doabbreviationsdef}{%
306   \ifpackageloaded{babel}%
307     {\providecommand{\abbreviationsname}{\acronymname}}%
308     {\providecommand{\abbreviationsname}{Abbreviations}}%
309     \newglossary[glg-abr]{abbreviations}{gls-abr}{glo-abr}{\abbreviationsname}%
310     \renewcommand*{\glsxtrabbrvtype}{abbreviations}%
311     \newcommand*{\printabbreviations}[1][1]{%
312       \printglossary[type=\glsxtrabbrvtype,##1]%
313     }%
314     \disable@keys{glossaries-extra.sty}{abbreviations}%

    If the acronym option hasn't been used, change \acronymtype to \glsxtrabbrvtype.
315     \ifglsacronym
316     \else
317       \renewcommand*{\acronymtype}{\glsxtrabbrvtype}%
318     \fi
319 }%

abbreviations If abbreviations, create a new glossary type for abbreviations.
320 \@glsxtr@declareoption{abbreviations}{%
321   \let\@glsxtr@abbreviationsdef\@glsxtr@doabbreviationsdef
322 }

```

`\GlsXtrDefineAbbreviationShortcuts` Enable shortcut commands for the abbreviations. Unlike the analogous command provided by glossaries, this uses `\newcommand` instead of `\let` as a safety feature.

```

323 \newcommand*{\GlsXtrDefineAbbreviationShortcuts}{%
324   \newcommand*{\ab}{\cgl{s}}%
325   \newcommand*{\abp}{\cgl{s}pl}%
326   \newcommand*{\as}{\gl{s}xtrshort}%
327   \newcommand*{\asp}{\gl{s}xtrshortpl}%
328   \newcommand*{\al}{\gl{s}xtrlong}%
329   \newcommand*{\alp}{\gl{s}xtrlongpl}%
330   \newcommand*{\af}{\gl{s}xtrfull}%
331   \newcommand*{\afp}{\gl{s}xtrfullpl}%
332   \newcommand*{\Ab}{\cGl{s}}%
333   \newcommand*{\Abp}{\cGl{s}pl}%
334   \newcommand*{\As}{\Gl{s}xtrshort}%
335   \newcommand*{\Asp}{\Gl{s}xtrshortpl}%
336   \newcommand*{\Al}{\Gl{s}xtrlong}%
337   \newcommand*{\Alp}{\Gl{s}xtrlongpl}%
338   \newcommand*{\Af}{\Gl{s}xtrfull}%
339   \newcommand*{\Afp}{\Gl{s}xtrfullpl}%
340   \newcommand*{\AB}{\cGL{S}}%
341   \newcommand*{\ABP}{\cGL{S}pl}%
342   \newcommand*{\AS}{\GL{S}xtrshort}%
343   \newcommand*{\ASP}{\GL{S}xtrshortpl}%
344   \newcommand*{\AL}{\GL{S}xtrlong}%
345   \newcommand*{\ALP}{\GL{S}xtrlongpl}%
346   \newcommand*{\AF}{\GL{S}xtrfull}%
347   \newcommand*{\AFP}{\GL{S}xtrfullpl}%
348   \newcommand*{\newabbr}{\newabbreviation}%

```

Disable this command after it's been used.

```

349 \let\GlsXtrDefineAbbreviationShortcuts\relax
350 }

```

`\GlsXtrDefineAcShortcuts` Enable shortcut commands for the abbreviations, but uses the analogous commands provided by glossaries.

```

351 \newcommand*{\GlsXtrDefineAcShortcuts}{%
352   \newcommand*{\ac}{\cgl{s}}%
353   \newcommand*{\acp}{\cgl{s}pl}%
354   \newcommand*{\acs}{\gl{s}xtrshort}%
355   \newcommand*{\acsp}{\gl{s}xtrshortpl}%
356   \newcommand*{\acl}{\gl{s}xtrlong}%
357   \newcommand*{\aclp}{\gl{s}xtrlongpl}%
358   \newcommand*{\acf}{\gl{s}xtrfull}%
359   \newcommand*{\acfp}{\gl{s}xtrfullpl}%
360   \newcommand*{\Ac}{\cGl{s}}%
361   \newcommand*{\Acp}{\cGl{s}pl}%
362   \newcommand*{\Acs}{\Gl{s}xtrshort}%
363   \newcommand*{\Acsp}{\Gl{s}xtrshortpl}%
364   \newcommand*{\Acl}{\Gl{s}xtrlong}%

```

```

365 \newcommand*{\Ac1p}{\Glsxtrlongpl}%
366 \newcommand*{\Acf}{\Glsxtrfull}%
367 \newcommand*{\Acfp}{\Glsxtrfullpl}%
368 \newcommand*{\AC}{\cGLS}%
369 \newcommand*{\ACP}{\cGLSp1}%
370 \newcommand*{\ACS}{\Glsxtrshort}%
371 \newcommand*{\ACSP}{\Glsxtrshortpl}%
372 \newcommand*{\ACL}{\Glsxtrlong}%
373 \newcommand*{\ACLP}{\Glsxtrlongpl}%
374 \newcommand*{\ACF}{\Glsxtrfull}%
375 \newcommand*{\ACFP}{\Glsxtrfullpl}%
376 \newcommand*{\newabbr}{\newabbreviation}%

```

Disable this command after it's been used.

```

377 \let\GlsXtrDefineAcShortcuts\relax
378 }

```

**@OtherShortcuts** Similarly provide shortcut versions for the commands provided by the symbols and numbers options.

```

379 \newcommand*{\GlsXtrDefineOtherShortcuts}{%
380 \newcommand*{\newentry}{\newglossaryentry}%
381 \ifdef\printsymbols
382 {%
383 \newcommand*{\newsym}{\glxtrnewsymbol}%
384 }{}%
385 \ifdef\printnumbers
386 {%
387 \newcommand*{\newnum}{\glxtrnewnumber}%
388 }{}%
389 \let\GlsXtrDefineOtherShortcuts\relax
390 }

```

Always use the long forms, not the shortcuts, where portability is an issue. (For example, when defining entries in a file that may be input by multiple documents.)

**@setupshortcuts** Command used to set the shortcuts option.

```

391 \newcommand*{\@glxtr@setupshortcuts}{}

```

**@shortcutsval** Store the value of the shortcuts option. (Needed by bib2gls.)

```

392 \newcommand*{\@glxtr@shortcutsval}{\ifglssacrshortcuts acro\else none\fi}%

```

Provide shortcuts option. Unlike the glossaries version, this is a choice rather than a boolean key but it also provides `shortcuts=true` and `shortcuts=false`, which are equivalent to `shortcuts=all` and `shortcuts=none`. Multiple use of this option in the *same* option list will override each other. New to v1.17: `shortcuts=ac` which implements `\GlsXtrDefineAcShortcuts` (not included in `shortcuts=all` as it conflicts with other shortcuts).

```

393 \define@choicekey{glossaries-extra.sty}{shortcuts}[\val\nr]%
394 {acronyms,acro,abbreviations,abbr,other,all,true,none,false,ac}[true]{%

```



```

395 \let\@glxtr@shortcutsval\val
396 \ifcase\nr\relax % acronyms
397   \renewcommand*\@glxtr@setupshortcuts}{%
398     \glsacrshortcutstrue
399     \DefineAcronymSynonyms
400   }%
401 \or % acro
402   \renewcommand*\@glxtr@setupshortcuts}{%
403     \glsacrshortcutstrue
404     \DefineAcronymSynonyms
405   }%
406 \or % abbreviations
407   \renewcommand*\@glxtr@setupshortcuts}{%
408     \GlsXtrDefineAbbreviationShortcuts
409   }%
410 \or % abbr
411   \renewcommand*\@glxtr@setupshortcuts}{%
412     \GlsXtrDefineAbbreviationShortcuts
413   }%
414 \or % other
415   \renewcommand*\@glxtr@setupshortcuts}{%
416     \GlsXtrDefineOtherShortcuts
417   }%
418 \or % all
419   \renewcommand*\@glxtr@setupshortcuts}{%
420     \glsacrshortcutstrue
421     \DefineAcronymSynonyms
422     \GlsXtrDefineAbbreviationShortcuts
423     \GlsXtrDefineOtherShortcuts
424   }%
425 \or % true
426   \renewcommand*\@glxtr@setupshortcuts}{%
427     \glsacrshortcutstrue
428     \DefineAcronymSynonyms
429     \GlsXtrDefineAbbreviationShortcuts
430     \GlsXtrDefineOtherShortcuts
431   }%
432 \or % none, false
433   \renewcommand*\@glxtr@setupshortcuts}{}%
434 \or % ac
435   \renewcommand*\@glxtr@setupshortcuts}{%
436     \glsacrshortcutstrue
437     \GlsXtrDefineAcShortcuts
438   }%
439 \fi
440 }

```

lsxtr@doaccsupp

```

441 \newcommand*\@glxtr@doaccsupp}{%

```

accsupp If accsupp, load glossaries-accsupp package.

```
442 \@glsxtr@declareoption{accsupp}{%
443 \renewcommand*{\@glsxtr@doaccsupp}{\RequirePackage{glossaries-accsupp}}}
```

GlossaryWarning Warning text displayed in document if the external glossary file given by the argument is missing.

```
444 \newcommand{\glsxtrNoGlossaryWarning}[1]{%
445 \@glsxtr@defaultnoglossarywarning{#1}%
446 }
```

omissingglstext If true, suppress the text produced if the external glossary file is missing.

```
447 \define@choicekey{glossaries-extra.sty}{nomissingglstext}[\val\nr]{%
448 {true,false}[true]{%
449 \ifcase\nr\relax % true
450 \renewcommand{\glsxtrNoGlossaryWarning}[1]{%
451 \null
452 }%
453 \else % false
454 \renewcommand{\glsxtrNoGlossaryWarning}[1]{%
455 \@glsxtr@defaultnoglossarywarning{#1}%
456 }%
457 \fi
458 }
```

Provide option to load glossaries-extra-stylemods (Deferred to the end.)

xtr@redefstyles

```
459 \newcommand*{\@glsxtr@redefstyles}{}
```

stylemods

```
460 \define@key{glossaries-extra.sty}{stylemods}[default]{%
461 \ifstrequal{#1}{default}%
462 {%
463 \renewcommand*{\@glsxtr@redefstyles}{%
464 \RequirePackage{glossaries-extra-stylemods}}%
465 }%
466 {%
467 \renewcommand*{\@glsxtr@redefstyles}{}%
468 \@for\@glsxtr@tmp:=#1\do{%
469 \IfFileExists{glossary-\@glsxtr@tmp.sty}%
470 {%
471 \eappto\@glsxtr@redefstyles{%
472 \noexpand\RequirePackage{glossary-\@glsxtr@tmp}}%
473 }%
474 {%
475 \PackageError{glossaries-extra}%
476 {Glossaries style package ‘glossary-\@glsxtr@tmp.sty’
477 doesn’t exist (did you mean to use the ‘style’ key?)}%
478 {The list of values (#{1}) in the ‘stylemods’ key should
```

```

479         match the glossary-xxx.sty files provided with
480         glossaries.sty}%
481     }%
482 }%
483 \appto\@glxtr@redefstyles{\RequirePackage{glossaries-extra-stylemods}}%
484 }%
485 }

```

glxtr@do@style

```

486 \newcommand*{\@glxtr@do@style}{%

```

style Since the stylemods option can automatically load extra style packages, deal with the style option after those packages have been loaded.

```

487 \define@key{glossaries-extra.sty}{style}{%

```

Defer actual style change:

```

488 \renewcommand*{\@glxtr@do@style}{%

```

Set this as the default style:

```

489 \setkeys{glossaries.sty}{style={#1}}%

```

Set this style:

```

490 \setglossarystyle{#1}%

```

```

491 }%

```

```

492 }

```

Pass all other options to glossaries.

```

493 \DeclareOptionX*{%

```

```

494 \expandafter\glxtr@doooption\expandafter{\CurrentOption}}

```

Process options.

```

495 \ProcessOptionsX

```

Load glossaries if not already loaded.

```

496 \RequirePackage{glossaries}

```

Load the glossaries-accsupp package if required.

```

497 \@glxtr@doaccsupp

```

g@doseeglossary Save original definition of \@do@seeglossary

```

498 \let\@glxtr@org@doseeglossary\@do@seeglossary

```

@org@gloautosee Save and restore original definition of \@glo@autosee. (That command may not be defined as it was only introduced to glossaries v4.30, in which case the synonym won't be defined either.)

```

499 \let\@glxtr@org@gloautosee\@glo@autosee

```

Check if user tried autoseeindex=false when it can't be supported.

```

500 \if@glxtr@autoseeindex

```

```

501 \else

```

```

502 \ifdef\@glxtr@org@gloautosee
503 {}%
504 {\PackageError{glossaries-extra}{‘autoseeindex=false’ package
505 option requires at least v4.30 of glossaries.sty}%
506 {You need to update the glossaries.sty package}%
507 }
508 \fi

```

\@glo@autosee If \@glo@autosee has been defined (glossaries v4.30 onwards), redefine it to test the autoseeindex option.

```

509 \ifdef\@glo@autosee
510 {%
511 \renewcommand*{\@glo@autosee}{%
512 \if@glxtr@autoseeindex\@glxtr@org@gloautosee\fi}%
513 }%
514 {}

```

checkseeallowed Don't prohibit the use of the see key before the indexing files have been opened if the automatic see indexing has been disabled, since it's no longer an issue.

```

515 \renewcommand*{\gls@checkseeallowed}{%
516 \if@glxtr@autoseeindex\@gls@see@noindex\fi
517 }

```

Define abbreviations glossaries if required.

```

518 \@glxtr@abbreviationsdef
519 \let\@glxtr@abbreviationsdef\relax

```

Setup shortcuts if required.

```

520 \@glxtr@setupshortcuts

```

Redefine \@glxtr@redef@for@gl@sentries if required.

```

521 \@glxtr@redef@for@gl@sentries

```

ariesextrasetup Allow user to set options after the package has been loaded. First modify \glxtr@doooption so that it now uses \setupglossaries:

```

522 \renewcommand{\glxtr@doooption}[1]{\setupglossaries{#1}}%

```

Now define the user command:

```

523 \newcommand*{\glossariesextrasetup}[1]{%
524 \let\glxtr@setup@record\relax
525 \let\@glxtr@setupshortcuts\relax
526 \let\@glxtr@redef@for@gl@sentries\relax
527 \setkeys{glossaries-extra.sty}{#1}%
528 \@glxtr@abbreviationsdef
529 \let\@glxtr@abbreviationsdef\relax
530 \@glxtr@setupshortcuts
531 \glxtr@setup@record
532 \@glxtr@redef@for@gl@sentries
533 }

```

`@do@wrglossary` Save original definition of `\@do@wrglossary`.  
534 `\let\glxtr@do@wrglossary\@do@wrglossary`

`saveentrycounter` Save original definition of `\@gls@saveentrycounter`.  
535 `\let\glxtr@saveentrycounter\@gls@saveentrycounter`

`saveentrycounter` Change `\@gls@saveentrycounter` so that it only stores the entry counter information if the indexing is on.  
536 `\let\@gls@saveentrycounter\glxtr@indexonly@saveentrycounter`

Set up record option if required.  
537 `\glxtr@setup@record`

Disable preamble-only options and switch on the undefined tag at the start of the document.  
538 `\AtBeginDocument{%`  
539 `\disable@keys{glossaries-extra.sty}{abbreviations,docdef,record}%`  
540 `\def\@glxtrundefrag{\glxtrundefrag}%`  
541 `}`

## 1.2 Extra Utilities

`trifemptyglossary` `\glxtrifemptyglossary{<type>}{<true>}{<false>}`

Provide command to determine if any entries have been added to the glossary (where the glossary label is provided in the first argument). The entries are stored in the comma-separated list `\glolist@<type>`. If this hasn't been defined, the glossary doesn't exist. If it has been defined and is simply a comma, the glossary exists and is empty. (It's initialised to a comma.)

```
542 \newcommand{\glxtrifemptyglossary}[3]{%
543   \ifcsdef{glolist@#1}%
544   {%
545     \ifcsstring{glolist@#1}{,}{#2}{#3}%
546   }%
547   {%
548     \glxtrundefaction{Glossary type '#1' doesn't exist}{}%
549     #2%
550   }%
551 }
```

`trifkeydefined` Tests if the key given in the first argument has been defined.

```
552 \newcommand*{\glxtrifkeydefined}[3]{%
553   \key@ifundefined{glossentry}{#1}{#3}{#2}%
554 }
```

providestoragekey Like \glsaddstoragekey but does nothing if the key has already been defined.

```
555 \newcommand*{\glsxtrprovidestoragekey}{%
556   \@ifstar\@sglsxtr@provide@storagekey\@glsxtr@provide@storagekey
557 }
```

vide@storagekey Unstarred version.

```
558 \newcommand*{\@glsxtr@provide@storagekey}[3]{%
559   \key@ifundefined{glossentry}{#1}%
560   {%
561     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
562     \appto\@gls@keymap{,{#1}{#1}}%
563     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
564     \appto\@newglossaryentryposthook{%
565       \letcs{\@glo@tmp}{@glo@#1}%
566       \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
567     }%
568   }
```

Allow the user to omit the user level command if they only intended fetching the value with

\glsxtrusefield

```
568   \ifblank{#3}
569   {%
570   {%
571     \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
572   }%
573 }%
574 {%
```

Provide the no-link command if not already defined.

```
575   \ifblank{#3}
576   {%
577   {%
578     \providecommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
579   }%
580 }%
581 }
```

vide@storagekey Starred version.

```
582 \newcommand*{\s@glsxtr@provide@storagekey}[1]{%
583   \key@ifundefined{glossentry}{#1}%
584   {%
585     \expandafter\newcommand\expandafter*\expandafter
586     {\csname gls@assign@#1@field\endcsname}[2]{%
587       \@gls@expand@field{##1}{#1}{##2}%
588     }%
589   }%
590   {%
591     \@glsxtr@provide@addstoragekey{#1}%
592   }
```

The name of a text-block control sequence can be stored in a field (given by `\GlsXtrFmtField`). This command can then be used with `\glstrfmt[⟨options⟩]{⟨label⟩}{⟨text⟩}` which effectively does `\glslink[⟨options⟩]{⟨label⟩}{⟨cs⟩{⟨text⟩}}` If the field hasn't been set for that entry just `⟨text⟩` is done.

`\GlsXtrFmtField`

```
593 \newcommand{\GlsXtrFmtField}{useri}
```

`tDefaultOptions`

```
594 \newcommand{\GlsXtrFmtDefaultOptions}{noindex}
```

`\glstrfmt` The post-link hook isn't done.

```
595 \newrobustcmd*{\glstrfmt}[3][[]]{%
596   \glsdoifexistsordo{#2}%
597   {%
598     \ifglshasfield{\GlsXtrFmtField}{#2}%
599     {%
600       \let\do@glsl@link@checkfirsthyper\relax
601       \expandafter\@glsl@link\expandafter[\GlsXtrFmtDefaultOptions,#1]{#2}%
602       {\csuse{\glscurrentfieldvalue}{#3}}%
603     }%
604     {#3}%
605   }%
606   {#3}%
607 }
```

`\glstrentryfmt` No link or indexing.

```
608 \ifdef\texorpdfstring
609 {
610   \newcommand*{\glstrentryfmt}[2]{%
611     \texorpdfstring{\@glstrentryfmt{#1}{#2}}{#2}%
612   }
613 }
614 {
615   \newcommand*{\glstrentryfmt}{\@glstrentryfmt}
616 }
```

`@glstrentryfmt`

```
617 \newrobustcmd*{\@glstrentryfmt}[2]{%
618   \glsdoifexistsordo
619   {%
620     \ifglshasfield{\GlsXtrFmtField}{#1}%
621     {%
622       \csuse{\glscurrentfieldvalue}{#2}%
623     }%
624     {#2}%
625   }%
626   {#2}%
627 }
```

`xtrfieldlistadd` If a field stores an etoolbox internal list (e.g. `loclist`) then this macro provides a convenient way of adding to the list via etoolbox's `\listcsadd`. The first argument is the entry's label, the second is the field label and the third is the element to add to the list.

```
628 \newcommand*\glxtrfieldlistadd}[3]{%
629   \listcsadd{glo@\glsdetoklabel{#1}@#2}{#3}%
630 }
```

`trfieldlistgadd` Similarly but uses `\listcsgadd`.

```
631 \newcommand*\glxtrfieldlistgadd}[3]{%
632   \listcsgadd{glo@\glsdetoklabel{#1}@#2}{#3}%
633 }
```

`trfieldlistead` Similarly but uses `\listcseadd`.

```
634 \newcommand*\glxtrfieldlistead}[3]{%
635   \listcseadd{glo@\glsdetoklabel{#1}@#2}{#3}%
636 }
```

`trfieldlistxadd` Similarly but uses `\listcsxadd`.

```
637 \newcommand*\glxtrfieldlistxadd}[3]{%
638   \listcsxadd{glo@\glsdetoklabel{#1}@#2}{#3}%
639 }
```

Now provide commands to iterate over these lists.

`fielddolistloop`

```
640 \newcommand*\glxtrfielddolistloop}[2]{%
641   \dolistcsloop{glo@\glsdetoklabel{#1}@#2}%
642 }
```

`fieldforlistloop`

```
643 \newcommand*\glxtrfieldforlistloop}[3]{%
644   \forlistcsloop{glo@\glsdetoklabel{#1}@#2}{#3}%
645 }
```

List element tests:

`trfieldifinlist` First argument label, second argument field, third argument item, fourth true part and fifth false part.

```
646 \newcommand*\glxtrfieldifinlist}[5]{%
647   \ifinlistcs{#3}{glo@\glsdetoklabel{#1}@#2}{#4}{#5}%
648 }
```

`trfieldxifinlist` Expands item.

```
649 \newcommand*\glxtrfieldxifinlist}[5]{%
650   \xifinlistcs{#3}{glo@\glsdetoklabel{#1}@#2}{#4}{#5}%
651 }
```



`\lsxtrifhasfield` A simpler alternative to `\ifglshasfield` that doesn't complain if the entry or the field doesn't exist. (No mapping is used.) Grouping is added to the unstarred version allow for nested use.

```
652 \newrobustcmd{\glxtrifhasfield}{%
653   \@ifstar{\s@glxtrifhasfield}{\@glxtrifhasfield}%
654 }
```

`\lsxtrifhasfield` Unstarred version adds grouping.

```
655 \newcommand{\@glxtrifhasfield}[4]{%
656   {\s@glxtrifhasfield{#1}{#2}{#3}{#4}}%
657 }
```

`\lsxtrifhasfield` Starred version omits grouping.

```
658 \newcommand{\s@glxtrifhasfield}[4]{%
659   \letcs{\glscurrentfieldvalue}{glo@\glsdetoklabel{#2}@#1}%
660   \ifdef\glscurrentfieldvalue
661     {%
662       \ifdefempty\glscurrentfieldvalue{#4}{#3}%
663     }%
664     {#4}%
665 }
```

`\glxtrusefield` Provide a user-level alternative to `\@gls@entry@field`. The first argument is the entry label. The second argument is the field label.

```
666 \newcommand*{\glxtrusefield}[2]{%
667   \@gls@entry@field{#1}{#2}%
668 }
```

`\Glsxtrusefield` Provide a user-level alternative to `\@Gls@entry@field`.

```
669 \newcommand*{\Glsxtrusefield}[2]{%
670   \@gls@entry@field{#1}{#2}%
671 }
```

`\glxtrdeffield` Just use `\csdef` to provide a field value for the given entry.

```
672 \newcommand*{\glxtrdeffield}[2]{\csdef{glo@\glsdetoklabel{#1}@#2}}
```

`glxstredeffield` Just use `\csedef` to provide a field value for the given entry.

```
673 \newcommand*{\glxstredeffield}[2]{\csedef{glo@\glsdetoklabel{#1}@#2}}
```

`etfieldifexists`

```
674 \newcommand*{\glxtrsetfieldifexists}[3]{\glsdoidexists{#1}{#3}}
```

`\GlsXtrSetField` Allow the user to set a field. First argument entry label, second argument field label, third argument value.

```
675 \newrobustcmd{\GlsXtrSetField}[3]{%
676   \glxtrsetfieldifexists{#1}{#2}%
677   {\csdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
678 }
```

`\GlsXtrLetField` Uses `\cslet` instead. Third argument should be a macro.

```

679 \newrobustcmd*{\GlsXtrLetField}[3]{%
680   \glstrsetfieldifexists{#1}{#2}%
681   {\cslet{glo@\glstoklabel{#1}@#2}{#3}}%
682 }

```

`\csGlsXtrLetField` Uses `\csletcs` instead. Third argument should be a control sequence name.

```

683 \newrobustcmd*{\csGlsXtrLetField}[3]{%
684   \glstrsetfieldifexists{#1}{#2}%
685   {\csletcs{glo@\glstoklabel{#1}@#2}{#3}}%
686 }

```

`LetFieldToField` Sets the field for one entry to the field for another entry. Third argument should be the other entry and the fourth argument that other field label.

```

687 \newrobustcmd*{\GlsXtrLetFieldToField}[4]{%
688   \glstrsetfieldifexists{#1}{#2}%
689   {\csletcs{glo@\glstoklabel{#1}@#2}{glo@\glstoklabel{#3}@#4}}%
690 }

```

`gGlsXtrSetField` Allow the user to set a field. First argument entry label, second argument field label, third argument value.

```

691 \newrobustcmd*{\gGlsXtrSetField}[3]{%
692   \glstrsetfieldifexists{#1}{#2}%
693   {\csgdef{glo@\glstoklabel{#1}@#2}{#3}}%
694 }

```

`xGlsXtrSetField`

```

695 \newrobustcmd*{\xGlsXtrSetField}[3]{%
696   \glstrsetfieldifexists{#1}{#2}%
697   {\protected@csxdef{glo@\glstoklabel{#1}@#2}{#3}}%
698 }

```

`eGlsXtrSetField`

```

699 \newrobustcmd*{\eGlsXtrSetField}[3]{%
700   \glstrsetfieldifexists{#1}{#2}%
701   {\protected@csedef{glo@\glstoklabel{#1}@#2}{#3}}%
702 }

```

`\glstrpageref` Like `\glssrefentry` but references the page number instead (if entry counting is on).

```

703 \ifglssentrycounter
704   \newcommand*{\glstrpageref}[1]{\pageref{glssentry-\glstoklabel{#1}}}
705 \else
706   \ifglssubentrycounter
707     \newcommand*{\glstrpageref}[1]{\pageref{glssentry-\glstoklabel{#1}}}
708   \else
709     \newcommand*{\glstrpageref}[1]{\gls{#1}}
710   \fi
711 \fi

```

lossarypreamble

```
712 \newcommand{\apptoglossarypreamble}[2][\glsdefaulttype]{%
713   \ifcsdef{glolist@#1}%
714   {%
715     \ifcsundef{@glossarypreamble@#1}%
716     {\csdef{@glossarypreamble@#1}{}}%
717     {}%
718     \csappto{@glossarypreamble@#1}{#2}%
719   }%
720   {%
721     \GlossariesExtraWarning{Glossary ‘#1’ is not defined}%
722   }%
723 }
```

lossarypreamble

```
724 \newcommand{\preglossarypreamble}[2][\glsdefaulttype]{%
725   \ifcsdef{glolist@#1}%
726   {%
727     \ifcsundef{@glossarypreamble@#1}%
728     {\csdef{@glossarypreamble@#1}{}}%
729     {}%
730     \cspreto{@glossarypreamble@#1}{#2}%
731   }%
732   {%
733     \GlossariesExtraWarning{Glossary ‘#1’ is not defined}%
734   }%
735 }
```

## 1.3 Modifications to Commands Provided by glossaries

Some of the commands provided by glossaries are modified to take into account new options or to change default behaviour.

Provide a starred version of `\longnewglossaryentry` that doesn't automatically insert `\leavevmode\unskip\nopostdesc` at the end of the description. The unstarred version is modified to use `\glxtrpostlongdescription` instead.

ewglossaryentry

```
736 \renewcommand*{\longnewglossaryentry}{%
737   \@ifstar{\glxtr@s@longnewglossaryentry}\glxtr@longnewglossaryentry
738 }
```

ewglossaryentry    Starred version.

```
739 \newcommand{\@glxtr@s@longnewglossaryentry}[3]{%
740   \glsdoifnoexists{#1}%
741   {%
742     \bgroup
743     \let\@org@newglossaryentryprehook\@newglossaryentryprehook
```

```

744 \long\def\@newglossaryentryprehook{%
745 \long\def\@glo@desc{#3}%
746 \@org@newglossaryentryprehook
747 }%
748 \renewcommand*{\gls@assign@desc}[1]{%
749 \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
750 \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@descplural}%
751 }
752 \gls@defglossaryentry{#1}{#2}%
753 \egroup
754 }%
755 }

```

`\newglossaryentry` Unstarred version.

```

756 \newcommand{\@glsxtr@longnewglossaryentry}[3]{%
757 \glsdoifnoexists{#1}%
758 {%
759 \bgroup
760 \let\@org@newglossaryentryprehook\@newglossaryentryprehook
761 \long\def\@newglossaryentryprehook{%
762 \long\def\@glo@desc{#3\glsxtrpostlongdescription}%
763 \@org@newglossaryentryprehook
764 }%
765 \renewcommand*{\gls@assign@desc}[1]{%
766 \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%

```

The following is different from the base glossaries.sty:

```

767 \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@descplural}%
768 }
769 \gls@defglossaryentry{#1}{#2}%
770 \egroup
771 }%
772 }

```

`\longdescription` Hook at the end of the description when using the unstarred `\longnewglossaryentry`.

```

773 \newcommand*{\glsxtrpostlongdescription}{\leavevmode\unskip\nopostdesc}

```

Provide a starred version of `\newignoredglossary` that doesn't add the glossary to the `nohyperlist` list.

`\ignoredglossary` Redefine to check for star.

```

774 \renewcommand{\newignoredglossary}{%
775 \@ifstar\glsxtr@s@newignoredglossary\glsxtr@org@newignoredglossary
776 }

```

`\ignoredglossary` The original definition is patched to check for existence.

```

777 \newcommand*{\glsxtr@org@newignoredglossary}[1]{%
778 \ifcsdef{glolist@#1}
779 {%

```

```

780   \glxtrundefaction{Glossary type ‘#1’ already exists}{}%
781 }%
782 {%
783   \ifdefempty\@ignored@glossaries
784   {%
785     \edef\@ignored@glossaries{#1}%
786   }%
787   {%
788     \eappto\@ignored@glossaries{,#1}%
789   }%
790   \csgdef{glolist@#1}{,}%
791   \ifcsundef{gls@#1@entryfmt}%
792   {%
793     \defglentryfmt[#1]{\glentryfmt}%
794   }%
795   {}%
796   \ifdefempty\@gls@nohyperlist
797   {%
798     \renewcommand*{\@gls@nohyperlist}{#1}%
799   }%
800   {%
801     \eappto\@gls@nohyperlist{,#1}%
802   }%
803 }%
804 }

```

ignoredglossary    Starred form.

```

805 \newcommand*{\glxtr@s@newignoredglossary}[1]{%
806   \ifcsdef{glolist@#1}
807   {%
808     \glxtrundefaction{Glossary type ‘#1’ already exists}{}%
809   }%
810   {%
811     \ifdefempty\@ignored@glossaries
812     {%
813       \edef\@ignored@glossaries{#1}%
814     }%
815     {%
816       \eappto\@ignored@glossaries{,#1}%
817     }%
818     \csgdef{glolist@#1}{,}%
819     \ifcsundef{gls@#1@entryfmt}%
820     {%
821       \defglentryfmt[#1]{\glentryfmt}%
822     }%
823     {}%
824   }%
825 }

```

`\glsettoctitle` Ignored glossaries don't have an associated title, so modify `\glsettoctitle` to check for it to prevent an undefined command written to the toc file.

```

826 \glusifusetranslator
827 {%
828   \renewcommand*{\glsettoctitle}[1]{%
829     \ifcsdef{gls@tr@set@#1@toctitle}%
830     {%
831       \csuse{gls@tr@set@#1@toctitle}%
832     }%
833     {%
834       \ifcsdef{@glotype@#1@title}%
835       {\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}%
836       {\def\glossarytoctitle{\glossarytitle}}%
837     }%
838   }%
839 }
840 {
841   \renewcommand*{\glsettoctitle}[1]{%
842     \ifcsdef{@glotype@#1@title}%
843     {\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}%
844     {\def\glossarytoctitle{\glossarytitle}}%
845   }
846 }

```

`ignoredglossary` As above but won't do anything if the glossary already exists.

```

847 \newcommand{\provideignoredglossary}{%
848   \@ifstar\glxtr@s@provideignoredglossary\glxtr@provideignoredglossary
849 }

```

`ignoredglossary` Unstarred version.

```

850 \newcommand*{\glxtr@provideignoredglossary}[1]{%
851   \ifcsdef{glolist@#1}
852   {}%
853   {%
854     \ifdefempty\@ignored@glossaries
855     {%
856       \edef\@ignored@glossaries{#1}%
857     }%
858     {%
859       \eappto\@ignored@glossaries{,#1}%
860     }%
861     \csgdef{glolist@#1}{,}%
862     \ifcsundef{gls@#1@entryfmt}%
863     {%
864       \defglentryfmt[#1]{\glentryfmt}%
865     }%
866     {}%
867     \ifdefempty\@gls@nohyperlist
868     {%

```

```

869     \renewcommand*{\@gls@nohyperlist}{#1}%
870   }%
871   {%
872     \eappto\@gls@nohyperlist{, #1}%
873   }%
874 }%
875 }

```

`ignoredglossary` Starred form.

```

876 \newcommand*{\glsxtr@s@provideignoredglossary}[1]{%
877   \ifcsdef{glolist@#1}
878   {%
879     {%
880       \ifdefempty\@ignored@glossaries
881       {%
882         \edef\@ignored@glossaries{#1}%
883       }%
884       {%
885         \eappto\@ignored@glossaries{, #1}%
886       }%
887       \csgdef{glolist@#1}{,}%
888       \ifcsundef{gls@#1@entryfmt}%
889       {%
890         \defglsentryfmt[#1]{\glsentryfmt}%
891       }%
892       {}%
893     }%
894 }

```

`rcopytoglossary` Adds an entry label to another glossary list. First argument is entry label. Second argument is glossary label.

```

895 \newcommand*{\glsxtrcopytoglossary}[2]{%
896   \glsdoifexists{#1}%
897   {%
898     \ifcsdef{glolist@#2}
899     {%
900       \cseappto{glolist@#2}{#1,}%
901     }%
902     {%
903       \glsxtrundefaction{Glossary type ‘#2’ doesn’t exist}{}%
904     }%
905   }%
906 }

```

### 1.3.1 Existence Checks

`\glsdoifexists` Modify `\glsdoifexists` to take account of the undefaction setting.

```

907 \renewcommand{\glsdoifexists}[2]{%
908   \ifglsentryexists{#1}{#2}%

```

909 {%

Define \glslabel in case it's needed after this command (for example in the post-link hook).

```
910 \edef\glslabel{\glsdetoklabel{#1}}%
911 \glstrundefaction{Glossary entry '\glslabel'
912 has not been defined}{You need to define a glossary entry before
913 you can reference it.}%
914 }%
915 }
```

glsdoifnoexists Modify \glsdoifnoexists to take account of the undefaction setting.

```
916 \renewcommand{\glsdoifnoexists}[2]{%
917 \ifglentryexists{#1}{%
918 \glstrundefaction{Glossary entry '\glsdetoklabel{#1}'
919 has already been defined}{}}{#2}%
920 }
```

glsdoifexistsordo Modify \glsdoifexistsordo to take account of the undefaction setting. This command was introduced in glossaries version 4.19, so check if it has been defined first.

```
921 \ifdef\glsdoifexistsordo
922 {%
923 \renewcommand{\glsdoifexistsordo}[3]{%
924 \ifglentryexists{#1}{#2}%
925 {%
926 \glstrundefaction{Glossary entry '\glsdetoklabel{#1}'
927 has not been defined}{You need to define a glossary entry
928 before you can use it.}%
929 #3%
930 }%
931 }%
932 }
933 {%
934 \glstr@warnonexistsordo\glsdoifexistsordo
935 \newcommand{\glsdoifexistsordo}[3]{%
936 \ifglentryexists{#1}{#2}%
937 {%
938 \glstrundefaction{Glossary entry '\glsdetoklabel{#1}'
939 has not been defined}{You need to define a glossary entry
940 before you can use it.}%
941 #3%
942 }%
943 }%
944 }
```

glsdoifglossarynoexistsordo Similarly for \doifglossarynoexistsordo.

```
945 \ifdef\doifglossarynoexistsordo
946 {%
947 \renewcommand{\doifglossarynoexistsordo}[3]{%
```



```

948 \ifglossaryexists{#1}%
949 {%
950 \glstrundefaction{Glossary type ‘#1’ already exists}{}%
951 #3%
952 }%
953 {#2}%
954 }%
955 }
956 {%
957 \glstr@warnonexistsordo\doifglossarynoexistsordo
958 \newcommand{\doifglossarynoexistsordo}[3]{%
959 \ifglossaryexists{#1}%
960 {%
961 \glstrundefaction{Glossary type ‘#1’ already exists}{}%
962 #3%
963 }%
964 {#2}%
965 }%
966 }
967

```

There are now three types of cross-references: the see key (as original), the alias key (from glossaries-extra v1.12) and the seealso key (from glossaries-extra v1.16). The original see key needs to have a corresponding field (which it doesn't with the base glossaries package).

`ryentryposthook` Hook into end of `\newglossaryentry` to add “see” value as a field.

```

968 \appto\@newglossaryentryposthook{%
969 \ifdefvoid\@glo@see
970 {\csxdef{glo@\@glo@label @see}{}}%
971 {%
972 \csxdef{glo@\@glo@label @see}{\@glo@see}%
973 \if@glstr@autoseeindex
974 \@glstr@autoindexcrossrefs
975 \fi
976 }%
977 }
978 \appto\@gls@keymap{,{see}{see}}

```

`\glstrusesee` Apply `\glsseeformat` to the see key if not empty.

```

979 \newcommand*{\glstrusesee}[1]{%
980 \glsdoifexists{#1}%
981 {%
982 \letcs{\@glo@see}{glo\glsdetoklabel{#1}@see}%
983 \ifdefempty\@glo@see
984 {}%
985 {%
986 \expandafter\glstr@usesee\@glo@see\@end@glstr@usesee
987 }%
988 }%

```

```

989 }

\glxtr@usesee
990 \newcommand*{\glxtr@usesee}[1][\seename]{%
991   \@glxtr@usesee[#1]%
992 }

\@glxtr@usesee
993 \def\@glxtr@usesee[#1]#2\@end@glxtr@usesee{%
994   \glxtruseseeformat{#1}{#2}%
995 }

xtruseseeformat  The format used by \glxtrusesee. The first argument is the tag (such as \seename). The
                  second argument is the comma-separated list of cross-referenced labels.
996 \newcommand*{\glxtruseseeformat}[2]{%
997   \glssseeformat[#1]{#2}{}%
998 }

lsxtruseseealso  Apply \glssseeformat to the seealso key if not empty. There's no optional tag to worry about
                  here.
999 \newcommand*{\glxtruseseealso}[1]{%
1000   \glsdoidexists{#1}%
1001   {%
1002     \letcs{\@glo@see}{glo\glsdetoklabel{#1}@seealso}%
1003     \ifdefempty\@glo@see
1004       {}%
1005       {%
1006         \expandafter\glxtruseseealsoformat\expandafter{\@glo@see}%
1007       }%
1008     }%
1009 }

sesealsoformat  The format used by \glxtruseseealso. The argument is the comma-separated list of
                  cross-referenced labels.
1010 \newcommand*{\glxtruseseealsoformat}[1]{%
1011   \glssseeformat[\seealso]{#1}{}%
1012 }

\glxtrseelist  Fully expands argument before passing to \glssseelist. (The argument to \glssseelist
                  must be a comma-separated list of entry labels.)
1013 \newrobustcmd{\glxtrseelist}[1]{%
1014   \edef\@glo@tmp{\noexpand\glssseelist{#1}}\@glo@tmp
1015 }

\seealso  In case this command hasn't been defined. (Should be provided by language packages.)
1016 \providecommand{\seealso}{}

```

`xtrindexseealso` If `\@xdycrossrefhook` is defined, provide a `seealso` crossref class. Otherwise this just does `\glssee` with `\seealso` as the tag. The hook is only defined if both `xindy` and `glossaries v4.30+` are being used.

```
1017 \ifdef\@xdycrossrefhook
1018 {
```

Add the cross-reference class definition to the hook.

```
1019 \appto\@xdycrossrefhook{%
1020 \write\glswrite{(define-crossref-class \string"seealso\string"
1021 :unverified )}%
1022 \write\glswrite{(markup-crossref-list
1023 :class \string"seealso\string"^^J\space\space\space
1024 :open \string"\string\glsxtruseealsoformat\glsopenbrace\string"
1025 :close \string"\glsclosebrace\string")}%
1026 }
```

Append to class list.

```
1027 \appto\@xdylocationclassorder{\space\string"seealso\string"}
```

This essentially works like `\@do@seeglossary` but uses the `seealso` class.

```
1028 \newrobustcmd*{\glsxtrindexseealso}[2]{%
1029 \def\@gls@xref{#2}%
1030 \@onelevel@sanitize\@gls@xref
1031 \@gls@checkmkidxchars\@gls@xref
1032 \gls@glossary{\csname glo@#1@type\endcsname}{%
1033 (indexentry
1034 :tkey (\csname glo@#1@index\endcsname)
1035 :xref (\string"\@gls@xref\string")
1036 :attr \string"seealso\string"
1037 )
1038 }%
1039 }
1040 }
1041 {
```

`xindy` not in use or `glossaries` version too old to support this.

```
1042 \newrobustcmd*{\glsxtrindexseealso}{\glssee[\seealso]}
1043 }
```

The alias key should be set to the label of the synonymous entry. The `seealso` key essentially behaves like `see=[\seealso]{\langle x r - l i s t \rangle}`. Neither of these new keys has the optional tag part allowed with `see`.

If `\gls@set@xr@key` has been defined (`glossaries v4.30`), use that, otherwise just use `\glsaddstoragekey`.

```
1044 \ifdef\gls@set@xr@key
1045 {
```

We have at least `glossaries v4.30`. This means the new keys can be governed by the same settings as the `see` key.

```

1046 \define@key{glossentry}{alias}{%
1047   \gls@set@xr@key{alias}{\@glo@alias}{#1}%
1048 }
1049 \define@key{glossentry}{seealso}{%
1050   \gls@set@xr@key{seealso}{\@glo@seealso}{#1}%
1051 }

```

Add to the key mappings.

```

1052 \appto\@gls@keymap{,{alias}{alias},{seealso}{seealso}}

```

Set the default value.

```

1053 \appto\@newglossaryentryprehook{\def\@glo@alias{}\def\@glo@seealso{}}%

```

Assign the field values.

```

1054 \appto\@newglossaryentryposthook{%
1055   \ifdefvoid\@glo@seealso
1056     {\csxdef{glo@\@glo@label @seealso}{}}%
1057   {%
1058     \csxdef{glo@\@glo@label @seealso}{\@glo@seealso}%
1059     \if@glxtr@autoseeindex
1060       \@glxtr@autoindexcrossrefs
1061     \fi
1062   }%

```

The alias field doesn't trigger the automatic cross-reference indexing performed at the end of the document.

```

1063   \ifdefvoid\@glo@alias
1064     {\csxdef{glo@\@glo@label @alias}{}}%
1065   {%
1066     \csxdef{glo@\@glo@label @alias}{\@glo@alias}%
1067   }%
1068 }

```

Provide user-level commands to access the values.

`\glxtralias`

```

1069 \newcommand*{\glxtralias}[1]{\@gls@entry@field{#1}{alias}}

```

`trseealsolabels`

```

1070 \newcommand*{\glxtrseealsolabels}[1]{\@gls@entry@field{#1}{seealso}}

```

Add to the `\@glo@autosee` hook.

```

1071 \appto\@glo@autoseehook{%
1072   \ifdefvoid\@glo@alias
1073     {%
1074       \ifdefvoid\@glo@seealso
1075         {}%
1076       {%
1077         \edef\@do@glsssee{\noexpand\glxtrindexseealso
1078           {\@glo@label}{\@glo@seealso}}%
1079         \@do@glsssee

```

```

1080     }%
1081     }%
1082     {%

```

Add cross-reference if see key hasn't been used.

```

1083     \ifdefvoid\@glo@see
1084     {%
1085         \edef\@do@glsssee{\noexpand\glsssee{\@glo@label}{\@glo@alias}}%
1086         \@do@glsssee
1087     }%
1088     {}%
1089     }%
1090 }%
1091 }
1092 {

```

We have an older version of glossaries, so just use \glsaddstoragekey.

\glsxtralias

```

1093 \glsaddstoragekey*{alias}{}{\glsxtralias}

```

trseealsolabels

```

1094 \glsaddstoragekey*{seealso}{}{\glsxtrseealsolabels}

```

If \gls@set@xr@key isn't defined, then \@glo@autosee won't be either, so use the post entry definition hook.

ryentryposthook Append to the hook to check for the alias and seealso keys.

```

1095 \appto\@newglossaryentryposthook{%
1096     \ifcsvoid{glo@\@glo@label @alias}%
1097     {%
1098         \ifcsvoid{glo@\@glo@label @seealso}%
1099         {}%
1100         {%
1101             \edef\@do@glsssee{\noexpand\glsxtrindexseealso
1102                 {\@glo@label}{\csuse{glo@\@glo@label @seealso}}}%
1103             \@do@glsssee
1104         }%
1105     }%
1106     {%

```

Add cross-reference if see key hasn't been used.

```

1107     \ifdefvoid\@glo@see
1108     {%
1109         \edef\@do@glsssee{\noexpand\glsssee
1110             {\@glo@label}{\csuse{glo@\@glo@label @alias}}}%
1111         \@do@glsssee
1112     }%
1113     {}%
1114     }%
1115 }

```

1116 }

Add all unused cross-references at the end of the document.

1117 \AtEndDocument{\if@glxtrindexcrossrefs\glxtraddallcrossrefs\fi}

addallcrossrefs Iterate through all used entries and if they have a cross-reference, make sure the cross-reference has been added.

```
1118 \newcommand*\glxtraddallcrossrefs{%
1119   \forallglossaries{\@glo@type}%
1120   {%
1121     \for@gl@entries[\@glo@type]{\@glo@label}%
1122     {%
1123       \if@gl@sused{\@glo@label}%
1124       {\expandafter\glxtr@addunusedxrefs\expandafter{\@glo@label}}{}%
1125     }%
1126   }%
1127 }
```

@addunusedxrefs If the given entry has a see or seealso field add all unused cross-references. (The alias field isn't checked.)

```
1128 \newcommand*\@glxtr@addunusedxrefs[1]{%
1129   \letcs{\@glo@see}{glo\gl@detoklabel{#1}@see}%
1130   \ifdefvoid\@glo@see
1131   {}%
1132   {%
1133     \expandafter\glxtr@addunused\@glo@see\@end@glxtr@addunused
1134   }%
1135   \letcs{\@glo@see}{glo\gl@detoklabel{#1}@seealso}%
1136   \ifdefvoid\@glo@see
1137   {}%
1138   {%
1139     \expandafter\glxtr@addunused\@glo@see\@end@glxtr@addunused
1140   }%
1141 }
```

lsxtr@addunused Adds all the entries if they haven't been used.

```
1142 \newcommand*\glxtr@addunused[1][ ]{%
1143   \@glxtr@addunused
1144 }
```

lsxtr@addunused Adds all the entries if they haven't been used.

```
1145 \def\@glxtr@addunused#1\@end@glxtr@addunused{%
1146   \@for\@glxtr@label:=#1\do
1147   {%
1148     \if@gl@sused{\@glxtr@label}{}%
1149     {%
1150       \gl@sadd[format=glxtrunusedformat]{\@glxtr@label}%
1151       \gl@unset{\@glxtr@label}%
1152       \expandafter\glxtr@addunusedxrefs\expandafter{\@glxtr@label}%

```

```

1153 }%
1154 }%
1155 }

```

extrunusedformat

```

1156 \newcommand*{\glxtrunusedformat}[1]{\unskip}

```

### 1.3.2 Document Definitions

noidxglossaries Modify \makenoidxglossaries so that it automatically switches off (unless the restricted setting is on) and disables the docdef key.

```

1157 \let\glxtr@orgmakenoidxglossaries\makenoidxglossaries
1158 \renewcommand{\makenoidxglossaries}{%
1159   \glxtr@orgmakenoidxglossaries
1160   \if@glxtrdocdefrestricted

```

If restricted document definitions allowed, adjust \@gls@reference so that it doesn't test for existence.

```

1161   \renewcommand*{\@gls@reference}[3]{%
1162     \ifcsundef{@glsref@##1}{\csgdef{@glsref@##1}{}}{}%
1163     \ifinlistcs{##2}{@glsref@##1}%
1164     {}%
1165     {\listcsgadd{@glsref@##1}{##2}}%
1166     \ifcsundef{glo@glsdetoklabel{##2}@loclist}%
1167     {\csgdef{glo@glsdetoklabel{##2}@loclist}{}}{}%
1168     {}%
1169     \listcsgadd{glo@glsdetoklabel{##2}@loclist}{##3}%
1170   }%
1171   \else

```

Disable document definitions.

```

1172   \@glxtrdocdeffalse
1173   \fi
1174   \disable@keys{glossaries-extra}{docdef}%
1175 }

```

ewglossaryentry Modify \gls@defdocnewglossaryentry so that it checks the docdef value.

```

1176 \renewcommand*{\gls@defdocnewglossaryentry}{%
1177   \ifcase\@glxtrdocdefval
1178     docdef=false:
1179     \renewcommand*{\newglossaryentry}[2]{%
1180       \PackageError{glossaries-extra}{Glossary entries must
1181       be \MessageBreak defined in the preamble with \MessageBreak
1182       package option 'docdef=false'\MessageBreak(consider using
1183       'docdef=restricted')}{Move your glossary definitions to
1184       the preamble. You can also put them in a \MessageBreak separate file
1185       and load them with \string\loadglsentries.}%
1186     }%
1187   \or

```

docdef=true Since the see value is now saved in a field, it can be used by entries that have been defined in the document.

```
1187 \let\gls@checkseeallowed\relax
1188 \let\newglossaryentry\new@glossaryentry
1189 \or
```

Restricted mode just needs to allow the see value.

```
1190 \let\gls@checkseeallowed\relax
1191 \fi
1192 }%
```

Permit a special form of document definition, but only allow it if the glossaries come at the end of the document. These commands behave a little like a combination of `\newterm` and `\gls`. This must be explicitly enabled with the following.

`\rEnableOnTheFly`

```
1193 \newcommand*{\GlsXtrEnableOnTheFly}{%
1194 \ifstar\@sGlsXtrEnableOnTheFly\@GlsXtrEnableOnTheFly
1195 }
```

`\rEnableOnTheFly`

The starred version attempts to allow UTF8 characters in the label, but this may break! (Formatting commands mustn't be used in the label, but the label may be a command whose replacement text is the actual label. This doesn't take into account a command that's defined in terms of another command that may eventually expand to the label text.)

```
1196 \newcommand*{\@sGlsXtrEnableOnTheFly}{%
1197 \renewcommand*{\glsdetoklabel}[1]{%
1198 \expandafter\@glsxtr@ifcsstart\string##1 \@glsxtr@end@
1199 {%
1200 \expandafter\detokenize\expandafter{##1}%
1201 }%
1202 {\detokenize{##1}}%
1203 }%
1204 \@GlsXtrEnableOnTheFly
1205 }
1206 \def\@glsxtr@ifcsstart#1#2\@glsxtr@end@#3#4{%
1207 \expandafter\if\glsbackslash#1%
1208 #3%
1209 \else
1210 #4%
1211 \fi
1212 }
```

`\sXtrstarflywarn`

```
1213 \newcommand*{\glsxtrstarflywarn}{%
1214 \GlossariesExtraWarning{Experimental starred version of
1215 \string\GlsXtrEnableOnTheFly\space in use (please ensure you have
1216 read the warnings in the glossaries-extra user manual)}}%
1217 }
```



rEnableOnTheFly

```
1218 \newcommand*{\@GlsXtrEnableOnTheFly}{%
```

Don't redefine \glsdetoklabel if LuaTeX or XeTeX is being used, since it's mainly to allow accented characters in the label.

These definitions are all assigned the category given by:

\glsxtrcat

```
1219 \newcommand*{\glsxtrcat}{general}
```

\glsxtr

```
1220 \newcommand*{\glsxtr}[1] [] {%
1221 \def\glsxtr@keylist{##1}%
1222 \@glsxtr
1223 }
```

\@glsxtr

```
1224 \newcommand*{\@glsxtr}[2] [] {%
1225 \ifglsentryexists{##2}%
1226 {%
1227 \ifblank{##1}{-}{\GlsXtrWarning{##1}{##2}}%
1228 }%
1229 {%
1230 \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
1231 description={\nopostdesc},##1}%
1232 }%
1233 \expandafter\gls\expandafter[\glsxtr@keylist]{##2}%
1234 }
```

\Glsxtr

```
1235 \newcommand*{\Glsxtr}[1] [] {%
1236 \def\glsxtr@keylist{##1}%
1237 \@Glsxtr
1238 }
```

\@Glsxtr

```
1239 \newcommand*{\@Glsxtr}[2] [] {%
1240 \ifglsentryexists{##2}%
1241 {%
1242 \ifblank{##1}{-}{\GlsXtrWarning{##1}{##2}}%
1243 }%
1244 {%
1245 \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
1246 description={\nopostdesc},##1}%
1247 }%
1248 \expandafter\Gls\expandafter[\glsxtr@keylist]{##2}%
1249 }
```

\glstrpl

```
1250 \newcommand*\glstrpl[1][]{%
1251 \def\glstr@keylist{##1}%
1252 \@glstrpl
1253 }
```

\@glstrpl

```
1254 \newcommand*\@glstrpl[2][]{%
1255 \ifglstryexists{##2}%
1256 {%
1257 \ifblank{##1}{\GlsXtrWarning{##1}{##2}}%
1258 }%
1259 {%
1260 \gls@defglossaryentry{##2}{name={##2},category=\glstrcat,
1261 description={\nopostdesc},##1}%
1262 }%
1263 \expandafter\glsp\expandafter[\glstr@keylist]{##2}%
1264 }
```

\Glsxtrpl

```
1265 \newcommand*\Glsxtrpl[1][]{%
1266 \def\glstr@keylist{##1}%
1267 \@Glsxtrpl
1268 }
```

\@Glsxtrpl

```
1269 \newcommand*\@Glsxtrpl[2][]{%
1270 \ifglstryexists{##2}
1271 {%
1272 \ifblank{##1}{\GlsXtrWarning{##1}{##2}}%
1273 }%
1274 {%
1275 \gls@defglossaryentry{##2}{name={##2},category=\glstrcat,
1276 description={\nopostdesc},##1}%
1277 }%
1278 \expandafter\Glsp\expandafter[\glstr@keylist]{##2}%
1279 }
```

\GlsXtrWarning

```
1280 \newcommand*\GlsXtrWarning[2]{%
1281 \def\@glstr@optlist{##1}%
1282 \@onelevel@sanitize\@glstr@optlist
1283 \GlossariesExtraWarning{The options ‘\@glstr@optlist’ have
1284 been ignored for entry ‘##2’ as it has already been defined}%
1285 }
```

Disable commands after the glossary:

```
1286 \renewcommand\@printglossary[2]{%
```

```

1287 \def\@glxtr@printglossopts{##1}%
1288 \@glxtr@orgprintglossary{##1}{##2}%
1289 \def\@glxtr{\@glxtr@disabledflycommand\glxtr}%
1290 \def\@glxtrpl{\@glxtr@disabledflycommand\glxtrpl}%
1291 \def\@Glsxtr{\@glxtr@disabledflycommand\Glsxtr}%
1292 \def\@Glsxtrpl{\@glxtr@disabledflycommand\Glsxtrpl}%
1293 }

```

abledflycommand

```

1294 \newcommand*{\@glxtr@disabledflycommand}[1]{%
1295 \PackageError{glossaries-extra}%
1296 {\string##1\space can't be used after any of the \MessageBreak
1297 glossaries have been displayed}%
1298 {The on-the-fly commands enabled by
1299 \string\GlsXtrEnableOnTheFly\space may only be used \MessageBreak
1300 before the glossaries. If you want to use any entries \MessageBreak
1301 after any of the glossaries, you must use the standard \MessageBreak
1302 method of first defining the entry and then using the \MessageBreak
1303 entry with commands like \string\glx}%
1304 \@glxtr@disabledflycommand
1305 }%
1306 \newcommand*{\@glxtr@disabledflycommand}[2][{}]{##2}

```

End of \GlsXtrEnableOnTheFly. Disable since it can only be used once.

```

1307 \let\GlsXtrEnableOnTheFly\relax
1308 }
1309 \@onlypreamble\GlsXtrEnableOnTheFly

```

### 1.3.3 Existing Glossary Style Modifications

Modify \setglossarystyle to keep track of the current style. This allows the \glossaries-extra-stylemods package to reset the current style after the required modifications have been made.

r@current@style Initialise the current style to the default style.

```

1310 \newcommand*{\@glxtr@current@style}{\@glossary@default@style}

```

Modify \setglossarystyle to set \@glxtr@current@style.

etglossarystyle

```

1311 \renewcommand*{\setglossarystyle}[1]{%
1312 \ifcsundef{@glstyle@#1}%
1313 {%
1314 \PackageError{glossaries-extra}{Glossary style ‘#1’ undefined}{}%
1315 }%
1316 {%
1317 \csname @glstyle@#1\endcsname

```

Only set the current style if it exists.

```

1318 \protected@edef\@glxtr@current@style{#1}%
1319 }%

```

```

1320 \ifx\@glossary@default@style\relax
1321   \protected@edef\@glossary@default@style{#1}%
1322 \fi
1323 }

```

In case we have an old version of glossaries:

```

1324 \ifdef\@glossary@default@style
1325 {}
1326 {%
1327   \let\@glossary@default@style\relax
1328 }

```

`listdottedwidth` If `\glslistdottedwidth` has been defined and is currently equal to `.5\hsize` then make the modification suggested in [bug report #92](#)

```

1329 \ifdef\glslistdottedwidth
1330 {%
1331   \ifdim\glslistdottedwidth=.5\hsize
1332     \setlength{\glslistdottedwidth}{-\dimexpr\maxdimen-1sp\relax}
1333     \AtBeginDocument{%
1334       \ifdim\glslistdottedwidth=-\dimexpr\maxdimen-1sp\relax
1335         \setlength{\glslistdottedwidth}{.5\columnwidth}%
1336       \fi
1337     }%
1338   \fi
1339 }
1340 {}%

```

Similarly for `\glsdescwidth`:

`\glsdescwidth`

```

1341 \ifdef\glsdescwidth
1342 {%
1343   \ifdim\glsdescwidth=.6\hsize
1344     \setlength{\glsdescwidth}{-\dimexpr\maxdimen-1sp\relax}
1345     \AtBeginDocument{%
1346       \ifdim\glsdescwidth=-\dimexpr\maxdimen-1sp\relax
1347         \setlength{\glsdescwidth}{.6\columnwidth}%
1348       \fi
1349     }%
1350   \fi
1351 }
1352 {}%

```

and for `\glspagelistwidth`:

`lspagelistwidth`

```

1353 \ifdef\glspagelistwidth
1354 {%
1355   \ifdim\glspagelistwidth=.1\hsize
1356     \setlength{\glspagelistwidth}{-\dimexpr\maxdimen-1sp\relax}

```

```

1357 \AtBeginDocument{%
1358 \ifdim\glspagelistwidth=-\dimexpr\maxdimen-1sp\relax
1359 \setlength{\glspagelistwidth}{.1\columnwidth}%
1360 \fi
1361 }%
1362 \fi
1363 }
1364 {}%

```

aryentrynumbers Has the nonumberlist option been used?

```

1365 \def\org@glossaryentrynumbers#1{#1\gls@save@numberlist{#1}}%
1366 \ifx\org@glossaryentrynumbers\glossaryentrynumbers
1367 \glsnonumberlistfalse
1368 \renewcommand*{\glossaryentrynumbers}[1]{%
1369 \ifglsentryexists{\glscurrententrylabel}%
1370 {%
1371 \@glsxtrpreloctag
1372 \GlsXtrFormatLocationList{#1}%
1373 \@glsxtrpostloctag
1374 \gls@save@numberlist{#1}%
1375 }{}%
1376 }%
1377 \else
1378 \glsnonumberlisttrue
1379 \renewcommand*{\glossaryentrynumbers}[1]{%
1380 \ifglsentryexists{\glscurrententrylabel}%
1381 {%
1382 \gls@save@numberlist{#1}%
1383 }{}%
1384 }%
1385 \fi

```

matLocationList Provide an easy interface to change the format of the location list without removing the save number list stuff.

```

1386 \newcommand*{\GlsXtrFormatLocationList}[1]{#1}

```

Sometimes users want to prefix the location list with “page”/“pages”. The simplest way to determine if the location list consists of a single location is to check for instances of `\delimN` or `\delimR`, but this isn’t so easy to do as they might be embedded inside the argument of formatting commands. With a bit of trickery we can find out by adjusting `\delimN` and `\delimR` to set a flag and then save information to the auxiliary file for the next run.

ePreLocationTag

```

1387 \newcommand*{\GlsXtrEnablePreLocationTag}[2]{%
1388 \let\@glsxtrpreloctag\@glsxtrpreloctag
1389 \let\@glsxtrpostloctag\@glsxtrpostloctag
1390 \renewcommand*{\@glsxtr@pagetag}{#1}%
1391 \renewcommand*{\@glsxtr@pagetag}{#2}%
1392 \renewcommand*{\@glsxtr@savepreloctag}[2]{%

```

```

1393 \csgdef{@glxtr@preloctag@##1}{##2}%
1394 }%
1395 \renewcommand*{\@glxtr@doloctag}{%
1396 \ifcsundef{@glxtr@preloctag@glscurrententrylabel}%
1397 {%
1398 \GlossariesWarning{Missing pre-location tag for ‘\glscurrententrylabel’.
1399 Rerun required}%
1400 }%
1401 {%
1402 \csuse{@glxtr@preloctag@glscurrententrylabel}%
1403 }%
1404 }%
1405 }
1406 \@onlypreamble\GlsXtrEnablePreLocationTag

```

glxtrpreloctag

```

1407 \newcommand*{\@glxtr@preloctag}{%
1408 \let\@glxtr@org@delimN\delimN
1409 \let\@glxtr@org@delimR\delimR
1410 \let\@glxtr@org@glsignore\glsignore
1411 \gdef is required as the delimiters may occur inside a scope.
1412 \gdef\@glxtr@thisloctag{\@glxtr@pagetag}%
1413 \renewcommand*{\delimN}{%
1414 \gdef\@glxtr@thisloctag{\@glxtr@pagetag}%
1415 \@glxtr@org@delimN}%
1416 \renewcommand*{\delimR}{%
1417 \gdef\@glxtr@thisloctag{\@glxtr@pagetag}%
1418 \@glxtr@org@delimR}%
1419 \renewcommand*{\glsignore}[1]{%
1420 \gdef\@glxtr@thisloctag{\relax}%
1421 \@glxtr@org@glsignore{##1}}%
1422 }

```

glxtrpreloctag

```

1423 \newcommand*{\@glxtr@preloctag}{}

```

@glxtr@pagetag

```

1424 \newcommand*{\@glxtr@pagetag}{}%

```

glxtr@pagetag

```

1425 \newcommand*{\@glxtr@pagetag}{}%

```

lsxtrpostloctag

```

1426 \newcommand*{\@glxtr@postloctag}{%
1427 \let\delimN\@glxtr@org@delimN
1428 \let\delimR\@glxtr@org@delimR
1429 \let\glsignore\@glxtr@org@glsignore

```

```

1430 \protected@write\@auxout{}\%
1431 {\string\@glxtr@savepreloctag{\glscurrententrylabel}\@glxtr@thisloctag}}%
1432 }

```

lsxtrpostloctag

```

1433 \newcommand*\@glxtrpostloctag{}\%

```

lsxtr@preloctag

```

1434 \newcommand*\@glxtr@savepreloctag}[2]\%
1435 \protected@write\@auxout{}\%
1436 \string\providecommand\string\@glxtr@savepreloctag[2]\%

```

glxtr@doloctag

```

1437 \newcommand*\@glxtr@doloctag{}\%

```

ss@nonumberlist Modify the nonumberlist key to use \GlsXtrFormatLocationList (and also save the number list):

```

1438 \renewcommand*\KV@printgloss@nonumberlist}[1]\%
1439 \XKV@plfalse
1440 \XKV@sttrue
1441 \XKV@checkchoice[\XKV@resa]{#1}{true,false}%
1442 {\%
1443 \csname glsnonumberlist\XKV@resa\endcsname
1444 \ifglsnonumberlist
1445 \def\glossaryentrynumbers##1{\gls@save@numberlist{##1}}%
1446 \else
1447 \def\glossaryentrynumbers##1{\%
1448 \@glxtrpreloctag
1449 \GlsXtrFormatLocationList{##1}%
1450 \@glxtrpostloctag
1451 \gls@save@numberlist{##1}}%
1452 \fi
1453 }%
1454 }

```

### 1.3.4 Entry Formatting, Hyperlinks and Indexing

**\glsentryfmt** Change default entry format. Use the generic format for regular terms (that is, entries that have a category with the regular attribute set) or non-regular terms without a short value and use the abbreviation format for non-regular terms that have a short value. If further attributes need to be checked, then **\glsentryfmt** will need redefining as appropriate (or use **\defglsentryfmt**). The abbreviation format is set here for entries that have a short form, even if they are regular entries to ensure the abbreviation fonts are correct.

```

1455 \renewcommand*\glsentryfmt{\%
1456 \ifglshasshort{\glslabel}\glssetabbrvfmt{\glscategory{\glslabel}}}%
1457 \glsifregular{\glslabel}%
1458 {\glxtrregularfont{\glsentryfmt}}%

```

```

1459 {%
1460   \ifglshasshort{\glslabel}%
1461   {\glsxtrgenabbrvfmt}%
1462   {\glsxtrregularfont{\glsgenentryfmt}}}%
1463 }%
1464 }

```

`sxtrregularfont` Font used for regular entries.

```

1465 \newcommand*{\glsxtrregularfont}[1]{#1}

```

Commands like `\glusifplural` are only used by the `\gls`-like commands in the glossaries package, but it might be useful for the postlink hook to know if the user has used, say, `\glsfirst` or `\glsplural`. This can provide better consistency with the formatting of the `\gls`-like commands, even though they don't use `\glsentryfmt`.

`\@gls@field@link` Redefine `\@gls@field@link` so that commands like `\glsfirst` can setup `\glxtrifwasfirstuse` etc to allow the postlink hook to work better. This now has an optional argument that sets up the defaults.

```

1466 \renewcommand{\@gls@field@link}[4][]{%

```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists.

```

1467   \@glxtr@record{#2}{#3}{\glslink}%
1468   \glsoifexists{#3}%
1469   {%

```

Save and restore the hyper setting (`\@gls@link` also does this, but that's too late if the optional argument of `\@gls@field@link` modifies it).

```

1470   \let\glxtrorg@ifKV@glslink@hyper\ifKV@glslink@hyper
1471   \let\do@glslink@checkfirsthyper\@glslink@nocheckfirsthyper
1472   \def\glscustomtext{#4}%
1473   \@glxtr@field@linkdefs
1474   #1%
1475   \@glslink[#2]{#3}{#4}%
1476   \let\ifKV@glslink@hyper\glxtrorg@ifKV@glslink@hyper
1477   }%
1478   \glspostlinkhook
1479 }

```

The commands `\gls`, `\Gls` etc don't use `\@gls@field@link`, so they need modifying as well to use `\@glxtr@record`.

`\@gls@` Save the original definition and redefine.

```

1480 \let\@glxtr@org@gls@\@gls@
1481 \def\@gls@#1#2{%
1482   \@glxtr@record{#1}{#2}{\glslink}%
1483   \@glxtr@org@gls@{#1}{#2}%
1484 }%

```



`\@glsp1@` Save the original definition and redefine.

```
1485 \let\@glxtr@org@glsp1@\@glsp1@
1486 \def\@glsp1@#1#2{%
1487   \@glxtr@record{#1}{#2}{glslink}%
1488   \@glxtr@org@glsp1@{#1}{#2}%
1489 }%
```

`\@Gls@` Save the original definition and redefine.

```
1490 \let\@glxtr@org@Gls@\@Gls@
1491 \def\@Gls@#1#2{%
1492   \@glxtr@record{#1}{#2}{glslink}%
1493   \@glxtr@org@Gls@{#1}{#2}%
1494 }%
```

`\@Glspl@` Save the original definition and redefine.

```
1495 \let\@glxtr@org@Glspl@\@Glspl@
1496 \def\@Glspl@#1#2{%
1497   \@glxtr@record{#1}{#2}{glslink}%
1498   \@glxtr@org@Glspl@{#1}{#2}%
1499 }%
```

`\@GLS@` Save the original definition and redefine.

```
1500 \let\@glxtr@org@GLS@\@GLS@
1501 \def\@GLS@#1#2{%
1502   \@glxtr@record{#1}{#2}{glslink}%
1503   \@glxtr@org@GLS@{#1}{#2}%
1504 }%
```

`\@GLSpl@` Save the original definition and redefine.

```
1505 \let\@glxtr@org@GLSpl@\@GLSpl@
1506 \def\@GLSpl@#1#2{%
1507   \@glxtr@record{#1}{#2}{glslink}%
1508   \@glxtr@org@GLSpl@{#1}{#2}%
1509 }%
```

`\@glsdisp` Save the original definition and redefine. Can't save and restore `\@glsdisp` since it has an optional argument.

```
1510 \renewcommand*{\@glsdisp}[3][{}]{%
1511   \@glxtr@record{#1}{#2}{glslink}%
1512   \glsdoifexists{#2}{%
1513     \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
1514     \let\glsifplural\@secondoftwo
1515     \let\glscapscase\@firstofthree
1516     \def\glscustomtext{#3}%
1517     \def\glsinsert{}%
1518     \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
1519     \@gl@link{#1}{#2}{\@glo@text}%
1520     \ifKV@glslink@local
```

```

1521     \glslocalunset{#2}%
1522   \else
1523     \glsunset{#2}%
1524   \fi
1525 }%
1526 \glspostlinkhook
1527 }

```

`\@gls@link@` Redefine to include `\@glstr@record`

```

1528 \renewcommand*{\@gls@link}[3][\%
1529   \@glstr@record{#1}{#2}{glslink}%
1530   \glsdoifexistsordo{#2}%
1531   {%
1532     \let\do@gls@link@checkfirsthyper\relax
1533     \@gls@link[#1]{#2}{#3}%
1534   }%
1535   {%
1536     \glstextformat{#3}%
1537   }%
1538   \glspostlinkhook
1539 }

```

`sxtrinitwrgloss` Set the default if the wrgloss is omitted.

```

1540 \newcommand*{\glstrinitwrgloss}{%
1541   \glsifattribute{\glslabel}{wrgloss}{after}%
1542   {%
1543     \glstrinitwrglossbeforefalse
1544   }%
1545   {%
1546     \glstrinitwrglossbeforetrue
1547   }%
1548 }

```

`trwrglossbefore` Conditional to determine if the indexing should be done before the link text.

```

1549 \newif\ifglstrinitwrglossbefore
1550 \glstrinitwrglossbeforetrue

```

Define a wrgloss key to determine whether to write the glossary information before or after the link text.

```

1551 \define@choicekey{glslink}{wrgloss}[\val\nr]{before,after}%
1552 {%
1553   \ifcase\nr\relax
1554     \glstrinitwrglossbeforetrue
1555   \or
1556     \glstrinitwrglossbeforefalse
1557   \fi
1558 }

```

```

1559 \define@key{glslink}{thevalue}{\def\@glstr@thevalue{#1}}

```

```
1560 \define@key{glslink}{theHvalue}{\def\@glxtr@theHvalue{#1}}
```

`\@gls@link` Redefine to allow the indexing to be placed after the link text. By default this is done before the link text to prevent problems that can occur from the `whatsit`, but there may be times when the user would like the indexing done afterwards even though it causes a `whatsit`.

```
1561 \def\@gls@link[#1]#2#3{%
1562   \leavevmode
1563   \edef\glslabel{\glsdetoklabel{#2}}%
1564   \def\@gls@link@opts{#1}%
1565   \let\@gls@link@label\glslabel
1566   \let\@glsnumberformat\@glxtr@defaultnumberformat
1567   \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
1568   \edef\gls@type{\csname glo@\glslabel @type\endcsname}%
1569   \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
```

Initialise `thevalue` and `theHvalue` (v1.19).

```
1570   \def\@glxtr@thevalue{%
1571   \def\@glxtr@theHvalue{\@glxtr@thevalue}%
```

Initialise when indexing should occur (new to v1.14).

```
1572   \glxtrinitwrgloss
```

As the original definition. Note that the default link options may override `\glxtrinitwrgloss`.

```
1573   \@gls@setdefault@glslink@opts
1574   \do@gl:disablehyperinlist
1575   \do@gls@link@checkfirsthyper
1576   \setkeys{glslink}{#1}%
1577   \glslinkpostsetkeys
```

Check `thevalue` and `theHvalue` before saving (v1.19).

```
1578   \ifdefempty{\@glxtr@thevalue}%
1579   {%
1580     \@gls@saveentrycounter
1581   }%
1582   {%
1583     \let\theglentrycounter\@glxtr@thevalue
1584     \def\theHglentrycounter{\@glxtr@theHvalue}%
1585   }%
1586   \@gls@setsort{\glslabel}%
```

Do write if it should occur before the link text:

```
1587   \ifglxtrinitwrglossbefore
1588     \do@wrglossary{#2}%
1589   \fi
```

Do the link text:

```
1590   \ifKV@glslink@hyper
1591     \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
1592   \else
1593     \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
1594   \fi
```

Do write if it should occur after the link text:

```
1595 \ifglstrinitwrglossbefore
1596 \else
1597   \do@wrglossary{#2}%
1598 \fi
```

As the original definition:

```
1599 \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
1600 }
```

```
1601 \define@key{glossadd}{thevalue}{\def\@glstr@thevalue{#1}}
```

```
1602 \define@key{glossadd}{theHvalue}{\def\@glstr@theHvalue{#1}}
```

`\glsadd` Redefine to include `\@glstr@record`

```
1603 \renewrobustcmd*{\glsadd}[2][ ]{%
1604   \@gl@adjustmode
1605   \@glstr@record{#1}{#2}{glossadd}%
1606   \glsdoifexists{#2}%
1607   {%
1608     \let\@gl@numberformat\@glstr@defaultnumberformat
1609     \edef\@gl@counter{\csname glo@\gl@detoklabel{#2}@counter\endcsname}%
1610     \def\@glstr@thevalue{%
1611       \def\@glstr@theHvalue{\@glstr@thevalue}%
1612       \setkeys{glossadd}{#1}%
1613       \ifdefempty{\@glstr@thevalue}%
1614       {%
1615         \@gl@saveentrycounter
1616       }%
1617       {%
1618         \let\theHgl@entrycounter\@glstr@thevalue
1619         \def\theHgl@entrycounter{\@glstr@theHvalue}%
1620       }%
1621       \@do@wrglossary{#2}%
1622     }%
1623 }
```

`@field@linkdefs` Default settings for `\@gl@field@link`

```
1624 \newcommand*{\@glstr@field@linkdefs}{%
1625   \let\glstrifwasfirstuse\@secondoftwo
1626   \let\gl@ifplural\@secondoftwo
1627   \let\gl@scapscase\@firstofthree
1628   \let\gl@insert\@empty
1629 }
```

Redefine the field link commands that need to modify the above. Also add accessibility support and set the abbreviation styles if required.

assignfieldfont

```
1630 \newcommand*{\glxtrassignfieldfont}[1]{%
1631   \ifglstryexists{#1}%
1632   {%
1633     \ifglshasshort{#1}%
1634     {%
1635       \glsetabbrvfmt{\glscategory{#1}}%
1636       \glrifregular{#1}%
1637       {\let\@gls@field@font\glxtrregularfont}%
1638       {\let\@gls@field@font\@firstofone}%
1639     }%
1640     {%
1641       \glrifnotregular{#1}%
1642       {\let\@gls@field@font\@firstofone}%
1643       {\let\@gls@field@font\glxtrregularfont}%
1644     }%
1645   }%
1646   {%
1647     \let\@gls@field@font\@gobble
1648   }%
1649 }
```

\@glstext@ The abbreviation format may also need setting.

```
1650 \def\@glstext@#1#2[#3]{%
1651   \glxtrassignfieldfont{#2}%
1652   \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccesstext{#2}#3}}%
1653 }
```

\@GLStext@ All uppercase version of \@glstext. The abbreviation format may also need setting.

```
1654 \def\@GLStext@#1#2[#3]{%
1655   \glxtrassignfieldfont{#2}%
1656   \@gls@field@link[\let\glscapscase\@thirdofthree]{#1}{#2}%
1657   {\@gls@field@font{\GLSaccesstext{#2}\mfirstucMakeUppercase{#3}}}%
1658 }
```

\@Glstext@ First letter uppercase version. The abbreviation format may also need setting.

```
1659 \def\@Glstext@#1#2[#3]{%
1660   \glxtrassignfieldfont{#2}%
1661   \@gls@field@link[\let\glscapscase\@secondofthree]{#1}{#2}%
1662   {\@gls@field@font{\Glsaccesstext{#2}#3}}%
1663 }
```

Version 1.07 ensures that \glsfirst etc honours the nohyperfirst attribute. Allow a convenient way for the user to revert to ignoring this attribute for these commands.

checknohyperfirst

```
1664 \newcommand*{\glxtrchecknohyperfirst}[1]{%
1665   \glrifattribute{#1}{nohyperfirst}{true}{\KVglslink@hyperfalse}{}%
1666 }
```

`\@glsfirst@` No case changing version. The abbreviation format may also need setting.

```
1667 \def\@glsfirst@#1#2[#3]{%
1668   \glsxtrassignfieldfont{#2}%
    Ensure that \glsfirst honours the nohyperfirst attribute.
1669   \@gls@field@link
1670   [\let\glsxtrifwasfirstuse\@firstoftwo
1671     \glsxtrchecknohyperfirst{#2}%
1672   ]{#1}{#2}%
1673   {\@gls@field@font{\glsaccessfirst{#2}#3}}%
1674 }
```

`\@Glsfirst@` First letter uppercase version. The abbreviation format may also need setting.

```
1675 \def\@Glsfirst@#1#2[#3]{%
1676   \glsxtrassignfieldfont{#2}%
    Ensure that \Glsfirst honours the nohyperfirst attribute.
1677   \@gls@field@link
1678   [\let\glsxtrifwasfirstuse\@firstoftwo
1679     \let\glscapscase\@secondofthree
1680     \glsxtrchecknohyperfirst{#2}%
1681   ]%
1682   {#1}{#2}{\@gls@field@font{\Glsaccessfirst{#2}#3}}%
1683 }
```

`\@GLSfirst@` All uppercase version. The abbreviation format may also need setting.

```
1684 \def\@GLSfirst@#1#2[#3]{%
1685   \glsxtrassignfieldfont{#2}%
    Ensure that \GLSfirst honours the nohyperfirst attribute.
1686   \@gls@field@link
1687   [\let\glsxtrifwasfirstuse\@firstoftwo
1688     \let\glscapscase\@thirdofthree
1689     \glsxtrchecknohyperfirst{#2}%
1690   ]%
1691   {#1}{#2}{\@gls@field@font{\GLSaccessfirst{#2}\mfirstucMakeUppercase{#3}}}%
1692 }
```

`\@glsplural@` No case changing version. The abbreviation format may also need setting.

```
1693 \def\@glsplural@#1#2[#3]{%
1694   \glsxtrassignfieldfont{#2}%
1695   \@gls@field@link[\let\glsifplural\@firstoftwo]{#1}{#2}%
1696   {\@gls@field@font{\glsaccessplural{#2}#3}}%
1697 }
```

`\@Glsplural@` First letter uppercase version. The abbreviation format may also need setting.

```
1698 \def\@Glsplural@#1#2[#3]{%
1699   \glsxtrassignfieldfont{#2}%
1700   \@gls@field@link
1701   [\let\glsifplural\@firstoftwo
```

```

1702 \let\glscapscase\@secondofthree
1703 ]%
1704 {#1}{#2}{\@gls@field@font{\Glsaccessplural{#2}#3}}%
1705 }

```

\@GLSplural@ All uppercase version. The abbreviation format may also need setting.

```

1706 \def\@GLSplural@#1#2[#3]{%
1707 \glstrassignfieldfont{#2}%
1708 \@gls@field@link
1709 [\let\glsifplural\@firstoftwo
1710 \let\glscapscase\@thirdofthree
1711 ]%
1712 {#1}{#2}{\@gls@field@font{\Glsaccessplural{#2}\mfirstucMakeUppercase{#3}}}%
1713 }

```

glsfirstplural@ No case changing version. The abbreviation format may also need setting.

```

1714 \def\@glsfirstplural@#1#2[#3]{%
1715 \glstrassignfieldfont{#2}%
1716 \gls@field@link
1717 [\let\glstrifwasfirstuse\@firstoftwo
1718 \let\glsifplural\@firstoftwo
1719 \glstrchecknohyperfirst{#2}%
1720 ]%
1721 {#1}{#2}{\@gls@field@font{\glsaccessfirstplural{#2}#3}}%
1722 }

```

GLsfirstplural@ First letter uppercase version. The abbreviation format may also need setting.

```

1723 \def\@GLsfirstplural@#1#2[#3]{%
1724 \glstrassignfieldfont{#2}%
1725 \gls@field@link
1726 [\let\glstrifwasfirstuse\@firstoftwo
1727 \let\glsifplural\@firstoftwo
1728 \let\glscapscase\@secondofthree
1729 \glstrchecknohyperfirst{#2}%
1730 ]%
1731 {#1}{#2}{\@gls@field@font{\Glsaccessfirstplural{#2}#3}}%
1732 }

```

GLSfirstplural@ All uppercase version. The abbreviation format may also need setting.

```

1733 \def\@GLSfirstplural@#1#2[#3]{%
1734 \glstrassignfieldfont{#2}%
1735 \gls@field@link
1736 [\let\glstrifwasfirstuse\@firstoftwo
1737 \let\glsifplural\@firstoftwo

```

```

1738 \let\glscapscase\@thirdofthree
1739 \glstrchecknohyperfirst{#2}%
1740 ]%
1741 {#1}{#2}%
1742 {\@gls@field@font{\GLSaccessfirstplural{#2}\mfirstucMakeUppercase{#3}}}%
1743 }

```

`\@glsname@` Redefine to use accessibility support. The abbreviation format may also need setting.

```

1744 \def\@glsname@#1#2[#3]{%
1745 \glstrassignfieldfont{#2}%
1746 \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccessname{#2}#3}}%
1747 }

```

`\@Glsname@` First letter uppercase version. The abbreviation format may also need setting.

```

1748 \def\@Glsname@#1#2[#3]{%
1749 \glstrassignfieldfont{#2}%
1750 \@gls@field@link
1751 [\let\glscapscase\@secondoftwo]{#1}{#2}%
1752 {\@gls@field@font{\Glsaccessname{#2}#3}}%
1753 }

```

`\@GLSname@` All uppercase version. The abbreviation format may also need setting.

```

1754 \def\@GLSname@#1#2[#3]{%
1755 \glstrassignfieldfont{#2}%
1756 \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1757 {#1}{#2}%
1758 {\@gls@field@font{\GLSaccessname{#2}\mfirstucMakeUppercase{#3}}}%
1759 }

```

`\@glsdesc@`

```

1760 \def\@glsdesc@#1#2[#3]{%
1761 \glstrassignfieldfont{#2}%
1762 \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccessdesc{#2}#3}}%
1763 }

```

`\@Glsdesc@` First letter uppercase version.

```

1764 \def\@Glsdesc@#1#2[#3]{%
1765 \glstrassignfieldfont{#2}%
1766 \@gls@field@link
1767 [\let\glscapscase\@secondoftwo]{#1}{#2}%
1768 {\@gls@field@font{\Glsaccessdesc{#2}#3}}%
1769 }

```

`\@GLSdesc@` All uppercase version.

```

1770 \def\@GLSdesc@#1#2[#3]{%
1771 \glstrassignfieldfont{#2}%
1772 \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1773 {#1}{#2}{\@gls@field@font{\GLSaccessdesc{#2}\mfirstucMakeUppercase{#3}}}%
1774 }

```



@glsdescplural@ No case-changing version.

```
1775 \def@glsdescplural@#1#2[#3]{%
1776   \glstrassignfieldfont{#2}%
1777   \@gls@field@link
1778   [\let\glscapscase\@secondoftwo
1779    \let\glsifplural\@firstoftwo
1780   ]{#1}{#2}{\@gls@field@font{\glsaccessdescplural{#2}#3}}%
1781 }
```

@Glsdescplural@ First letter uppercase version.

```
1782 \def@Glsdescplural@#1#2[#3]{%
1783   \glstrassignfieldfont{#2}%
1784   \@gls@field@link
1785   [\let\glscapscase\@secondoftwo
1786    \let\glsifplural\@firstoftwo
1787   ]{#1}{#2}{\@gls@field@font{\Glsaccessdescplural{#2}#3}}%
1788 }
```

@GLSdescplural@ All uppercase version.

```
1789 \def@GLSdesc@#1#2[#3]{%
1790   \glstrassignfieldfont{#2}%
1791   \@gls@field@link
1792   [\let\glscapscase\@thirdoftwo
1793    \let\glsifplural\@firstoftwo
1794   ]%
1795   {#1}{#2}%
1796   {\@gls@field@font{\GLSaccessdescplural{#2}\mfirstucMakeUppercase{#3}}}%
1797 }
```

\@glssymbol@

```
1798 \def@glssymbol@#1#2[#3]{%
1799   \glstrassignfieldfont{#2}%
1800   \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccesssymbol{#2}#3}}%
1801 }
```

\@Glsymbol@ First letter uppercase version.

```
1802 \def@Glsymbol@#1#2[#3]{%
1803   \glstrassignfieldfont{#2}%
1804   \@gls@field@link
1805   [\let\glscapscase\@secondoftwo]%
1806   {#1}{#2}{\@gls@field@font{\Glsaccesssymbol{#2}#3}}%
1807 }
```

\@GLSsymbol@ All uppercase version.

```
1808 \def@GLSsymbol@#1#2[#3]{%
1809   \glstrassignfieldfont{#2}%
1810   \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1811   {#1}{#2}{\@gls@field@font{\GLSaccesssymbol{#2}\mfirstucMakeUppercase{#3}}}%
1812 }
```

lssymbolplural@ No case-changing version.

```
1813 \def\@glssymbolplural@#1#2[#3]{%
1814   \glstrassignfieldfont{#2}%
1815   \@gls@field@link
1816   [\let\glscapscase\@secondoftwo
1817    \let\glsifplural\@firstoftwo
1818   ]{#1}{#2}{\@gls@field@font{\glsaccesssymbolplural{#2}#3}}%
1819 }
```

lssymbolplural@ First letter uppercase version.

```
1820 \def\@Glsymbolplural@#1#2[#3]{%
1821   \glstrassignfieldfont{#2}%
1822   \@gls@field@link
1823   [\let\glscapscase\@secondoftwo
1824    \let\glsifplural\@firstoftwo
1825   ]{#1}{#2}{\@gls@field@font{\Glsaccesssymbolplural{#2}#3}}%
1826 }
```

LSsymbolplural@ All uppercase version.

```
1827 \def\@GLSsymbol@#1#2[#3]{%
1828   \glstrassignfieldfont{#2}%
1829   \@gls@field@link
1830   [\let\glscapscase\@thirdoftwo
1831    \let\glsifplural\@firstoftwo
1832   ]%
1833   {#1}{#2}%
1834   {\@gls@field@font{\GLSaccesssymbolplural{#2}\mfirstucMakeUppercase{#3}}}%
1835 }
```

\@Glsuseri@ First letter uppercase version.

```
1836 \def\@Glsuseri@#1#2[#3]{%
1837   \glstrassignfieldfont{#2}%
1838   \@gls@field@link
1839   [\let\glscapscase\@secondoftwo]{#1}{#2}%
1840   {\@gls@field@font{\Glsentryuseri{#2}#3}}%
1841 }
```

\@GLSuseri@ All uppercase version.

```
1842 \def\@GLSuseri@#1#2[#3]{%
1843   \glstrassignfieldfont{#2}%
1844   \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1845   {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}}%
1846 }
```

\@Glsuserii@ First letter uppercase version.

```
1847 \def\@Glsuserii@#1#2[#3]{%
1848   \glstrassignfieldfont{#2}%
1849   \@gls@field@link
```

```

1850 [\let\glscapscase\@secondoftwo]%
1851 {#1}{#2}{\@gls@field@font{\Glsentryuserii{#2}#3}}%
1852 }

```

\@GLSuserii@ All uppercase version.

```

1853 \def\@GLSuserii@#1#2[#3]{%
1854 \glstrassignfieldfont{#2}%
1855 \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1856 {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glentryuserii{#2}#3}}}%
1857 }

```

\@Glsuseriii@ First letter uppercase version.

```

1858 \def\@Glsuseriii@#1#2[#3]{%
1859 \glstrassignfieldfont{#2}%
1860 \@gls@field@link
1861 [\let\glscapscase\@secondoftwo]%
1862 {#1}{#2}{\@gls@field@font{\Glsentryuseriii{#2}#3}}%
1863 }

```

\@GLSuseriii@ All uppercase version.

```

1864 \def\@GLSuseriii@#1#2[#3]{%
1865 \glstrassignfieldfont{#2}%
1866 \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1867 {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glentryuseriii{#2}#3}}}%
1868 }

```

\@Glsuseriv@ First letter uppercase version.

```

1869 \def\@Glsuseriv@#1#2[#3]{%
1870 \glstrassignfieldfont{#2}%
1871 \@gls@field@link
1872 [\let\glscapscase\@secondoftwo]%
1873 {#1}{#2}{\@gls@field@font{\Glsentryuseriv{#2}#3}}%
1874 }

```

\@GLSuseriv@ All uppercase version.

```

1875 \def\@GLSuseriv@#1#2[#3]{%
1876 \glstrassignfieldfont{#2}%
1877 \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1878 {#1}{#2}%
1879 {\@gls@field@font{\mfirstucMakeUppercase{\glentryuseriv{#2}#3}}}%
1880 }

```

\@Glsuserv@ First letter uppercase version.

```

1881 \def\@Glsuserv@#1#2[#3]{%
1882 \glstrassignfieldfont{#2}%
1883 \@gls@field@link
1884 [\let\glscapscase\@secondoftwo]%
1885 {#1}{#2}{\@gls@field@font{\Glsentryuserv{#2}#3}}%
1886 }

```

\@GLSuserv@ All uppercase version.

```
1887 \def\@GLSuserv@#1#2[#3]{%
1888   \glstrassignfieldfont{#2}%
1889   \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1890   {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glentryuserv{#2}#3}}}%
1891 }
```

\@Glsuservi@ First letter uppercase version.

```
1892 \def\@Glsuservi@#1#2[#3]{%
1893   \glstrassignfieldfont{#2}%
1894   \@gls@field@link
1895   [\let\glscapscase\@secondoftwo]%
1896   {#1}{#2}{\@gls@field@font{\Glsentryuservi{#2}#3}}%
1897 }
```

\@GLSuservi@ All uppercase version.

```
1898 \def\@GLSuservi@#1#2[#3]{%
1899   \glstrassignfieldfont{#2}%
1900   \@gls@field@link[\let\glscapscase\@thirdoftwo]%
1901   {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glentryuservi{#2}#3}}}%
1902 }
```

Commands like \acrshort already set \glsifplural, but they don't set \glxtrifwasfirstuse so they need adjusting.

\@acrshort No case change.

```
1903 \def\@acrshort#1#2[#3]{%
1904   \glsdoifexists{#2}%
1905   {%
1906     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
1907     \let\glxtrifwasfirstuse\@secondoftwo
1908     \let\glsifplural\@secondoftwo
1909     \let\glscapscase\@firstofthree
1910     \let\glsinsert\@empty
1911     \def\glscustomtext{%
1912       \acronymfont{\glsaccesssshort{#2}}#3%
1913     }%
1914     \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
1915   }%
1916   \glspostlinkhook
1917 }
```

\@Acrshort First letter uppercase.

```
1918 \def\@Acrshort#1#2[#3]{%
1919   \glsdoifexists{#2}%
1920   {%
1921     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
1922     \let\glxtrifwasfirstuse\@secondoftwo
```

```

1923 \let\glsifplural\@secondoftwo
1924 \let\glscapscase\@secondofthree
1925 \let\glsinsert\@empty
1926 \def\glscustomtext{%
1927     \acronymfont{\Glsaccessshort{#2}}#3%
1928 }%
1929 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
1930 }%
1931 \glspostlinkhook
1932 }

```

\@ACRshort All uppercase.

```

1933 \def\@ACRshort#1#2[#3]{%
1934     \glsdoifexists{#2}%
1935     {%
1936         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
1937         \let\glstrifwasfirstuse\@secondoftwo
1938         \let\glsifplural\@secondoftwo
1939         \let\glsapscase\@thirdofthree
1940         \let\glsinsert\@empty
1941         \def\glscustomtext{%
1942             \mfirstucMakeUppercase{\acronymfont{\glsaccessshort{#2}}#3}%
1943         }%
1944         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
1945     }%
1946     \glspostlinkhook
1947 }

```

\@acrshortpl No case change.

```

1948 \def\@acrshortpl#1#2[#3]{%
1949     \glsdoifexists{#2}%
1950     {%
1951         \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
1952         \let\glstrifwasfirstuse\@secondoftwo
1953         \let\glsifplural\@firstoftwo
1954         \let\glsapscase\@firstofthree
1955         \let\glsinsert\@empty
1956         \def\glscustomtext{%
1957             \acronymfont{\glsaccessshortpl{#2}}#3%
1958         }%
1959         \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
1960     }%
1961     \glspostlinkhook
1962 }

```

\@Acrshortpl First letter uppercase.

```

1963 \def\@Acrshortpl#1#2[#3]{%
1964     \glsdoifexists{#2}%
1965     {%

```

```

1966 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
1967 \let\glxtrifwasfirstuse\@secondoftwo
1968 \let\gl@sifplural\@firstoftwo
1969 \let\glscapscase\@secondofthree
1970 \let\gl$insert\@empty
1971 \def\glscustomtext{%
1972     \acronymfont{\Glsaccessshortpl{#2}}#3%
1973 }%
1974 \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
1975 }%
1976 \glspostlinkhook
1977 }

```

\@ACRshortpl All uppercase.

```

1978 \def\@ACRshortpl#1#2[#3]{%
1979     \glsdofexists{#2}%
1980     {%
1981         \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
1982         \let\glxtrifwasfirstuse\@secondoftwo
1983         \let\gl@sifplural\@firstoftwo
1984         \let\glscapscase\@thirdofthree
1985         \let\gl$insert\@empty
1986         \def\glscustomtext{%
1987             \mfirstucMakeUppercase{\acronymfont{\Glsaccessshortpl{#2}}#3}%
1988         }%
1989         \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
1990     }%
1991     \glspostlinkhook
1992 }

```

\@acrlong No case change.

```

1993 \def\@acrlong#1#2[#3]{%
1994     \glsdofexists{#2}%
1995     {%
1996         \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
1997         \let\glxtrifwasfirstuse\@secondoftwo
1998         \let\gl@sifplural\@secondoftwo
1999         \let\glscapscase\@firstofthree
2000         \let\gl$insert\@empty
2001         \def\glscustomtext{%
2002             \acronymfont{\Glsaccesslong{#2}}#3%
2003         }%
2004         \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
2005     }%
2006     \glspostlinkhook
2007 }

```

\@Acrlong First letter uppercase.

```

2008 \def\@Acrlong#1#2[#3]{%

```

```

2009 \glsdoifexists{#2}%
2010 {%
2011   \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
2012   \let\glxtrifwasfirstuse\@secondoftwo
2013   \let\gl@ifplural\@secondoftwo
2014   \let\glscapscase\@secondofthree
2015   \let\glinsert\@empty
2016   \def\glscustomtext{%
2017     \acronymfont{\Glsaccesslong{#2}}#3%
2018   }%
2019   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
2020 }%
2021 \glspostlinkhook
2022 }

```

\@ACRlong All uppercase.

```

2023 \def\@ACRlong#1#2[#3]{%
2024   \glsdoifexists{#2}%
2025   {%
2026     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
2027     \let\glxtrifwasfirstuse\@secondoftwo
2028     \let\gl@ifplural\@secondoftwo
2029     \let\glscapscase\@thirdofthree
2030     \let\glinsert\@empty
2031     \def\glscustomtext{%
2032       \mfirstucMakeUppercase{\acronymfont{\Glsaccesslong{#2}}#3}%
2033     }%
2034     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
2035   }%
2036   \glspostlinkhook
2037 }

```

\@acrlongpl No case change.

```

2038 \def\@acrlongpl#1#2[#3]{%
2039   \glsdoifexists{#2}%
2040   {%
2041     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
2042     \let\glxtrifwasfirstuse\@secondoftwo
2043     \let\gl@ifplural\@firstoftwo
2044     \let\glscapscase\@firstofthree
2045     \let\glinsert\@empty
2046     \def\glscustomtext{%
2047       \acronymfont{\Glsaccesslongpl{#2}}#3%
2048     }%
2049     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
2050   }%
2051   \glspostlinkhook
2052 }

```

`\@Acrlongpl` First letter uppercase.

```
2053 \def\@Acrlongpl#1#2[#3]{%
2054   \glsdoifexists{#2}%
2055   {%
2056     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
2057     \let\glxtrifwasfirstuse\@secondoftwo
2058     \let\gl@sifplural\@firstoftwo
2059     \let\glscapscase\@secondofthree
2060     \let\gl$insert\@empty
2061     \def\glscustomtext{%
2062       \acronymfont{\Glsaccesslongpl{#2}}#3%
2063     }%
2064     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
2065   }%
2066   \glspostlinkhook
2067 }
```

`\@ACRlongpl` All uppercase.

```
2068 \def\@ACRlongpl#1#2[#3]{%
2069   \glsdoifexists{#2}%
2070   {%
2071     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
2072     \let\glxtrifwasfirstuse\@secondoftwo
2073     \let\gl@sifplural\@firstoftwo
2074     \let\glscapscase\@thirdofthree
2075     \let\gl$insert\@empty
2076     \def\glscustomtext{%
2077       \mfirstucMakeUppercase{\acronymfont{\Glsaccesslongpl{#2}}#3}%
2078     }%
2079     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
2080   }%
2081   \glspostlinkhook
2082 }
```

Modify `\@glsaddkey` so additional keys provided by the user can be treated in a similar way.

`\@glsaddkey`

```
2083 \renewcommand*{\@glsaddkey}[7]{%
2084   \key@ifundefined{glossentry}{#1}%
2085   {%
2086     \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
2087     \appto\@gl@keymap{, {#1}{#1}}%
2088     \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
2089     \appto\@newglossaryentryposthook{%
2090       \letcs{\@glo@tmp}{@glo@#1}%
2091       \gl@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
2092     }%
2093     \newcommand*{#3}[1]{\@gl@entry@field{##1}{#1}}%
```



```
2094 \newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
```

Now for the commands with links. First the version with no case change (same as before):

```
2095 \ifcsdef{@glS@user@#1@}%
2096 {%
2097   \PackageError{glossaries}%
2098   {Can't define '\string#5' as helper command
2099   '\expandafter\string\csname @glS@user@#1@\endcsname' already
2100   exists}%
2101   }%
2102 }%
2103 {%
2104   \expandafter\newcommand\expandafter*\expandafter
2105   {\csname @glS@user@#1@\endcsname}[2][ ]{%
2106     \new@ifnextchar[%
2107       {\csuse{@glS@user@#1@}{##1}{##2}}%
2108       {\csuse{@glS@user@#1@}{##1}{##2}[ ]}%
2109     \csdef{@glS@user@#1@}##1##2[##3]{%
2110       \@glS@field@link{##1}{##2}{#3{##2}##3}%
2111     }%
2112     \newrobustcmd*{#5}{%
2113       \expandafter\@glS@hyp@opt\csname @glS@user@#1@\endcsname}%
2114     }%
```

Next the version with the first letter converted to upper case (modified):

```
2115 \ifcsdef{@Gls@user@#1@}%
2116 {%
2117   \PackageError{glossaries}%
2118   {Can't define '\string#6' as helper command
2119   '\expandafter\string\csname @Gls@user@#1@\endcsname' already
2120   exists}%
2121   }%
2122 }%
2123 {%
2124   \expandafter\newcommand\expandafter*\expandafter
2125   {\csname @Gls@user@#1@\endcsname}[2][ ]{%
2126     \new@ifnextchar[%
2127       {\csuse{@Gls@user@#1@}{##1}{##2}}%
2128       {\csuse{@Gls@user@#1@}{##1}{##2}[ ]}%
2129     \csdef{@Gls@user@#1@}##1##2[##3]{%
2130       \@glS@field@link[\let\glScapsCase\@secondofthree]%
2131       {##1}{##2}{#4{##2}##3}%
2132     }%
2133     \newrobustcmd*{#6}{%
2134       \expandafter\@glS@hyp@opt\csname @Gls@user@#1@\endcsname}%
2135     }%
```

Finally the all caps version (modified):

```
2136 \ifcsdef{@GLS@user@#1@}%
2137 {%
2138   \PackageError{glossaries}%
```

```

2139     {Can't define '\string#7' as helper command
2140     '\expandafter\string\csname @GLS@user@#1\endcsname' already
2141     exists}%
2142   }%
2143 }%
2144 {%
2145   \expandafter\newcommand\expandafter*\expandafter
2146   {\csname @GLS@user@#1\endcsname}[2][\%
2147     \new@ifnextchar[%
2148       {\csuse{@GLS@user@#1@}{##1}{##2}}%
2149       {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
2150   \csdef{@GLS@user@#1@}##1##2[##3]{%
2151     \@gls@field@link[\let\gls@caps@case\@thirdofthree]%
2152     {##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
2153   }%
2154   \newrobustcmd*{#7}{%
2155     \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
2156   }%
2157 }%
2158 {%
2159   \PackageError{glossaries-extra}{Key '#1' already exists}{}%
2160 }%
2161 }

```

checkfirsthyper Old versions of glossaries don't define this, so provide it just in case it hasn't been defined.

```

2162 \providecommand*{\@gls@link@nocheckfirsthyper}{}

```

checkfirsthyper Modify check to determine if the hyperlink should be automatically suppressed, but save the original in case the acronyms are restored.

```

2163 \let\@gls@xtr@org@checkfirsthyper\@gls@link@checkfirsthyper
2164 \renewcommand*{\@gls@link@checkfirsthyper}{%

```

\ifglsused isn't useful in the post link hook as it's already been unset by then, so define a command that can be used in the post link hook. Since \@gls@link@checkfirsthyper is only used by commands like \gls but not by other commands, this seems the best place to put it.

```

2165   \ifglsused{\glslabel}%
2166   {\let\gls@xtrifwasfirstuse\@secondoftwo}
2167   {\let\gls@xtrifwasfirstuse\@firstoftwo}%

```

Store the category label for convenience.

```

2168   \edef\gls@categorylabel{\gls@category{\glslabel}}%
2169   \ifglsused{\glslabel}%
2170   {%
2171     \glsifcategoryattribute{\gls@categorylabel}{nohypernext}{true}%
2172     {\KV@glslink@hyperfalse}{}%
2173   }%
2174   {%
2175     \glsifcategoryattribute{\gls@categorylabel}{nohyperfirst}{true}%

```

```

2176     {\KV@glslink@hyperfalse}{}%
2177 }%
2178 \glslinkcheckfirsthyperhook
2179 }

```

ablehyperinlist This command was introduced in glossaries v4.19. If it hasn't been defined, we're using an earlier version, in which case the nohyper attribute can't be implemented.

```

2180 \ifdef\do@gl:disablehyperinlist
2181 {%
2182   \let\@gl:extr@do@gl:disablehyperinlist\do@gl:disablehyperinlist
2183   \renewcommand*{\do@gl:disablehyperinlist}{%
2184     \@gl:extr@do@gl:disablehyperinlist
2185     \gl:sifattribute{\gl:label}{nohyper}{true}{\KV@glslink@hyperfalse}{}%
2186   }
2187 }
2188 {}

```

Define a noindex key to prevent writing information to the external file.

```

2189 \define@boolkey{gl:link}{noindex}[true]{}
2190 \KV@glslink@noindexfalse

```

If \@gl:s@setdefault@gl:link@opts has been defined (glossaries v4.20) use it to set the default keys in \@gl:link.

lt@gl:link@opts

```

2191 \ifdef\@gl:s@setdefault@gl:link@opts
2192 {
2193   \renewcommand*{\@gl:s@setdefault@gl:link@opts}{%
2194     \KV@glslink@noindexfalse
2195     \@gl:extr:setaliasnoindex
2196   }
2197 }
2198 {

```

Not defined so prepend it to \do@gl:disablehyperinlist to achieve the same effect.

```

2199   \newcommand*{\@gl:s@setdefault@gl:link@opts}{%
2200     \KV@glslink@noindexfalse
2201     \@gl:extr:setaliasnoindex
2202   }
2203   \pretol\do@gl:disablehyperinlist{\@gl:s@setdefault@gl:link@opts}
2204 }

```

setaliasnoindex Allow user to hook into the alias noindex setting. Default behaviour switches off indexing for aliases. If the record option is on, this will have been defined to do nothing. (bib2gls will deal with records for aliased entries.)

```

2205 \providecommand*{\gl:extr:setaliasnoindex}{%
2206   \KV@glslink@noindextrue
2207 }

```

setaliasnoindex

```
2208 \newcommand*{\@glxtrsetaliasnoindex}{%
2209   \ifglshasfield{alias}{\glslabel}%
2210   {%
2211     \let\glxtrindexaliased\@glxtrindexaliased
2212     \glxtrsetaliasnoindex
2213     \let\glxtrindexaliased\@no@glxtrindexaliased
2214   }%
2215   }%
2216 }
```

xtrindexaliased

```
2217 \newcommand{\@glxtrindexaliased}{%
2218   \ifKV@glslink@noindex
2219   \else
2220     \begingroup
2221     \let\@glsnnumberformat\@glxtr@defaultnumberformat
2222     \edef\@glscounter{\csname glo@glsdetoklabel{\glslabel}@counter\endcsname}%
2223     \glxtr@saveentrycounter
2224     \@@do@wrglossary{\glxtralias{\glslabel}}%
2225     \endgroup
2226   \fi
2227 }
```

xtrindexaliased

```
2228 \newcommand{\@no@glxtrindexaliased}{%
2229   \PackageError{glossaries-extra}{\string\glxtrindexaliased\space
2230   not permitted outside definition of \string\glxtrsetaliasnoindex}%
2231   {}%
2232 }
```

xtrindexaliased Provide a command to redirect alias indexing, but only allow it to be used within \glxtrsetaliasnoindex.

```
2233 \let\glxtrindexaliased\@no@glxtrindexaliased
```

tDefaultGlsOpts Set the default options for \glslink etc.

```
2234 \newcommand*{\GlsXtrSetDefaultGlsOpts}[1]{%
2235   \renewcommand*{\@glscsetdefault@glslink@opts}{%
2236     \setkeys{glslink}{#1}%
2237     \@glxtrsetaliasnoindex
2238   }%
2239 }
```

lsxtrifindexing Provide user level command to access it in \glswriteentry.

```
2240 \newcommand*{\glxtrifindexing}[2]{%
2241   \ifKV@glslink@noindex #2\else #1\fi
2242 }
```

\glswriteentry Redefine to test for indexonlyfirst category attribute.

```

2243 \renewcommand*{\glswriteentry}[2]{%
2244   \glxtrifindexing
2245   {%
2246     \ifglindexonlyfirst
2247       \ifglused{#1}
2248       {\glxtrdoautoindexname{#1}{dualindex}}%
2249       {#2}%
2250     \else
2251       \gl@ifattribute{#1}{indexonlyfirst}{true}%
2252       {\ifglused{#1}
2253        {\glxtrdoautoindexname{#1}{dualindex}}%
2254        {#2}}%
2255       {#2}%
2256     \fi
2257   }%
2258   {}%
2259 }

```

`@do@wrglossary` Hook into glossary indexing command so that it can also use `\index` at the same time if required and add user hook.

```

2260 \appto@do@wrglossary{\glxtr@do@wrindex
2261   \glxtrdowrglossaryhook{\glsl@label}%
2262 }

```

(The label can be obtained from `\glsl@label` at this point.)

Similarly for the “noidx” version:

`s@noidxglossary`

```

2263 \appto\glsl@noidxglossary{\glxtr@do@wrindex
2264   \glxtrdowrglossaryhook{\glsl@label}%
2265 }

```

`xtr@do@wrindex`

```

2266 \newcommand*{\glxtr@do@wrindex}{%
2267   \glxtrdoautoindexname{\glsl@label}{dualindex}%
2268 }

```

`owrglossaryhook` Allow user to hook into indexing code. (Always used by `\glsladd`. Used by `\glsl` when indexing, which may or may not occur depending on the indexing settings.)

```

2269 \newcommand*{\glxtrdowrglossaryhook}[1]{%

```

`glsl@alt@hyp@opt` Commands like `\glsl` have a star or plus version. Provide a third symbol that the user can adapt for convenience.

```

2270 \newcommand*{\glsl@alt@hyp@opt}[1]{%
2271   \let\glslinkvar\@firstofthree
2272   \let\glsl@hyp@opt@cs#1\relax
2273   \ifstar{\s@glsl@hyp@opt}%
2274   {\@ifnextchar+%

```

```

2275 {\@firstoftwo{\p@glshyp@opt}}%
2276 {%
2277 \expandafter\@ifnextchar\@glshalt@hyp@opt@char
2278 {\@firstoftwo{\@alt@glshyp@opt}}%
2279 {#1}%
2280 }%
2281 }%
2282 }

```

`\alt@glshyp@opt` User version

```

2283 \newcommand*{\@alt@glshyp@opt}[1][{}]{%
2284 \let\glslinkvar\@firstofthree
2285 \expandafter\@glshyp@opt@cs\expandafter[\@glshalt@hyp@opt@keys,#1]}

```

`\lt@hyp@opt@char` Contains the character used as the command modifier.

```

2286 \newcommand*{\@glshalt@hyp@opt@char}{}

```

`\lt@hyp@opt@keys` Contains the option list used as the command modifier.

```

2287 \newcommand*{\@glshalt@hyp@opt@keys}{}

```

`\rSetAltModifier`

```

2288 \newcommand*{\GlsXtrSetAltModifier}[2]{%
2289 \let\@glshyp@opt\@glshalt@hyp@opt
2290 \def\@glshalt@hyp@opt@char{#1}%
2291 \def\@glshalt@hyp@opt@keys{#2}%
2292 }

```

`\glsdohyperlink` Unpleasant complications can occur if the text or first key etc contains `\glsh`, particularly if there are hyperlinks. To get around this problem, patch `\glsdohyperlink` so that it temporarily makes `\glsh` behave like `\glstext[hyper=false,noindex]`. (This will be overridden if the user explicitly cancels either of those options in the optional argument of `\glsh` or using the plus version.) This also patches the short form commands like `\acrshort` and `\glsxtrshort` to use `\glstryshort` and, similarly, the long form commands like `\acrlong` and `\glsxtrlong` to use `\glstrylong`. Added attribute check.

```

2293 \renewcommand*{\glsdohyperlink}[2]{%
2294 \glshasattribute{\glslabel}{targeturl}%
2295 {%
2296 \glshasattribute{\glslabel}{targetname}%
2297 {%
2298 \glshasattribute{\glslabel}{targetcategory}%
2299 {%
2300 \hyperref{\glsggetattribute{\glslabel}{targeturl}}{%
2301 \glsggetattribute{\glslabel}{targetcategory}}%
2302 \glsggetattribute{\glslabel}{targetname}}%
2303 {\glsxtrprotectlinks#2}}%
2304 }%
2305 {%
2306 \hyperref{\glsggetattribute{\glslabel}{targeturl}}%

```

```

2307         {}%
2308         {\glsgetattribute{\glslabel}{targetname}}%
2309         {\glsxtrprotectlinks#2}}%
2310     }%
2311 }%
2312 {%
2313     \href{\glsgetattribute{\glslabel}{targeturl}}%
2314     {\glsxtrprotectlinks#2}}%
2315 }%
2316 }%
2317 {%

```

Check for alias.

```

2318     \glsfieldfetch{\glslabel}{alias}{\gloaliaslabel}%
2319     \ifvoid\gloaliaslabel
2320     {%
2321         \glsxtrhyperlink{#1}{\glsxtrprotectlinks#2}}%
2322     }%
2323     {%

```

Redirect link to the alias target.

```

2324     \glsxtrhyperlink
2325     {\glolinkprefix\glsdetoklabel{\gloaliaslabel}}%
2326     {\glsxtrprotectlinks#2}}%
2327 }%
2328 }%
2329 }

```

`\glsxtrhyperlink` Allows integration with the base glossaries package's `debug=showtargets` option.

```

2330 \ifdef\@glsshowtarget
2331 {
2332     \newcommand{\glsxtrhyperlink}[2]{%
2333         \@glsshowtarget{#1}%
2334         \hyperlink{#1}{#2}%
2335     }%
2336 }
2337 {
2338     \newcommand{\glsxtrhyperlink}[2]{\hyperlink{#1}{#2}}%
2339 }

```

`\glsdisablehyper` Redefine to set `\glslabel` (to allow it to be picked up by `\glsdohyperlink`). Also made it robust and added grouping to localise the definition of `\glslabel`. The original internal command `@glo@label` could probably be simply replaced with `\glslabel`, but it's retained in case its removal causes unexpected problems.

```

2340 \renewrobustcmd*{\glsdohyperlink}[2][\glsentrytext{\@glo@label}]{%
2341     \def\@glo@label{#2}%
2342     {\edef\glslabel{#2}%
2343         \@glslink{\glolinkprefix\glslabel}{#1}}%
2344 }

```

`glsdisablehyper` Redefine in case we have an old version of glossaries.

```
2345 \ifundef\glsdonohyperlink
2346 {%
2347   \renewcommand{\glsdisablehyper}{%
2348     \KV@glslink@hyperfalse
2349     \let\@glslink\glsdonohyperlink
2350     \let\@glstarget\@secondoftwo
2351   }
2352 }
2353 {}
```

`glsdonohyperlink` This command was only introduced in glossaries v4.20, so it may not be defined. For older glossaries versions, this won't be used if hyperref hasn't been loaded, which means the indexing will still take place. The generated text is scoped.

```
2354 \def\glsdonohyperlink#1#2{{\glstrprotectlinks #2}}
```

Reset `\@glslink` with patched versions:

```
2355 \ifcsundef{hyperlink}%
2356 {%
2357   \let\@glslink\glsdonohyperlink
2358 }%
2359 {%
2360   \let\@glslink\glsdohyperlink
2361 }
```

`glstrprotectlinks` Make `\gls` (and variants) behave like the corresponding `\glstext` (and variants) with hyperlinking and indexing off.

```
2362 \newcommand*{\glstrprotectlinks}{%
2363   \KV@glslink@hyperfalse
2364   \KV@glslink@noindextrue
2365   \let\@gls@\@glstr@p@text@
2366   \let\@Gls@\@Glsxtr@p@text@
2367   \let\@GLS@\@GLSxtr@p@text@
2368   \let\@glspl@\@glstr@p@plural@
2369   \let\@Glspl@\@Glsxtr@p@plural@
2370   \let\@GLSpl@\@GLSxtr@p@plural@
2371   \let\@glstrshort@\@glstr@p@short@
2372   \let\@Glsxtrshort@\@Glsxtr@p@short@
2373   \let\@GLSxtrshort@\@GLSxtr@p@short@
2374   \let\@glstrlong@\@glstr@p@long@
2375   \let\@Glsxtrlong@\@Glsxtr@p@long@
2376   \let\@GLSxtrlong@\@GLSxtr@p@long@
2377   \let\@glstrshortpl@\@glstr@p@shortpl@
2378   \let\@Glsxtrshortpl@\@Glsxtr@p@shortpl@
2379   \let\@GLSxtrshortpl@\@GLSxtr@p@shortpl@
2380   \let\@glstrlongpl@\@glstr@p@longpl@
2381   \let\@Glsxtrlongpl@\@Glsxtr@p@longpl@
2382   \let\@GLSxtrlongpl@\@GLSxtr@p@longpl@
```



```

2383 \let\@acrshort\@glsxtrp@acrshort@
2384 \let\@Acrshort\@Glsxtrp@acrshort@
2385 \let\@ACRshort\@GLSxtrp@acrshort@
2386 \let\@acrshortpl\@glsxtrp@acrshortpl@
2387 \let\@Acrshortpl\@Glsxtrp@acrshortpl@
2388 \let\@ACRshortpl\@GLSxtrp@acrshortpl@
2389 \let\@acrlong\@glsxtrp@acrlong@
2390 \let\@Acrlong\@Glsxtrp@acrlong@
2391 \let\@ACRlong\@GLSxtrp@acrlong@
2392 \let\@acrlongpl\@glsxtrp@acrlongpl@
2393 \let\@Acrlongpl\@Glsxtrp@acrlongpl@
2394 \let\@ACRlongpl\@GLSxtrp@acrlongpl@
2395 }

```

These protected versions need grouping to prevent the label from getting confused.

@glsxtrp@text@

```
2396 \def\@glsxtrp@text@#1#2[#3]{\@glstext@{#1}{#2}[#3]}
```

@Glsxtrp@text@

```
2397 \def\@Glsxtrp@text@#1#2[#3]{\@Glstext@{#1}{#2}[#3]}
```

@GLSxtrp@text@

```
2398 \def\@GLSxtrp@text@#1#2[#3]{\@GLStext@{#1}{#2}[#3]}
```

lsxtrp@plural@

```
2399 \def\@lsxtrp@plural@#1#2[#3]{\@glsplural@{#1}{#2}[#3]}
```

lsxtrp@plural@

```
2400 \def\@lsxtrp@plural@#1#2[#3]{\@Glsplural@{#1}{#2}[#3]}
```

LSxtrp@plural@

```
2401 \def\@LSxtrp@plural@#1#2[#3]{\@GLSplural@{#1}{#2}[#3]}
```

glsxtrp@short@

```

2402 \def\@glsxtrp@short@#1#2[#3]{%
2403   {%
2404     \glssetabbrvfmt{\glscategory{#2}}%
2405     \glsabbrvfont{\glstryshort{#2}}#3%
2406   }%
2407 }

```

Glsxtrp@short@

```

2408 \def\@Glsxtrp@short@#1#2[#3]{%
2409   {%
2410     \glssetabbrvfmt{\glscategory{#2}}%
2411     \glsabbrvfont{\Glsentryshort{#2}}#3%
2412   }%
2413 }

```

GLSxtr@p@short@

```
2414 \def\@GLSxtr@p@short@#1#2[#3]{%
2415   {%
2416     \glsetabbrvfmt{\glscategory{#2}}%
2417     \mfirstucMakeUppercase{\glabbrvfont{\glentryshort{#2}}#3}%
2418   }%
2419 }
```

sxtr@p@shortpl@

```
2420 \def\@glxtr@p@shortpl@#1#2[#3]{%
2421   {%
2422     \glsetabbrvfmt{\glscategory{#2}}%
2423     \glabbrvfont{\glentryshortpl{#2}}#3%
2424   }%
2425 }
```

Sxtr@p@shortpl@

```
2426 \def\@Glsxtr@p@shortpl@#1#2[#3]{%
2427   {%
2428     \glsetabbrvfmt{\glscategory{#2}}%
2429     \glabbrvfont{\Glsentryshortpl{#2}}#3%
2430   }%
2431 }
```

Sxtr@p@shortpl@

```
2432 \def\@GLSxtr@p@shortpl@#1#2[#3]{%
2433   {%
2434     \glsetabbrvfmt{\glscategory{#2}}%
2435     \mfirstucMakeUppercase{\glabbrvfont{\glentryshortpl{#2}}#3}%
2436   }%
2437 }
```

@glxtr@p@long@

```
2438 \def\@glxtr@p@long@#1#2[#3]{\{\glentrylong{#2}#3}}
```

@Glsxtr@p@long@

```
2439 \def\@Glsxtr@p@long@#1#2[#3]{\{\Glsentrylong{#2}#3}}
```

@GLSxtr@p@long@

```
2440 \def\@GLSxtr@p@long@#1#2[#3]{%
2441   {\mfirstucMakeUppercase{\glslongfont{\glentrylong{#2}}#3}}}
```

lsxtr@p@longpl@

```
2442 \def\@glxtr@p@longpl@#1#2[#3]{\{\glentrylongpl{#2}#3}}
```

lsxtr@p@longpl@

```
2443 \def\@Glsxtr@p@longpl@#1#2[#3]{\{\glslongfont{\Glsentrylongpl{#2}}#3}}
```

```

LSxtr@p@longpl@
2444 \def\@GLSxtr@p@longpl@#1#2[#3]{%
2445   {\mfirstucMakeUppercase{\glslongfont{\glsentrylongpl{#2}}#3}}}

xtr@p@acrshort@
2446 \def\@glsxtr@p@acrshort@#1#2[#3]{\{\acronymfont{\glsentryshort{#2}}#3}}

xtr@p@acrshort@
2447 \def\@GLSxtr@p@acrshort@#1#2[#3]{\{\acronymfont{\Glsentryshort{#2}}#3}}

xtr@p@acrshort@
2448 \def\@GLSxtr@p@acrshort@#1#2[#3]{%
2449   {\mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}}}

r@p@acrshortpl@
2450 \def\@glsxtr@p@acrshortpl@#1#2[#3]{\{\acronymfont{\glsentryshortpl{#2}}#3}}

r@p@acrshortpl@
2451 \def\@GLSxtr@p@acrshortpl@#1#2[#3]{\{\acronymfont{\Glsentryshortpl{#2}}#3}}

r@p@acrshortpl@
2452 \def\@GLSxtr@p@acrshortpl@#1#2[#3]{%
2453   {\mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}}}

sxtr@p@acrlong@
2454 \def\@glsxtr@p@acrlong@#1#2[#3]{\{\glsentrylong{#2}}#3}}

sxtr@p@acrlong@
2455 \def\@GLSxtr@p@acrlong@#1#2[#3]{\{\Glsentrylong{#2}}#3}}

Sxtr@p@acrlong@
2456 \def\@GLSxtr@p@acrlong@#1#2[#3]{%
2457   {\mfirstucMakeUppercase{\glsentrylong{#2}}#3}}}

tr@p@acrlongpl@
2458 \def\@glsxtr@p@acrlongpl@#1#2[#3]{\{\glsentrylongpl{#2}}#3}}

tr@p@acrlongpl@
2459 \def\@GLSxtr@p@acrlongpl@#1#2[#3]{\{\Glsentrylongpl{#2}}#3}}

tr@p@acrlongpl@
2460 \def\@GLSxtr@p@acrlongpl@#1#2[#3]{%
2461   {\mfirstucMakeUppercase{\glsentrylongpl{#2}}#3}}}

Commands to minimise conflict.

\@glsxtrp@opt
2462 \newcommand*{\@glsxtrp@opt}{hyper=false,noindex}

```

`\glsxtrsetpopts` Used in glossary to switch hyperlinks on for the `\glsxtrp` type of commands.

```
2463 \newcommand*{\glsxtrsetpopts}[1]{%
2464   \renewcommand*{\@glsxtrp@opt}{#1}%
2465 }
```

`\glossxtrsetpopts` Used in glossary to switch hyperlinks on for the `\glsxtrp` type of commands.

```
2466 \newcommand*{\glossxtrsetpopts}{%
2467   \glsxtrsetpopts{noindex}%
2468 }
```

`\@@glsxtrp`

```
2469 \newrobustcmd*{\@@glsxtrp}[2]{%
  Add scope.
2470   {%
2471     \let\glspostlinkhook\relax
2472     \csname#1\expandafter\endcsname\expandafter[\@glsxtrp@opt]{#2}[]%
2473   }%
2474 }
```

`\@glsxtrp`

```
2475 \newrobustcmd*{\@glsxtrp}[2]{%
2476   \ifcsdef{gls#1}%
2477     {%
2478       \@glsxtrp{gls#1}{#2}%
2479     }%
2480     {%
2481       \ifcsdef{glsxtr#1}%
2482         {%
2483           \@glsxtrp{glsxtr#1}{#2}%
2484         }%
2485         {%
2486           \PackageError{glossaries-extra}{‘#1’ not recognised by
2487             \string\glsxtrp}{}%
2488         }%
2489       }%
2490 }
```

`\@Glsxtrp`

```
2491 \newrobustcmd*{\@Glsxtrp}[2]{%
2492   \ifcsdef{Gls#1}%
2493     {%
2494       \@glsxtrp{Gls#1}{#2}%
2495     }%
2496     {%
2497       \ifcsdef{Glsxtr#1}%
2498         {%
2499           \@glsxtrp{Glsxtr#1}{#2}%
2500         }%
2501     }
```

```

2501   {%
2502       \PackageError{glossaries-extra}{‘#1’ not recognised by
2503           \string\Glsxtrp}{}%
2504   }%
2505 }%
2506 }

```

\@GLSxtrp

```

2507 \newrobustcmd*{\@GLSxtrp}[2]{%
2508   \ifcsdef{GLS#1}%
2509   {%
2510       \@@glxtrp{GLS#1}{#2}%
2511   }%
2512   {%
2513       \ifcsdef{GLSxtr#1}%
2514       {%
2515           \@@glxtrp{GLSxtr#1}{#2}%
2516       }%
2517       {%
2518           \PackageError{glossaries-extra}{‘#1’ not recognised by
2519               \string\GLSxtrp}{}%
2520       }%
2521   }%
2522 }

```

\glxtr@entry@p

```

2523 \newrobustcmd*{\glxtr@headentry@p}[2]{%
2524   \gl@ifattribute{#1}{headuc}{true}%
2525   {%
2526       \mfirstucMakeUppercase{\@glx@entry@field{#1}{#2}}%
2527   }%
2528   {%
2529       \@glx@entry@field{#1}{#2}%
2530   }%
2531 }

```

\glxtrp Not robust as it needs to expand somewhat.

```

2532 \ifdef\texorpdfstring
2533 {
2534   \newcommand{\glxtrp}[2]{%
2535     \protect\NoCaseChange
2536     {%
2537         \protect\texorpdfstring
2538         {%
2539             \protect\glxtrifinmark
2540             {%
2541                 \ifcsdef{glxtrhead#1}%
2542                 {%
2543                     {\protect\csuse{glxtrhead#1}{#2}}%

```

```

2544         }%
2545         {%
2546         \glxtr@headentry@p{#2}{#1}%
2547         }%
2548     }%
2549     {%
2550     \@glxtrp{#1}{#2}%
2551     }%
2552 }%
2553 {%
2554     \protect\@gls@entry@field{#2}{#1}%
2555 }%
2556 }%
2557 }
2558 }
2559 {
2560 \newcommand{\glxtrp}[2]{%
2561     \protect\NoCaseChange
2562     {%
2563         \protect\glxtrifinmark
2564         {%
2565             \ifcsdef{glxtrhead#1}%
2566             {%
2567                 {\protect\csuse{glxtrhead#1}}%
2568             }%
2569             {%
2570                 \glxtr@headentry@p{#2}{#1}%
2571             }%
2572         }%
2573         {%
2574             \@glxtrp{#1}{#2}%
2575         }%
2576     }%
2577 }
2578 }

```

Provide short synonyms for the most common option.

`\glsps`

```
2579 \newcommand*{\glsps}{\glxtrp{short}}
```

`\glspt`

```
2580 \newcommand*{\glspt}{\glxtrp{text}}
```

`\Glsxtrp` As above but use first letter upper case (but not for the bookmarks, which can't process `\uppercase`).

```
2581 \ifdef\teorpdfstring
```

```
2582 {
```

```
2583     \newcommand{\Glsxtrp}[2]{%
```

```

2584 \protect\NoCaseChange
2585 {%
2586 \protect\texorpdfstring
2587 {%
2588 \protect\glxtrifinmark
2589 {%
2590 \ifcsdef{Glsxtrhead#1}%
2591 {%
2592 {\protect\csuse{Glsxtrhead#1}{#2}}%
2593 }%
2594 {%
2595 \protect\@Gls@entry@field{#2}{#1}%
2596 }%
2597 }%
2598 {%
2599 \@Glsxtrp{#1}{#2}%
2600 }%
2601 }%
2602 {%
2603 \protect\@gls@entry@field{#2}{#1}%
2604 }%
2605 }%
2606 }
2607 }
2608 {
2609 \newcommand{\Glsxtrp}[2]{%
2610 \protect\NoCaseChange
2611 {%
2612 \protect\glxtrifinmark
2613 {%
2614 \ifcsdef{Glsxtrhead#1}%
2615 {%
2616 {\protect\csuse{Glsxtrhead#1}}%
2617 }%
2618 {%
2619 \protect\@Gls@entry@field{#2}{#1}%
2620 }%
2621 }%
2622 {%
2623 \@Glsxtrp{#1}{#2}%
2624 }%
2625 }%
2626 }
2627 }

```

`\Glsxtrp` As above but all upper case (but not for the bookmarks, which can't process `\uppercase`).

```

2628 \ifdef\texorpdfstring
2629 {
2630 \newcommand{\GlsXtrp}[2]{%

```

```

2631 \protect\NoCaseChange
2632 {%
2633 \protect\texorpdfstring
2634 {%
2635 \protect\glstrifinmark
2636 {%
2637 \ifcsdef{GLSxtr#1}%
2638 {%
2639 {\protect\GLSxtrshort[noindex,hyper=false]{#1}[]}%
2640 }%
2641 {%
2642 \protect\mfirstucMakeUppercase
2643 {%
2644 \protect\@gls@entry@field{#2}{#1}%
2645 }%
2646 }%
2647 }%
2648 {%
2649 \@GLSxtrp{#1}{#2}%
2650 }%
2651 }%
2652 {%
2653 \protect\@gls@entry@field{#2}{#1}%
2654 }%
2655 }%
2656 }
2657 }
2658 {
2659 \newcommand{\GLSxtrp}[2]{%
2660 \protect\NoCaseChange
2661 {%
2662 \protect\glstrifinmark
2663 {%
2664 \ifcsdef{GLSxtr#1}%
2665 {%
2666 {\protect\GLSxtrshort[noindex,hyper=false]{#1}[]}%
2667 }%
2668 {%
2669 \protect\mfirstucMakeUppercase
2670 {%
2671 \protect\@gls@entry@field{#2}{#1}%
2672 }%
2673 }%
2674 }%
2675 {%
2676 \@GLSxtrp{#1}{#2}%
2677 }%
2678 }%
2679 }

```



2680 }

### 1.3.5 Entry Counting

The entry counting mechanism from glossaries is adjusted here to work with category attributes. Provide a convenient command to enable entry counting, set the entrycount attribute for given categories and redefine \gls etc to use \cgl's instead.

First adjust definitions of the unset and reset commands to provide a hook.

\@glsunset    Global unset.

```
2681 \renewcommand*{\@glsunset}[1]{%
2682   \@glsunset{#1}%
2683   \glsxtrpostunset{#1}%
2684 }%
```

glsxtrpostunset

```
2685 \newcommand*{\glsxtrpostunset}[1]{}
```

\@glslocalunset    Local unset.

```
2686 \renewcommand*{\@glslocalunset}[1]{%
2687   \@glslocalunset{#1}%
2688   \glsxtrpostlocalunset{#1}%
2689   }%
```

rpostlocalunset

```
2690 \newcommand*{\glsxtrpostlocalunset}[1]{}
```

\@glsreset    Global reset.

```
2691 \renewcommand*{\@glsreset}[1]{%
2692   \@glsreset{#1}%
2693   \glsxtrpostreset{#1}%
2694 }%
```

glsxtrpostreset

```
2695 \newcommand*{\glsxtrpostreset}[1]{}
```

\@glslocalreset    Local reset.

```
2696 \renewcommand*{\@glslocalreset}[1]{%
2697   \@glslocalreset{#1}%
2698   \glsxtrpostlocalreset{#1}%
2699   }%
```

rpostlocalreset

```
2700 \newcommand*{\glsxtrpostlocalreset}[1]{}
```

leEntryCounting    The first argument is the list of categories and the second argument is the value of the entrycount attribute.

```
2701 \newcommand*{\GlsXtrEnableEntryCounting}[2]{%
```

Enable entry counting:

```
2702 \glsenableentrycount
```

Redefine \gls etc:

```
2703 \renewcommand*{\gls}{\cglsl}%
2704 \renewcommand*{\Gls}{\cGls}%
2705 \renewcommand*{\glspl}{\cglspl}%
2706 \renewcommand*{\Glspl}{\cGlspl}%
2707 \renewcommand*{\GLS}{\cGLS}%
2708 \renewcommand*{\GLSpl}{\cGLSpl}%
```

Set the entrycount attribute:

```
2709 \@glsxtr@setentrycountunsetattr{#1}{#2}%
```

In case this command is used again:

```
2710 \let\GlsXtrEnableEntryCounting\@glsxtr@setentrycountunsetattr
2711 \renewcommand*{\GlsXtrEnableEntryUnitCounting}[3]{%
2712   \PackageError{glossaries-extra}{\string\GlsXtrEnableEntryUnitCounting\space
2713     can't be used with \string\GlsXtrEnableEntryCounting}%
2714   {Use one or other but not both commands}}%
2715 }
```

ycountunsetattr

```
2716 \newcommand*{\@glsxtr@setentrycountunsetattr}[2]{%
2717   \@for\@glsxtr@cat:=#1\do
2718   {%
2719     \ifdefempty{\@glsxtr@cat}{}%
2720     {%
2721       \glssetcategoryattribute{\@glsxtr@cat}{entrycount}{#2}%
2722     }%
2723   }%
2724 }
```

Redefine the entry counting commands to take into account the entrycount attribute.

nableentrycount

```
2725 \renewcommand*{\glsenableentrycount}{%
```

Enable new fields:

```
2726 \appto\@newglossaryentry@defcounters{\@newglossaryentry@defcounters}%
```

Just in case the user has switched on the docdef option.

```
2727 \renewcommand*{\gls@defdocnewglossaryentry}{%
2728   \renewcommand*{\newglossaryentry}[2]{%
2729     \PackageError{glossaries}{\string\newglossaryentry\space
2730       may only be used in the preamble when entry counting has
2731       been activated}{If you use \string\glsenableentrycount\space
2732       you must place all entry definitions in the preamble not in
2733       the document environment}%
2734   }%
2735 }
```

New commands to access new fields:

```

2736 \newcommand*{\glsentrycurrcount}[1]{%
2737   \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
2738   {0}{\@gls@entry@field{##1}{currcount}}%
2739 }%
2740 \newcommand*{\glsentryprevcount}[1]{%
2741   \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
2742   {0}{\@gls@entry@field{##1}{prevcount}}%
2743 }%

```

Adjust post unset and reset:

```

2744 \let\@glsxtr@entrycount@org@unset\glsxtrpostunset
2745 \renewcommand*{\glsxtrpostunset}[1]{%
2746   \@glsxtr@entrycount@org@unset{##1}%
2747   \@gls@increment@currcount{##1}%
2748 }%
2749 \let\@glsxtr@entrycount@org@localunset\glsxtrpostlocalunset
2750 \renewcommand*{\glsxtrpostlocalunset}[1]{%
2751   \@glsxtr@entrycount@org@localunset{##1}%
2752   \@gls@local@increment@currcount{##1}%
2753 }%
2754 \let\@glsxtr@entrycount@org@reset\glsxtrpostreset
2755 \renewcommand*{\glsxtrpostreset}[1]{%
2756   \@glsxtr@entrycount@org@reset{##1}%
2757   \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
2758 }%
2759 \let\@glsxtr@entrycount@org@localreset\glsxtrpostlocalreset
2760 \renewcommand*{\glsxtrpostlocalreset}[1]{%
2761   \@glsxtr@entrycount@org@localreset{##1}%
2762   \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
2763 }%

```

Modifications to take into account the attributes that govern whether the entry should be unset.

```

2764 \let\@cgl@s@\@cgl@s@
2765 \let\@cgl spl@\@cgl spl@

2766 \let\@cGls@\@cGls@
2767 \let\@cGlspl@\@cGlspl@
2768 \let\@cGLS@\@cGLS@
2769 \let\@cGLSpl@\@cGLSpl@

```

The rest is as the original definition.

```

2770 \AtEndDocument{\@gls@write@entrycounts}%
2771 \renewcommand*{\@gls@entry@count}[2]{%
2772   \csgdef{glo@\glsdetoklabel{##1}@prevcount}{##2}%
2773 }%
2774 \let\glsenableentrycount\relax
2775 \renewcommand*{\glsenableentryunitcount}{%
2776   \PackageError{glossaries-extra}{\string\glsenableentryunitcount\space

```

```

2777     can't be used with \string\glsenableentrycount}%
2778     {Use one or other but not both commands}%
2779 }%
2780 }

```

`ite@entrycounts` Modify this command so that it only writes the information for entries with the entrycount attribute and issue warning if no entries have this attribute set.

```

2781 \newcommand*{\@gls@write@entrycounts}{%
2782   \immediate\write\@auxout
2783   {\string\providecommand*\string\@gls@entry@count}[2]{}%
2784   \count@=0\relax
2785   \forallglsentries{\@glsentry}{%
2786     \glshasattribute{\@glsentry}{entrycount}%
2787     {%
2788       \ifglsused{\@glsentry}%
2789       {%
2790         \immediate\write\@auxout
2791         {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
2792       }%
2793     }%
2794     \advance\count@ by \@ne
2795   }%
2796 }%
2797 }%
2798 \ifnum\count@=0
2799   \GlossariesExtraWarningNoLine{Entry counting has been enabled
2800   \MessageBreak with \string\glsenableentrycount\space but the
2801   \MessageBreak attribute 'entrycount' hasn't
2802   \MessageBreak been assigned to any of the defined
2803   \MessageBreak entries}%
2804 \fi
2805 }

```

`trifcounttrigger` `\glstrifcounttrigger{<label>}{<trigger format>}{<normal>}`

```

2806 \newcommand*{\glstrifcounttrigger}[3]{%
2807   \glshasattribute{#1}{entrycount}%
2808   {%
2809     \ifnum\glsentryprevcount{#1}>\glsgetattribute{#1}{entrycount}\relax
2810     #3%
2811   \else
2812     #2%
2813   \fi
2814 }%
2815 {#3}%
2816 }

```

Actual internal definitions of \cgl's used when entry counting is enabled.

\@@cgl's@

```

2817 \def\@@cgl's@#1#2[#3]{%
2818   \gl'sxtrifcounttrigger{#2}%
2819   {%
2820     \cgl'sformat{#2}{#3}%
2821     \gl'sunset{#2}%
2822   }%
2823   {%
2824     \@gl's@{#1}{#2}[#3]%
2825   }%
2826 }%
```

\@@cgl'spl@

```

2827 \def\@@cgl'spl@#1#2[#3]{%
2828   \gl'sxtrifcounttrigger{#2}%
2829   {%
2830     \cgl'splformat{#2}{#3}%
2831     \gl'sunset{#2}%
2832   }%
2833   {%
2834     \@gl'spl@{#1}{#2}[#3]%
2835   }%
2836 }%
```

\@@cGl's@

```

2837 \def\@@cGl's@#1#2[#3]{%
2838   \gl'sxtrifcounttrigger{#2}%
2839   {%
2840     \cGl'sformat{#2}{#3}%
2841     \gl'sunset{#2}%
2842   }%
2843   {%
2844     \@Gl's@{#1}{#2}[#3]%
2845   }%
2846 }%
```

\@@cGl'spl@

```

2847 \def\@@cGl'spl@#1#2[#3]{%
2848   \gl'sxtrifcounttrigger{#2}%
2849   {%
2850     \cGl'splformat{#2}{#3}%
2851     \gl'sunset{#2}%
2852   }%
2853   {%
2854     \@Gl'spl@{#1}{#2}[#3]%
2855   }%
2856 }%
```

\@@cGLS@

```
2857 \def\@@cGLS@#1#2[#3]{%
2858   \glxtrifcounttrigger{#2}%
2859   {%
2860     \cGLSformat{#2}{#3}%
2861     \glset{#2}%
2862   }%
2863   {%
2864     \@GLS@{#1}{#2}[#3]%
2865   }%
2866 }%
```

\@@cGLSpl@

```
2867 \def\@@cGLSpl@#1#2[#3]{%
2868   \glxtrifcounttrigger{#2}%
2869   {%
2870     \cGLSplformat{#2}{#3}%
2871     \glset{#2}%
2872   }%
2873   {%
2874     \@GLSpl@{#1}{#2}[#3]%
2875   }%
2876 }%
```

Remove default warnings from \cgl's etc so that it can be used interchangeable with \gl's etc.

\@cgl's@

```
2877 \def\@cgl's@#1#2[#3]{\@gl's@{#1}{#2}[#3]}
```

\@cGL's@

```
2878 \def\@cGL's@#1#2[#3]{\@GL's@{#1}{#2}[#3]}
```

\@cgl'spl@

```
2879 \def\@cgl'spl@#1#2[#3]{\@gl'spl@{#1}{#2}[#3]}
```

\@cGL'spl@

```
2880 \def\@cGL'spl@#1#2[#3]{\@GL'spl@{#1}{#2}[#3]}
```

Add all upper case versions not provided by glossaries.

\cGLS

```
2881 \newrobustcmd*{\cGLS}{\@gl'shyp@opt\@cGLS}
```

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

```
2882 \newcommand*{\@cGLS}[2][ ]{%
2883   \new@ifnextchar[{\@cGLS@{#1}{#2}}{\@cGLS@{#1}{#2}[ ]}%
2884 }
```

```

\@cGLS@
2885 \def\@cGLS@#1#2[#3]{\@GLS@{#1}{#2}[#3]}

\cGLSformat  Format used by \cGLS if entry only used once on previous run. The first argument is the label,
              the second argument is the insert text.
2886 \newcommand*\cGLSformat[2]{%
2887   \expandafter\mfirstuc\MakeUppercase\expandafter{\cGLSformat{#1}{#2}}%
2888 }

\cGLSpl
2889 \newrobustcmd*\cGLSpl{\@gls@hyp@opt\@cGLSpl}

\@cGLSpl  Defined the un-starred form. Need to determine if there is a final optional argument
2890 \newcommand*\@cGLSpl[2][{}]{%
2891   \new@ifnextchar[\@cGLSpl@{#1}{#2}]{\@cGLSpl@{#1}{#2}[{}]}%
2892 }

\@cGLSpl@
2893 \def\@cGLSpl@#1#2[#3]{\@GLSpl@{#1}{#2}[#3]}

\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.
2894 \newcommand*\cGLSplformat[2]{%
2895   \expandafter\mfirstuc\MakeUppercase\expandafter{\cGLSplformat{#1}{#2}}%
2896 }

      Modify the trigger formats to check for the regular attribute.

\cglformat
2897 \renewcommand*\cglformat[2]{%
2898   \glsifregular{#1}
2899   {\glsentryfirst{#1}}%
2900   {\ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}}%#2%
2901 }

\cGlsformat
2902 \renewcommand*\cGlsformat[2]{%
2903   \glsifregular{#1}
2904   {\Glsentryfirst{#1}}%
2905   {\ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}}%#2%
2906 }

\cglsplformat
2907 \renewcommand*\cglsplformat[2]{%
2908   \glsifregular{#1}
2909   {\glsentryfirstplural{#1}}%
2910   {\ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}}%#2%
2911 }

```

\cGlsplformat

```
2912 \renewcommand*{\cGlsplformat}[2]{%
2913   \glsifregular{#1}
2914   {\Glsentryfirstplural{#1}}%
2915   {\ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2}%
2916 }
```

New code similar to above for unit counting.

defunitcounters

```
2917 \newcommand*{\@@newglossaryentry@defunitcounters}{%
2918   \edef\@glo@countunit{\csuse{\@glsxtr@categoryattr@@\@glo@category @unitcount}}%
2919   \ifdefined\@glo@countunit
2920     {}%
2921     {%
2922       \@glsxtr@ifunitcounter{\@glo@countunit}%
2923       {}%
2924       {\expandafter\@glsxtr@addunitcounter\expandafter{\@glo@countunit}}%
2925     }%
2926 }
```

r@unitcountlist List to keep track of which counters are being used by the entry unit count facility.

```
2927 \newcommand*{\@glsxtr@unitcountlist}{}%
```

@addunitcounter

```
2928 \newcommand*{\@glsxtr@addunitcounter}[1]{%
2929   \listadd{\@glsxtr@unitcountlist}{#1}%
2930   \ifcsundef{glsxtr@theunit@#1}
2931     {%
2932       \ifcsdef{theH#1}%
2933       {\csdef{glsxtr@theunit@#1}{\csuse{theH#1}}}%
2934       {\csdef{glsxtr@theunit@#1}{\csuse{the#1}}}%
2935     }%
2936   {}%
2937 }
```

r@ifunitcounter

```
2938 \newcommand*{\@glsxtr@ifunitcounter}[3]{%
2939   \xifinlist{#1}{\@glsxtr@unitcountlist}{#2}{#3}%
2940 }
```

urrentunitcount

```
2941 \newcommand*{\@glsxtr@currentunitcount}[1]{%
2942   glo@\glsdetoklabel{#1}@currunit@\glsgetattribute{#1}{unitcount}.%
2943   \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
2944 }
```



previousunitcount

```
2945 \newcommand*\@glxtr@previousunitcount[1]{%
2946   glo@\glstdetoklabel{#1}@prevunit@\glsggetattribute{#1}{unitcount}.%
2947   \csuse{glxtr@theunit@\glsggetattribute{#1}{unitcount}}}%
2948 }
```

t@currunitcount

```
2949 \newcommand*\@gl@increment@currunitcount[1]{%
2950   \glshasattribute{#1}{unitcount}%
2951   {%
2952     \edef\@glxtr@csname{\@glxtr@currentunitcount{#1}}%
2953     \ifcsundef{\@glxtr@csname}%
2954     {%
2955       \csgdef{\@glxtr@csname}{1}%
2956       \listcsxadd
2957         {glo@\glstdetoklabel{#1}@unitlist}%
2958         {\glsggetattribute{#1}{unitcount}.%
2959         \csuse{glxtr@theunit@\glsggetattribute{#1}{unitcount}}}%
2960       }%
2961     }%
2962     {%
2963       \csxdef{\@glxtr@csname}%
2964       {\number\numexpr\csname\@glxtr@csname\endcsname+1}%
2965     }%
2966   }%
2967   {}%
2968 }
```

t@currunitcount

```
2969 \newcommand*\@gl@local@increment@currunitcount[1]{%
2970   \glshasattribute{#1}{unitcount}%
2971   {%
2972     \edef\@glxtr@csname{\@glxtr@currentunitcount{#1}}%
2973     \ifcsundef{\@glxtr@csname}%
2974     {%
2975       \csdef{\@glxtr@csname}{1}%
2976       \listcseadd
2977         {glo@\glstdetoklabel{#1}@unitlist}%
2978         {\glsggetattribute{#1}{unitcount}.%
2979         \csuse{glxtr@theunit@\glsggetattribute{#1}{unitcount}}}%
2980       }%
2981     }%
2982     {%
2983       \csedef{\@glxtr@csname}%
2984       {\number\numexpr\csname\@glxtr@csname\endcsname+1}%
2985     }%
2986   }%
2987   {}%
2988 }
```

r@currunitcount

```
2989 \newcommand*{\@glxstr@currunitcount}[2]{%
2990   \ifcsundef
2991     {glo@\glstetoklabel{#1}@currunit@#2}%
2992     {0}%
2993     {\csuse{glo@\glstetoklabel{#1}@currunit@#2}}%
2994 }%
```

r@prevunitcount

```
2995 \newcommand*{\@glxstr@prevunitcount}[2]{%
2996   \ifcsundef
2997     {glo@\glstetoklabel{#1}@prevunit@#2}%
2998     {0}%
2999     {\csuse{glo@\glstetoklabel{#1}@prevunit@#2}}%
3000 }%
```

entryunitcount

```
3001 \newcommand*{\glsenableentryunitcount}{%
  Enable new fields:
3002   \appto\@newglossaryentry@defcounters{\@newglossaryentry@defunitcounters}%
  Just in case the user has switched on the docdef option.
3003   \renewcommand*{\glstetoknewglossaryentry}{%
3004     \renewcommand*\newglossaryentry[2]{%
3005       \PackageError{glossaries}{\string\newglossaryentry\space
3006         may only be used in the preamble when entry counting has
3007         been activated}{If you use \string\glsenableentryunitcount\space
3008         you must place all entry definitions in the preamble not in
3009         the document environment}%
3010     }%
3011   }%
  New commands to access new fields:
3012   \newcommand*{\glstentrycurrcount}[1]{%
3013     \@glxstr@currunitcount{##1}{\glstgetattribute{##1}{unitcount}}.%
3014     \csuse{glstetoklabel{##1}@currunit@#2}%
3015   }%
3016   \newcommand*{\glstentryprevcount}[1]{%
3017     \@glxstr@prevunitcount{##1}{\glstgetattribute{##1}{unitcount}}.%
3018     \csuse{glstetoklabel{##1}@prevunit@#2}%
3019   }%
  Access total count:
3020   \newcommand*{\glstentryprevtotalcount}[1]{%
3021     \ifcsundef{glo@\glstetoklabel{##1}@prevunittotal}%
3022     {0}%
3023     {%
3024       \number\csuse{glo@\glstetoklabel{##1}@prevunittotal}
3025     }%
3026   }%
```

Access max value:

```

3027 \newcommand*{\glstentryprevmaxcount}[1]{%
3028   \ifcsundef{glo@\glstetoklabel{##1}@prevunitmax}%
3029     {0}%
3030     {%
3031       \number\csuse{glo@\glstetoklabel{##1}@prevunitmax}
3032     }%
3033 }%

```

Adjust post unset and reset:

```

3034 \let\@glstxtr@entryunitcount@org@unset\glstxtrpostunset
3035 \renewcommand*{\glstxtrpostunset}[1]{%
3036   \@glstxtr@entryunitcount@org@unset{##1}%
3037   \@glst@increment@currunitcount{##1}%
3038 }%
3039 \let\@glstxtr@entryunitcount@org@localunset\glstxtrpostlocalunset
3040 \renewcommand*{\glstxtrpostlocalunset}[1]{%
3041   \@glstxtr@entryunitcount@org@localunset{##1}%
3042   \@glst@local@increment@currunitcount{##1}%
3043 }%
3044 \let\@glstxtr@entryunitcount@org@reset\glstxtrpostreset
3045 \renewcommand*{\glstxtrpostreset}[1]{%
3046   \glshasattribute{##1}{unitcount}%
3047   {%
3048     \edef\@glstxtr@csname{\@glstxtr@currentunitcount{##1}}%
3049     \ifcsundef{\@glstxtr@csname}%
3050       {}%
3051       {\csgdef{\@glstxtr@csname}{0}}%
3052     }%
3053   }%
3054 }%
3055 \let\@glstxtr@entryunitcount@org@localreset\glstxtrpostlocalreset
3056 \renewcommand*{\glstxtrpostlocalreset}[1]{%
3057   \@glstxtr@entryunitcount@org@localreset{##1}%
3058   \glshasattribute{##1}{unitcount}%
3059   {%
3060     \edef\@glstxtr@csname{\@glstxtr@currentunitcount{##1}}%
3061     \ifcsundef{\@glstxtr@csname}%
3062       {}%
3063       {\csdef{\@glstxtr@csname}{0}}%
3064     }%
3065   }%
3066 }%

```

Modifications to take into account the attributes that govern whether the entry should be unset.

```

3067 \let\@cglst@\@cglst@
3068 \let\@cglstpl@\@cglstpl@
3069 \let\@cGls@\@cGls@

```

```

3070 \let\@cGlspl@\@cGlspl@
3071 \let\@cGLS@\@cGLS@
3072 \let\@cGLSpl@\@cGLSpl@

```

Write information to the aux file.

```

3073 \AtEndDocument{\@gls@write@entryunitcounts}%
3074 \renewcommand*{\@gls@entry@unitcount}[3]{%
3075   \csgdef{glo@\glsdetoklabel{##1}@prevunit@##3}{##2}%
3076   \ifcsundef{glo@\glsdetoklabel{##1}@prevunittotal}%
3077   {\csgdef{glo@\glsdetoklabel{##1}@prevunittotal}{##2}}%
3078   {%
3079     \csxdef{glo@\glsdetoklabel{##1}@prevunittotal}{
3080       \number\numexpr\csuse{glo@\glsdetoklabel{##1}@prevunittotal}+##2}%
3081     }%
3082     \ifcsundef{glo@\glsdetoklabel{##1}@prevunitmax}%
3083     {\csgdef{glo@\glsdetoklabel{##1}@prevunitmax}{##2}}%
3084     {%
3085       \ifnum\csuse{glo@\glsdetoklabel{##1}@prevunitmax}<##2
3086       \csgdef{glo@\glsdetoklabel{##1}@prevunitmax}{##2}%
3087       \fi
3088     }%
3089   }%
3090 \let\glsenableentryunitcount\relax
3091 \renewcommand*{\glsenableentrycount}{%
3092   \PackageError{glossaries-extra}{\string\glsenableentrycount\space
3093     can't be used with \string\glsenableentryunitcount}%
3094   {Use one or other but not both commands}%
3095 }%
3096 }
3097 \@onlypreamble\glsenableentryunitcount

```

entry@unitcount

```

3098 \newcommand*{\@gls@entry@unitcount}[3]{}

```

ryunitcounts@do

```

3099 \newcommand*{\@gls@write@entryunitcounts@do}[1]{%
3100   \immediate\write\@auxout
3101   {\string\@gls@entry@unitcount
3102     {\@glsentry}%
3103     {\@glsxtr@currunitcount{\@glsentry}{#1}%
3104     }%
3105     {#1}}%
3106 }

```

entryunitcounts

```

3107 \newcommand*{\@gls@write@entryunitcounts}{%
3108   \immediate\write\@auxout
3109   {\string\providecommand*{\string\@gls@entry@unitcount}[3]{}}%
3110   \count@=0\relax

```

```

3111 \foralllglsentries{\@glsentry}{%
3112   \glshasattribute{\@glsentry}{unitcount}%
3113   {%
3114     \ifglused{\@glsentry}%
3115     {%
3116       \forlistcsloop
3117         {\@gls@write@entryunitcounts@do}%
3118         {glo@\glsdetoklabel{\@glsentry}@unitlist}%
3119     }%
3120   }%
3121   \advance\count@ by \@ne
3122 }%
3123 {}%
3124 }%
3125 \ifnum\count@=0
3126   \GlossariesExtraWarningNoLine{Entry counting has been enabled
3127     \MessageBreak with \string\glsenableentryunitcount\space but the
3128     \MessageBreak attribute ‘unitcount’ hasn’t
3129     \MessageBreak been assigned to any of the defined
3130     \MessageBreak entries}%
3131 \fi
3132 }

```

**tryUnitCounting** The first argument is the list of categories, the second argument is the value of the entrycount attribute and the third is the counter name.

```

3133 \newcommand*{\GlsXtrEnableEntryUnitCounting}[3]{%

```

Enable entry counting:

```

3134   \glsenableentryunitcount

```

Redefine \gls etc:

```

3135   \renewcommand*{\gls}{\cgl}%
3136   \renewcommand*{\Gls}{\cGls}%
3137   \renewcommand*{\glspl}{\cglspl}%
3138   \renewcommand*{\Glspl}{\cGlspl}%
3139   \renewcommand*{\GLS}{\cGLS}%
3140   \renewcommand*{\GLSpl}{\cGLSpl}%

```

Set the entrycount attribute:

```

3141   \@glsxtr@setentryunitcountunsetattr{#1}{#2}{#3}%

```

In case this command is used again:

```

3142   \let\GlsXtrEnableEntryUnitCounting\@glsxtr@setentryunitcountunsetattr
3143   \renewcommand*{\GlsXtrEnableEntryCounting}[2]{%
3144     \PackageError{glossaries-extra}{\string\GlsXtrEnableEntryCounting\space
3145       can’t be used with \string\GlsXtrEnableEntryUnitCounting}%
3146     {Use one or other but not both commands}}%
3147 }

```

**countunsetattr**

```

3148 \newcommand*{\@glsxtr@setentryunitcountunsetattr}[3]{%
3149   \@for\@glsxtr@cat:=#1\do
3150   {%
3151     \ifdefempty{\@glsxtr@cat}{}%
3152     {%
3153       \glssetcategoryattribute{\@glsxtr@cat}{entrycount}{#2}%
3154       \glssetcategoryattribute{\@glsxtr@cat}{unitcount}{#3}%
3155     }%
3156   }%
3157 }

```

### 1.3.6 Acronym Modifications

It's more consistent to use the abbreviation code for acronyms, but make some adjustments to allow for continued use of the glossaries package's custom acronym format. (For example, user may already have defined some acronym styles with `\newacronymstyle` which they would like to continue to use.) The original glossaries acronym code can be restored with `\RestoreAcronyms`, but adjust `\SetGenericNewAcronym` so that `\newacronym` adds the category.

`\SetGenericNewAcronym`

```

3158 \renewcommand*{\SetGenericNewAcronym}{%
3159   \let\@Gls@entryname\@Gls@acrenryname
3160   \renewcommand{\newacronym}[4][]{%
3161     \ifdefempty{\@glsacronymlists}%
3162     {%
3163       \def\@glo@type{\acronymtype}%
3164       \setkeys{glossentry}{##1}%
3165       \DeclareAcronymList{\@glo@type}%
3166     }%
3167   }%
3168   \glskeylisttok{##1}%
3169   \glslabeltok{##2}%
3170   \glsshorttok{##3}%
3171   \glslongtok{##4}%
3172   \newacronymhook
3173   \protected@edef\@do@newglossaryentry{%
3174     \noexpand\newglossaryentry{\the\glslabeltok}%
3175     {%
3176       type=\acronymtype,%
3177       name={\expandonce{\acronymentry{##2}}},%
3178       sort={\acronymssort{\the\glsshorttok}{\the\glslongtok}},%
3179       text={\the\glsshorttok},%
3180       short={\the\glsshorttok},%
3181       shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
3182       long={\the\glslongtok},%
3183       longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
3184       category=acronym,

```

```

3185         \GenericAcronymFields,%
3186         \the\glskeylisttok
3187     }%
3188 }%
3189 \do@newglossaryentry
3190 }%
3191 \renewcommand*{\acrfullfmt}[3]{%
3192     \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
3193 \renewcommand*{\Acrfullfmt}[3]{%
3194     \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
3195 \renewcommand*{\ACRfullfmt}[3]{%
3196     \glslink[##1]{##2}{%
3197         \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
3198 \renewcommand*{\acrfullplfmt}[3]{%
3199     \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
3200 \renewcommand*{\Acrfullplfmt}[3]{%
3201     \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
3202 \renewcommand*{\ACRfullplfmt}[3]{%
3203     \glslink[##1]{##2}{%
3204         \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
3205 \renewcommand*{\glstryfull}[1]{\genacrfullformat{##1}{}}}%
3206 \renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
3207 \renewcommand*{\glstryfullpl}[1]{\genplacrfullformat{##1}{}}}%
3208 \renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
3209 }

```

This will cause a problem for glossaries that contain a mixture of acronyms and abbreviations, so redefine `\newacronym` to use the new abbreviation interface.

First save the original definitions:

```

3210 \let\@glstr@org@setacronymstyle\setacronymstyle
3211 \let\@glstr@org@newacronymstyle\newacronymstyle

```

**msAbbreviations** Make acronyms use the same interface as abbreviations. Note that `\newacronymstyle` has a different implementation to `\newabbreviationstyle` so disable `\newacronymstyle` and `\setacronymstyle`.

```

3212 \newcommand*{\MakeAcronymsAbbreviations}{%
3213     \renewcommand*{\newacronym}[4][]{%
3214         \glstr@newabbreviation{type=\acronymtype,category=acronym,##1}{##2}{##3}{##4}%
3215     }%
3216 \renewcommand*{\firstacronymfont}[1]{\glstr@firstabbrvfont{##1}}%
3217 \renewcommand*{\acronymfont}[1]{\glstr@abbrvfont{##1}}%
3218 \renewcommand*{\setacronymstyle}[1]{%
3219     \PackageError{glossaries-extra}{\string\setacronymstyle{##1}}
3220     unavailable.
3221     Use \string\setabbreviationstyle\space instead.
3222     The original acronym interface can be restored with
3223     \string\RestoreAcronyms}%
3224 }%
3225 \renewcommand*{\newacronymstyle}[1]{%

```

```

3226 \GlossariesExtraWarning{New acronym style ‘##1’ won’t be
3227 available unless you restore the original acronym interface with
3228 \string\RestoreAcronyms}%
3229 \@glstr@org@newacronymstyle{##1}%
3230 }%
3231 }

```

Switch acronyms to abbreviations:

```
3232 \MakeAcronymsAbbreviations
```

**RestoreAcronyms** Restore acronyms to glossaries interface.

```

3233 \newcommand*{\RestoreAcronyms}{%
3234 \SetGenericNewAcronym
3235 \renewcommand{\firstacronymfont}[1]{\acronymfont{##1}}%
3236 \renewcommand{\acronymfont}[1]{##1}%
3237 \let\setacronymstyle\@glstr@org@setacronymstyle
3238 \let\newacronymstyle\@glstr@org@newacronymstyle

```

Need to restore the original definition of \@gls@link@checkfirsthyper but \glstrifwasfirstuse still needs setting for the benefit of the post-link hook.

```

3239 \renewcommand*\@gls@link@checkfirsthyper{%
3240 \ifglused{\glslabel}%
3241 {\let\glstrifwasfirstuse\@secondoftwo}
3242 {\let\glstrifwasfirstuse\@firstoftwo}%
3243 \@glstr@org@checkfirsthyper
3244 }
3245 \glssetcategoryattribute{acronym}{regular}{false}%
3246 \setacronymstyle{long-short}%
3247 }

```

**\glsacspace** Allow the user to customise the maximum value.

```

3248 \renewcommand*{\glsacspace}[1]{%
3249 \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{#1}})}%
3250 \ifdim\dimen@<\glsacspacemax~\else\space\fi
3251 }

```

**\glsacspacemax** Value used in the above.

```
3252 \newcommand*{\glsacspacemax}{3em}
```

### 1.3.7 Indexing and Displaying Glossaries

From time-to-time users ask if they can have one glossary sorted normally and another sorted by definition or usage. With the base glossaries package this can only be achieved with the “noidx” commands (Option 1). This is an attempt to mix and match.

First we need a list of the glossaries that require **makeindex/xindy**.

**r@reg@glosslist**

```
3253 \newcommand*{\@glstr@reg@glosslist}{}
```



Save the original definition of `\makeglossaries`:

```
3254 \let\@glstr@org@makeglossaries\makeglossaries
```

Redefine `\makeglossaries` to take an optional argument. This should be empty for the usual behaviour (all glossaries need processing with an indexing application) or a comma-separated list of glossary labels indicating those glossaries that should be processed with an indexing application.

`\makeglossaries`

```
3255 \renewcommand*{\makeglossaries}[1][]{%
3256   \ifblank{#1}%
3257   {\@glstr@org@makeglossaries}%
3258   {%
3259     \edef\@glstr@reg@glosslist{#1}%
3260     \ifundef{\glswrite}{\newwrite\glswrite}{}%
3261     \protected@write\@auxout{}{\string\providecommand
3262       \string\@glorder[1]}{}
3263     \protected@write\@auxout{}{\string\providecommand
3264       \string\@istfilename[1]}{}
3265     \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
3266     \protected@write\@auxout{}{\string\@glorder{\glorder}}
3267     \protected@write\@auxout{}{\string\glstr@makeglossaries{#1}}
3268     \write\@auxout{\string\providecommand\string\@gl@reference[3]}{}}%
```

Iterate through each supplied glossary type and activate it.

```
3269   \@for\@glo@type:=#1\do{%
3270     \ifdefempty{\@glo@type}{\@makeglossary{\@glo@type}}%
3271   }%
```

New glossaries must be created before `\makeglossaries`:

```
3272 \renewcommand*\newglossary[4][]{%
3273   \PackageError{glossaries}{New glossaries
3274     must be created before \string\makeglossaries}{You need
3275     to move \string\makeglossaries\space after all your
3276     \string\newglossary\space commands}}%
```

Any subsequent instances of this command should have no effect

```
3277 \let\@makeglossary\relax
3278 \let\makeglossary\relax
3279 \renewcommand\makeglossaries[1][]{}%
```

Disable all commands that have no effect after `\makeglossaries`

```
3280 \@disable@onlypremakeg
```

Allow see key:

```
3281 \let\gl@checkseeallowed\relax
```

Adjust `\do@seeglossary`

```
3282 \renewcommand*{\do@seeglossary}[2]{%
3283   \edef\@gl@label{\gl@detoklabel{#1}}%
3284   \edef\@gl@type{\csname glo@\@gl@label @type\endcsname}%
3285   \expandafter\DTLifinlist\expandafter{\@gl@type}{\@glstr@reg@glosslist}%
```

```

3286   {\@glsxtr@org@doseeglossary{##1}{##2}}%
3287   {%
3288     \protected@write\@auxout{}\@auxout{%
3289       \string\@gls@reference
3290       {\@gls@type}{\@gls@label}{\string\glsseeformat##2{}}}%
3291     }%
3292   }%
3293 }%

Adjust \@do@wrglossary
3294 \let\@glsxtr@do@wrglossary\@do@wrglossary
3295 \def\@do@wrglossary{%
3296   \edef\@gls@type{\csname glo@\@gls@label @type\endcsname}%
3297   \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
3298   {\@glsxtr@do@wrglossary}%
3299   {\gls@noidxglossary}%
3300 }%

Suppress warning about no \makeglossaries
3301 \let\warn@nomakeglossaries\relax
3302 \def\warn@noprintglossary{%
3303   \GlossariesWarningNoLine{No \string\printglossary\space
3304     or \string\printglossaries\space
3305     found.^^J(Remove \string\makeglossaries\space if you don't want
3306     any glossaries.)^^JThis document will not have a glossary}%
3307 }%

Only warn for glossaries not listed.
3308 \renewcommand{\@gls@noref@warn}[1]{%
3309   \edef\@gls@type{##1}%
3310   \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
3311   {%
3312     \GlossariesExtraWarning{Can't use
3313       \string\printnoidxglossary[type={\@gls@type}]
3314       when '\@gls@type' is listed in the optional argument of
3315       \string\makeglossaries}%
3316   }%
3317   {%
3318     \GlossariesWarning{Empty glossary for
3319       \string\printnoidxglossary[type={##1}].
3320       Rerun may be required (or you may have forgotten to use
3321       commands like \string\gls)}%
3322   }%
3323 }%

Adjust display number list to check for type:
3324 \renewcommand*\@glsdisplaynumberlist[1]{%
3325   \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
3326   {\@glsxtr@idx@displaynumberlist{##1}}%
3327   {\@glsxtr@noidx@displaynumberlist{##1}}%
3328 }%

```

Adjust entry list:

```

3329 \renewcommand*{\glsentrynumberlist}[1]{%
3330   \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
3331   {\@glsxtr@idx@entrynumberlist{##1}}%
3332   {\@glsxtr@noidx@entrynumberlist{##1}}%
3333 }%

```

Adjust number list loop

```

3334 \renewcommand*{\glsnumberlistloop}[2]{%
3335   \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
3336   {%
3337     \PackageError{glossaries-extra}{\string\glsnumberlistloop\space
3338       not available for glossary ‘##1’}{}%
3339   }%
3340   {\@glsxtr@noidx@numberlistloop{##1}{##2}}%
3341 }%

```

Only sanitize sort for normal indexing glossaries.

```

3342 \renewcommand*{\glsprestandardsort}[3]{%
3343   \expandafter\DTLifinlist\expandafter{##2}{\@glsxtr@reg@glosslist}%
3344   {%
3345     \glsdosanitizesort
3346   }%
3347   {%
3348     \ifglssanitizesort
3349     \@gls@noidx@sanitizesort
3350   \else
3351     \@gls@noidx@nosanitizesort
3352   \fi
3353 }%
3354 }%

```

Unlike \makenoidxglossaries we can't automatically set sanitizesort=false. All entries must be defined in the preamble.

```

3355 \renewcommand*\new@glossaryentry[2]{%
3356   \PackageError{glossaries-extra}{Glossary entries must be defined
3357     in the preamble\MessageBreak when you use the optional argument
3358     of \string\makeglossaries}{Either move your definitions to the
3359     preamble or don't use the optional argument of
3360     \string\makeglossaries}%
3361 }%

```

Only activate sort key for glossaries that aren't listed in #1 (glossary label is stored in \@glo@type but this defaults to \glsdefaulttype so some expansion is required).

```

3362 \let\@glo@assign@sortkey\@glsxtr@mixed@assign@sortkey
3363 \renewcommand*\@printgloss@setsort{%

```

Need to extract just the type value.

```

3364   \expandafter\@glsxtr@gettype\expandafter,\@glsxtr@printglossopts,%
3365   type=\glsdefaulttype,\@end@glsxtr@gettype
3366   \def\@glo@sorttype{\@glo@default@sorttype}%

```

3367 }%

Check automake setting:

```
3368 \ifglsautomake
3369   \renewcommand*{\@gls@doautomake}{%
3370     \@for\@gls@type:=\@glsxtr@reg@glosslist\do{%
3371       \ifdefempty{\@gls@type}{\@gls@automake{\@gls@type}}%
3372     }%
3373   }%
3374 \fi
```

Check the sort setting (glossaries v4.30 onwards):

```
3375 \ifdef\@glo@check@sortallowed{\@glo@check@sortallowed\makeglossaries}{}%
3376 }%
3377 }
```

The optional argument version of `\makeglossaries` needs an adjustment to `\@printglossary` to allow `\@glo@assign@sortkey` to pick up the glossary type.

`\orgprintglossary` This no longer simply saves `\@printglossary` with `\let` is actually defined to check for the existence of the title, since ignored glossaries don't have a title assigned. (bib2gls writes `\provideignoredglossary` to the `glstex` file for some settings, so the glossary might not have been defined.) (This command is also used for on-the-fly setting.)

```
3378 \newcommand{\@glsxtr@orgprintglossary}[2]{%
3379   \def\@glo@type{\glsdefaulttype}%
```

Add check here.

```
3380 \def\glossarytitle{%
3381   \ifcsdef{\glo@type\@glo@type @title}%
3382     {\csuse{\glo@type\@glo@type @title}}%
3383     {\glossaryname}}%
3384 \def\glossarytoctitle{\glossarytitle}%
3385 \let\org@glossarytitle\glossarytitle
3386 \def\@glossarystyle{%
3387   \ifx\@glossary@default@style\relax
3388     \GlossariesWarning{No default glossary style provided \MessageBreak
3389       for the glossary '\@glo@type'. \MessageBreak
3390       Using deprecated fallback. \MessageBreak
3391       To fix this set the style with \MessageBreak
3392       \string\setglossarystyle\space or use the \MessageBreak
3393       style key=value option}%
3394   \fi
3395 }%
3396 \def\gls@dotoc@title{\gls@set@toc@title{\@glo@type}}%
3397 \let\@org@glossaryentrynumbers\glossaryentrynumbers
3398 \bgroup
3399   \@printgloss@setsort
3400   \setkeys{printgloss}{#1}%
3401   \ifx\glossarytitle\org@glossarytitle
3402   \else
```

```

3403     \cslet{@glo@type@}@glo@type @title-{\glossarytitle}%
3404     \fi
3405     \let\currentglossary\@glo@type
3406     \let\org@glossaryentrynumbers\glossaryentrynumbers
3407     \let\glsnonextpages\@glsnonextpages
3408     \let\glsnextpages\@glsnextpages
3409     \let\nopostdesc\@nopostdesc
3410     \gls@dotoc@title
3411     \@glossarystyle
3412     \let\gls@org@glossaryentryfield\glossentry
3413     \let\gls@org@glossarysubentryfield\subglossentry
3414     \renewcommand{\glossentry}[1]{%
3415         \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
3416         \gls@org@glossaryentryfield{##1}%
3417     }%
3418     \renewcommand{\subglossentry}[2]{%
3419         \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
3420         \gls@org@glossarysubentryfield{##1}{##2}%
3421     }%
3422     \@gls@preglossaryhook
3423     #2%
3424 \egroup
3425 \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
3426 \global\let\warn@noprintglossary\relax
3427 }

```

`\@printglossary` Redefine.

```

3428 \renewcommand{\@printglossary}[2]{%
3429     \def\@glsxtr@printglossopts{#1}%
3430     \@glsxtr@orgprintglossary{#1}{#2}%
3431 }

```

Add a key that switches off the entry targets:

```

3432 \define@choicekey{printgloss}{target}[\val\nr]{true,false}[true]{%
3433     \ifcase\nr
3434         \let\@gls@target\glsdohypertarget
3435     \else
3436         \let\@gls@target\@secondoftwo
3437     \fi
3438 }

```

`@makeglossaries` For the benefit of `makeglossaries`

```

3439 \newcommand*{\glsxtr@makeglossaries}[1]{%

```

`@glsxtr@gettype` Get just the type.

```

3440 \def\@glsxtr@gettype#1,type=#2,#3\@end@glsxtr@gettype{%
3441     \def\@glo@type{#2}%
3442 }

```

@assign@sortkey Assign the sort key.

```
3443 \newcommand\@glstr@mixed@assign@sortkey[1]{%
3444   \edef\@glo@type{\@glo@type}%
3445   \expandafter\DTLifinlist\expandafter{\@glo@type}{\@glstr@reg@glosslist}%
3446   {%
3447     \@glo@no@assign@sortkey{#1}%
3448   }%
3449   {%
3450     \@glo@assign@sortkey{#1}%
3451   }%
3452 }%
```

Display number list for the regular version:

splaynumberlist

```
3453 \let\@glstr@idx@displaynumberlist\glsdisplaynumberlist
```

Display number list for the “noidx” version:

splaynumberlist

```
3454 \newcommand*{\@glstr@noidx@displaynumberlist}[1]{%
3455   \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
3456   \ifdef\@gls@loclist
3457   {%
3458     \def\@gls@noidx@loclist@sep{%
3459       \def\@gls@noidx@loclist@sep{%
3460         \def\@gls@noidx@loclist@sep{%
3461           \glsnumlistsep
3462         }%
3463         \def\@gls@noidx@loclist@finalsep{\glsnumlistlastsep}%
3464       }%
3465     }%
3466     \def\@gls@noidx@loclist@finalsep{}%
3467     \def\@gls@noidx@loclist@prev{}%
3468     \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
3469     \@gls@noidx@loclist@finalsep
3470     \@gls@noidx@loclist@prev
3471   }%
3472   {%
3473     \glstrundeftag
3474     \glsdoifexists{#1}%
3475     {%
3476       \GlossariesWarning{Missing location list for ‘#1’. Either
3477         a rerun is required or you haven’t referenced the entry.}%
3478     }%
3479   }%
3480 }%
3481
```

And for the number list loop:

@numberlistloop

```
3482 \newcommand*{\@glsxtr@noidx@numberlistloop}[3]{%
3483   \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
3484   \let\@gls@org@glsnoidxdisplayloc@glsnoidxdisplayloc
3485   \let\@gls@org@glsseeformat@glsseeformat
3486   \let\glsnoidxdisplayloc#2\relax
3487   \let\glsseeformat#3\relax
3488   \ifdef\@gls@loclist
3489   {%
3490     \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
3491   }%
3492   {%

3493     \glsxtrundeftag
3494     \glsdoifexists{#1}%
3495     {%
3496       \GlossariesWarning{Missing location list for ‘##1’. Either
3497         a rerun is required or you haven’t referenced the entry.}%
3498     }%
3499   }%
3500   \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
3501   \let\glsseeformat\@gls@org@glsseeformat
3502 }%
```

Same for entry number list.

entrynumberlist

```
3503 \newcommand*{\@glsxtr@noidx@entrynumberlist}[1]{%
3504   \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
3505   \ifdef\@gls@loclist
3506   {%
3507     \glsnoidxloclist{\@gls@loclist}%
3508   }%
3509   {%

3510     \glsxtrundeftag
3511     \glsdoifexists{#1}%
3512     {%
3513       \GlossariesWarning{Missing location list for ‘#1’. Either
3514         a rerun is required or you haven’t referenced the entry.}%
3515     }%
3516   }%
3517 }%
```

entrynumberlist

```
3518 \newcommand*{\@glsxtr@idx@entrynumberlist}[1]{\glsentrynumberlist{#1}}
```

x@getgrouptitle Patch.

```
3519 \renewcommand*{\@gls@noidx@getgrouptitle}[2]{%
3520   \protected@edef\@glsxtr@titlelabel{#1}%
```

```

3521 \ifdefvoid\@glsxtr@titlelabel
3522 {}%
3523 {%
3524   \protected@edef\@glsxtr@titlelabel{\csuse{glsxtr@grouptitle@#1}}%
3525 }%
3526 \ifdefvoid{\@glsxtr@titlelabel}%
3527 {%
3528   \DTLifint{#1}%
3529   {%
3530     \ifnum#1<256\relax
3531       \edef#2{\char#1\relax}%
3532     \else
3533       \edef#2{#1}%
3534     \fi
3535   }%
3536   {%
3537     \ifcsundef{#1groupname}%
3538       {\def#2{#1}}%
3539       {\letcs#2{#1groupname}}%
3540   }%
3541 }%
3542 {%
3543   \let#2\@glsxtr@titlelabel
3544 }%
3545 }

```

`gls@getgrouptitle` Save original definition of `\@gls@getgrouptitle`

```
3546 \let\glsxtr@org@getgrouptitle\@gls@getgrouptitle
```

`trgetgrouptitle` Provide a user-level command to fetch the group title. The first argument is the group label. The second argument is a control sequence in which to store the title.

```

3547 \newrobustcmd{\glsxtrgetgrouptitle}[2]{%
3548   \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
3549   \@onelevel@sanitize\@glsxtr@titlelabel
3550   \ifcsdef{\@glsxtr@titlelabel}
3551     {\letcs{#2}{\@glsxtr@titlelabel}}%
3552     {\glsxtr@org@getgrouptitle{#1}{#2}}%
3553 }
3554 \let\@gls@getgrouptitle\glsxtrgetgrouptitle

```

`trsetgrouptitle` Sets the title for the given group label.

```

3555 \newcommand{\glsxtrsetgrouptitle}[2]{%
3556   \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
3557   \@onelevel@sanitize\@glsxtr@titlelabel
3558   \csxdef{\@glsxtr@titlelabel}{#2}%
3559 }

```

`\glsnavigation` Redefine to use new user-level command.

```
3560 \renewcommand*{\glsnavigation}{%
```



```

3561 \def\@gls@between{}%
3562 \ifcsundef\@gls@hypergroup\list\@glo@type}%
3563 {%
3564   \def\@gls@list{}%
3565 }%
3566 {%
3567   \expandafter\let\expandafter\@gls@list
3568     \csname @gls@hypergroup\list\@glo@type\endcsname
3569 }%
3570 \@for\@gls@tmp:=\@gls@list\do{%
3571   \@gls@between
3572   \glsxtrgetgrouptitle{\@gls@tmp}{\@gls@grptitle}%
3573   \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
3574   \let\@gls@between\glshypernavsep
3575 }%
3576 }

```

@noidx@glossary

```

3577 \renewcommand*{\@print@noidx@glossary}{%
3578   \ifcsdef\@glsref\@glo@type}%
3579   {%
3580     \ifcsdef\@glo@sortmacro\@glo@sorttype}%
3581     {%
3582       \csuse\@glo@sortmacro\@glo@sorttype{\@glo@type}%
3583     }%
3584     {%
3585       \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
3586     }%
3587     \glossarysection[\glossarytoctitle]{\glossarytitle}%
3588     \glossarypreamble

```

Moved this command definition outside of environment in case of scoping issues (e.g. in tabular-like styles).

```

3589   \def\@gls@currentlettergroup{}%
3590   \begin{theglossary}%
3591     \glossaryheader
3592     \glsresetentrylist
3593     \forlistcsloop{\@gls@noidx@do}{\@glsref\@glo@type}%
3594     \end{theglossary}%
3595     \glossarypostamble
3596   }%
3597   {%

```

Add section header if there are actually entries defined in this glossary as the document is likely pending a re-run.

```

3598     \glsxtrifemptyglossary{\@glo@type}%
3599     {}%
3600     {\glossarysection[\glossarytoctitle]{\glossarytitle}}%
3601     \@gls@noref@warn{\@glo@type}%
3602   }%

```

3603 }

noidxdisplayloc Patch to check for range formations.

```
3604 \renewcommand*{\glsnoidxdisplayloc}[4]{%
3605   \setentrycounter[#1]{#2}%
3606   \@glsxtr@display@loc#3\empty\end@glsxtr@display@loc{#4}%
3607 }
```

xtr@display@loc Patch to check for range formations.

```
3608 \def\@glsxtr@display@loc#1#2\end@glsxtr@display@loc#3{%
3609   \ifx#1\relax
3610     \glsxtrdisplaystartloc{#2}{#3}%
3611   \else
3612     \ifx#1)\relax
3613       \glsxtrdisplayendloc{#2}{#3}%
3614     \else
3615       \glsxtrdisplaysingleloc{#1#2}{#3}%
3616     \fi
3617   \fi
3618 }
```

isplaysingleloc Single location.

```
3619 \newcommand*{\glsxtrdisplaysingleloc}[2]{%
3620   \csuse{#1}{#2}%
3621 }
```

By default the range identifiers are simply ignored. A custom list loop handler can be defined by the user to test for ranges by checking the definition of `\glsxtrlocrangefmt`.

displaystartloc Start of a location range.

```
3622 \newcommand*{\glsxtrdisplaystartloc}[2]{%
3623   \edef\glsxtrlocrangefmt{#1}%
3624   \ifx\glsxtrlocrangefmt\empty
3625     \def\glsxtrlocrangefmt{glsnumberformat}%
3626   \fi
3627   \expandafter\glsxtrdisplaysingleloc
3628   \expandafter{\glsxtrlocrangefmt}{#2}%
3629 }
```

trdisplayendloc End of a location range.

```
3630 \newcommand*{\glsxtrdisplayendloc}[2]{%
3631   \edef\@glsxtr@tmp{#1}%
3632   \ifdefempty{\@glsxtr@tmp}{\def\@glsxtr@tmp{glsnumberformat}}{}}%
3633   \ifx\glsxtrlocrangefmt\@glsxtr@tmp
3634   \else
3635     \GlossariesExtraWarning{Mismatched end location range
3636       (start=\glsxtrlocrangefmt, end=\@glsxtr@tmp)}%
3637   \fi
3638   \expandafter\glsxtrdisplayendlochook\expandafter{\@glsxtr@tmp}{#2}%

```

```

3639 \expandafter\glxtrdisplaysingleloc
3640 \expandafter{\glxtrlocrangefmt}{#2}%
3641 \def\glxtrlocrangefmt{}%
3642 }

```

splayendlochook Allow the user to hook into the end of range command.

```

3643 \newcommand*{\glxtrdisplayendlochook}[2]{%

```

sxtrlocrangefmt Current range format. Empty if not in a range.

```

3644 \newcommand*{\glxtrlocrangefmt}{%

```

ls@removespaces Redefine to allow adjustments to location hyperlink.

```

3645 \def\@gls@removespaces#1 #2\@nil{%
3646 \toks@=\expandafter{\the\toks@#1}%
3647 \ifx\#2\%
3648 \edef\x{\the\toks@}%
3649 \ifx\x\empty
3650 \else
3651 \glxtrlocationhyperlink{\glstrycounter}{\@glo@counterprefix}{\the\toks@}%
3652 \fi
3653 \else
3654 \@gls@ReturnAfterFi{%
3655 \@gls@removespaces#2\@nil
3656 }%
3657 \fi
3658 }

```

cationhyperlink

```

3659 \newcommand*{\glxtrlocationhyperlink}[3]{%
3660 \ifdefined\glxtrsupplocationurl
3661 {%
3662 \glxtrhyperlink{#1#2#3}{#3}%
3663 }%
3664 {%
3665 \hyperref{\glxtrsupplocationurl}{#1#2#3}{#3}%
3666 }%
3667 }

```

supphypernumber

```

3668 \newcommand*{\glxtrsupphypernumber}[1]{%
3669 {%
3670 \glshasattribute{\glscurrententrylabel}{externallocation}%
3671 {%
3672 \def\glxtrsupplocationurl{%
3673 \glsgetaattribute{\glscurrententrylabel}{externallocation}}%
3674 }%
3675 {%
3676 \def\glxtrsupplocationurl{%
3677 }%

```

```

3678 \glshypernumber{#1}%
3679 }%
3680 }

```

Give a bit of assistance to new users who are confused and don't know how to read transcript messages.

@print@glossary

```

3681 \renewcommand{\@print@glossary}{%
3682   \makeatletter
3683   \@input{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
3684   \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
3685   {}%
3686   {\glstrNoGlossaryWarning{\@glo@type}}%
3687   \ifglxindy
3688     \ifcsundef{\xdy@\@glo@type @language}%
3689     {%
3690       \edef\@do@auxoutstuff{%
3691         \noexpand\AtEndDocument{%
3692           \noexpand\immediate\noexpand\write\@auxout{%
3693             \string\providecommand\string\@xdylanguage[2]{}}%
3694           \noexpand\immediate\noexpand\write\@auxout{%
3695             \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
3696         }%
3697       }%
3698     }%
3699   {%
3700     \edef\@do@auxoutstuff{%
3701       \noexpand\AtEndDocument{%
3702         \noexpand\immediate\noexpand\write\@auxout{%
3703           \string\providecommand\string\@xdylanguage[2]{}}%
3704         \noexpand\immediate\noexpand\write\@auxout{%
3705           \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
3706             @language\endcsname}}%
3707       }%
3708     }%
3709   }%
3710   \@do@auxoutstuff
3711   \edef\@do@auxoutstuff{%
3712     \noexpand\AtEndDocument{%
3713       \noexpand\immediate\noexpand\write\@auxout{%
3714         \string\providecommand\string\@gls@codepage[2]{}}%
3715       \noexpand\immediate\noexpand\write\@auxout{%
3716         \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
3717     }%
3718   }%
3719   \@do@auxoutstuff
3720   \fi
3721   \renewcommand*{\@warn@nomakeglossaries}{%
3722     \GlossariesWarningNoLine{\string\makeglossaries\space

```

```

3723     hasn't been used,^^Jthe glossaries will not be updated}%
3724 }%
3725 }

```

Setup the warning text to display if the external file for the given glossary is missing.

`oGlsWarningHead` Header message.

```

3726 \newcommand{\GlsXtrNoGlsWarningHead}[2]{%
3727   This document is incomplete. The external file associated with
3728   the glossary '#1' (which should be called \texttt{#2})
3729   hasn't been created.%
3730 }

```

`rningEmptyStart` No entries have been added to the glossary.

```

3731 \newcommand{\GlsXtrNoGlsWarningEmptyStart}{%
3732   This has probably happened because there are no entries defined
3733   in this glossary.%
3734 }

```

`arningEmptyMain` The default “main” glossary is empty.

```

3735 \newcommand{\GlsXtrNoGlsWarningEmptyMain}{%
3736   If you don't want this glossary,
3737   add \texttt{nomain} to your package option list when you load
3738   \texttt{glossaries-extra.sty}. For example:%
3739 }

```

`ingEmptyNotMain` A glossary that isn't the default “main” glossary is empty.

```

3740 \newcommand{\GlsXtrNoGlsWarningEmptyNotMain}[1]{%
3741   Did you forget to use \texttt{type=#1} when you defined your
3742   entries? If you tried to load entries into this glossary with
3743   \texttt{\string\loadglsentries} did you remember to use
3744   \texttt{[#1]} as the optional argument? If you did, check that
3745   the definitions in the file you loaded all had the type set
3746   to \texttt{\string\glsdefaulttype}.%
3747 }

```

`arningCheckFile` Advisory message to check the file contents.

```

3748 \newcommand{\GlsXtrNoGlsWarningCheckFile}[1]{%
3749   Check the contents of the file \texttt{#1}. If
3750   it's empty, that means you haven't indexed any of your entries in this
3751   glossary (using commands like \texttt{\string\gls} or
3752   \texttt{\string\glsadd}) so this list can't be generated.
3753   If the file isn't empty, the document build process hasn't been
3754   completed.%
3755 }

```

`WarningAutoMake` Message when automake option has been used.

```

3756 \newcommand{\GlsXtrNoGlsWarningAutoMake}[1]{%
3757   You may need to rerun \LaTeX. If you already have, it may be that

```

```

3758 \TeX's shell escape doesn't allow you to run
3759 \ifglxindy xindy\else makeindex\fi. Check the
3760 transcript file \texttt{\jobname.log}. If the shell escape is
3761 disabled, try one of the following:
3762
3763 \begin{itemize}
3764   \item Run the external (Lua) application:
3765
3766       \texttt{makeglossaries-lite.lua \string"\jobname\string"}
3767
3768   \item Run the external (Perl) application:
3769
3770       \texttt{makeglossaries \string"\jobname\string"}
3771 \end{itemize}
3772
3773 Then rerun \LaTeX\ on this document.
3774 \GlossariesExtraWarning{Rerun required to build the
3775 glossary '#1' or check TeX's shell escape allows
3776 you to run \ifglxindy xindy\else makeindex\fi}%
3777 }

```

WarningMismatch Mismatching \makenoidxglossaries.

```

3778 \newcommand{\GlsXtrNoGlsWarningMismatch}{%
3779   You need to either replace \texttt{\string\makenoidxglossaries}
3780   with \texttt{\string\makeglossaries} or replace
3781   \texttt{\string\printglossary} (or \texttt{\string\printglossaries}) with
3782   \texttt{\string\printnoidxglossary}
3783   (or \texttt{\string\printnoidxglossaries}) and then rebuild
3784   this document.%
3785 }

```

WarningBuildInfo Build advice.

```

3786 \newcommand{\GlsXtrNoGlsWarningBuildInfo}{%
3787   Try one of the following:
3788   \begin{itemize}
3789     \item Add \texttt{automake} to your package option list when you load
3790         \texttt{glossaries-extra.sty}. For example:
3791
3792         \texttt{\string\usepackage[automake]%
3793             \glsopenbrace glossaries-extra\glsclosebrace}
3794
3795     \item Run the external (Lua) application:
3796
3797         \texttt{makeglossaries-lite.lua \string"\jobname\string"}
3798
3799     \item Run the external (Perl) application:
3800
3801         \texttt{makeglossaries \string"\jobname\string"}

```

```

3802 \end{itemize}
3803
3804 Then rerun \LaTeX\ on this document.%
3805 }

```

oGlsWarningTail Final paragraph.

```

3806 \newcommand{\GlsXtrNoGlsWarningTail}{%
3807 This message will be removed once the problem has been fixed.%
3808 }

```

GlsWarningNoOut No out file created. Build advice.

```

3809 \newcommand{\GlsXtrNoGlsWarningNoOut}[1]{%
3810 The file \texttt{#1} doesn't exist. This most likely means you haven't used
3811 \texttt{\string\makeglossaries} or you have used
3812 \texttt{\string\nofiles}. If this is just a draft version of the
3813 document, you can suppress this message using the
3814 \texttt{\nomissingglstext} package option.%
3815 }

```

glossarywarning

```

3816 \newcommand*{@\glstr@defaultnoglossarywarning}[1]{%
3817 \glossarysection[\glossarytoctitle]{\glossarytitle}
3818 \GlsXtrNoGlsWarningHead{#1}{\jobname.\csname @glotype@\@glo@type @in\endcsname}
3819 \par
3820 \glstrifemptyglossary{#1}%
3821 {%
3822 \GlsXtrNoGlsWarningEmptyStart\space
3823 \ifthenelse{\equal{#1}{main}}{\GlsXtrNoGlsWarningEmptyMain\par
3824 \medskip
3825 \noindent\texttt{\string\usepackage[nomain\ifglacronym ,acronym\fi]%
3826 \glsoopenbrace glossaries-extra\glsclosebrace}
3827 \medskip
3828 }%
3829 {\GlsXtrNoGlsWarningEmptyNotMain{#1}}%
3830 }%
3831 {%
3832 \IfFileExists{\jobname.\csname @glotype@\@glo@type @out\endcsname}
3833 {%
3834 \GlsXtrNoGlsWarningCheckFile
3835 {\jobname.\csname @glotype@\@glo@type @out\endcsname}
3836
3837 \ifglautomake
3838
3839 \GlsXtrNoGlsWarningAutoMake{#1}
3840
3841 \else
3842
3843 \ifthenelse{\equal{#1}{main}}{%
3844 {%

```

```

3845         \GlsXtrNoGlsWarningEmptyMain\par
3846         \medskip
3847         \noindent\texttt{\string\usepackage[nomain]%
3848         \glsopenbrace glossaries-extra\glsclosebrace}
3849         \medskip
3850     }%
3851     {%
3852
3853         \ifdefequal\makeglossaries\@no@makeglossaries
3854         {%
3855             \GlsXtrNoGlsWarningMisMatch
3856         }%
3857         {%
3858             \GlsXtrNoGlsWarningBuildInfo
3859         }%
3860     \fi
3861 }%
3862 {%
3863     \GlsXtrNoGlsWarningNoOut
3864     {\jobname.\csname @glotype@\@glo@type @out\endcsname}%
3865 }%
3866 }%
3867 \par
3868 \GlsXtrNoGlsWarningTail
3869 }

```

Provide some commands to accompany the record option for use with **bib2gls**.

**xttrresourcefile** Since it's dangerous for an external application to create a file with a .tex extension, as from v1.11 this enforces a .glstex extension to avoid conflict.

```

3870 \newcommand*{\glxtrresourcefile}[2] [] {%
3871     \glxtr@writefields
3872     \protected@write\@auxout{}\string\glxtr@resource{#1}{#2}}%
3873     \let\@glxtr@org@see@noindex\@gls@see@noindex
3874     \let\@gls@see@noindex\relax
3875     \IfFileExists{#2.glstex}%
3876     {%

```

Can't scope \@input so save and restore the category code of @ to allow for internal commands in the location list.

```

3877     \edef\@bibgls@restoreat{\noexpand\catcode\noexpand'\noexpand\@=\number\catcode'\@}%
3878     \makeatletter
3879     \@input{#2.glstex}%
3880     \@bibgls@restoreat
3881 }%
3882 {%
3883     \GlossariesExtraWarning{No file '#2.glstex'}%
3884 }%
3885     \let\@gls@see@noindex\@glxtr@org@see@noindex
3886 }

```



```

3887 \@onlypreamble\glxtrresourcefile

trresourcecount
3888 \newcount\glxtrresourcecount

trLoadResources  Short cut that uses \glxtrresourcefile with \jobname as the mandatory argument.
3889 \newcommand*{\GlsXtrLoadResources}[1] [] {%
3890   \ifnum\glxtrresourcecount=0\relax
3891     \glxtrresourcefile[#1]{\jobname}%
3892   \else
3893     \glxtrresourcefile[#1]{\jobname-\the\glxtrresourcecount}%
3894   \fi
3895   \advance\glxtrresourcecount by 1\relax
3896 }

glxtr@resource
3897 \newcommand*{\glxtr@resource}[2] {}

\glxtr@fields
3898 \newcommand*{\glxtr@fields}[1] {}

xtr@texencoding
3899 \newcommand*{\glxtr@texencoding}[1] {}

\glxtr@langtag
3900 \newcommand*{\glxtr@langtag}[1] {}

@pluralsuffixes
3901 \newcommand*{\glxtr@pluralsuffixes}[4] {}

tr@shortcutsval
3902 \newcommand*{\glxtr@shortcutsval}[1] {}

sxtr@linkprefix
3903 \newcommand*{\glxtr@linkprefix}[1] {}

xtr@writefields  This information only needs to be written once, so disable it after it's been used.
3904 \newcommand*{\glxtr@writefields}{%
3905   \protected@write\@auxout{%
3906     {\string\providecommand*{\string\glxtr@fields}[1]{}}%
3907   \protected@write\@auxout{%
3908     {\string\providecommand*{\string\glxtr@resource}[2]{}}%
3909   \protected@write\@auxout{%
3910     {\string\providecommand*{\string\glxtr@pluralsuffixes}[4]{}}%
3911   \protected@write\@auxout{%
3912     {\string\providecommand*{\string\glxtr@shortcutsval}[1]{}}%
3913   \protected@write\@auxout{%
3914     {\string\providecommand*{\string\glxtr@linkprefix}[1]{}}%
3915   \protected@write\@auxout{{\string\glxtr@fields{\@gls@keymap}}}%

```

If any languages have been loaded, the language tag will be available in `\CurrentTrackedLanguageTag` (provided by `tracklang`). For multilingual documents, the required locale will have to be indicated in the sort key when using `\glxtrresourcefile`.

```

3916 \ifdef\CurrentTrackedLanguageTag
3917 {%
3918     \protected@write\@auxout{}\{%
3919         \string\glxtr@langtag{\CurrentTrackedLanguageTag}}%
3920     }%
3921 }%
3922 \protected@write\@auxout{}\{\string\glxtr@pluralsuffixes
3923     {\glspluralsuffix}{\abbrvpluralsuffix}{\acrpluralsuffix}%
3924     {\glxtrabbrvpluralsuffix}}%
3925 \ifdef\inputencodingname
3926 {%
3927     \protected@write\@auxout{}\{\string\glxtr@texencoding{\inputencodingname}}%
3928     }%
3929 {%

```

If `fontspec` has been loaded, assume UTF-8. (The encoding can be changed with `\XeTeXinputencoding`, but I can't work out how to determine the current encoding.)

```

3930     \@ifpackageloaded{fontspec}%
3931     {\protected@write\@auxout{}\{\string\glxtr@texencoding{utf8}}}%
3932     }%
3933 }%
3934 \protected@write\@auxout{}\{\string\glxtr@shortcutsval{\glxtr@shortcutsval}}%

```

Prefix deferred until the beginning of the document in case it's redefined later in the preamble. This is picked up by `bib2gls` when the external option is used.

```

3935 \AtBeginDocument
3936 {\protected@write\@auxout{}\{\string\glxtr@linkprefix{\glolinkprefix}}}%
3937 \let\glxtr@writefields\relax

```

If the `automake` option is on, try running `bib2gls` if the aux file exists. The double-quotes around `\jobname` have been removed (v1.19) since `\jobname` will include double-quotes if the file name has spaces.

```

3938 \ifglautomake
3939 \IfFileExists{\jobname.aux}%
3940 {\immediate\write18{bib2gls \jobname}}}%

```

If `\makeglossaries` is also used, allow `makeindex/xindy` to also be run, otherwise disable the error message about requiring `\makeglossaries` with `automake=true`.

```

3941 \ifx\@gls@doautomake\@gls@doautomake@err
3942 \let\@gls@doautomake\relax
3943 \fi
3944 \fi
3945 }

```

`do@automake@err`

```

3946 \newcommand*{\@gls@doautomake@err}{%
3947 \PackageError{glossaries}{You must use

```

```

3948 \string\makeglossaries\space with automake=true}
3949 {%
3950     Either remove the automake=true setting or
3951     add \string\makeglossaries\space to your document preamble.%
3952 }%
3953 }

```

Allow locations specific to a particular counter to be recorded.

`\glstr@record`

```

3954 \newcommand*{\glstr@record}[5]{%

```

`r@counterrecord` Aux file command.

```

3955 \newcommand*{\glstr@counterrecord}[3]{%
3956 \glstrfieldlistgadd{#1}{record.#2}{#3}%
3957 }

```

`unterrecordhook` Hook used by `\@glstr@dorecord`.

```

3958 \newcommand*{\@glstr@counterrecordhook}{%

```

`trRecordCounter` Activate recording for a particular counter (identified in the argument).

```

3959 \newcommand*{\GlsXtrRecordCounter}[1]{%
3960 \@glstr@recordcounter{#1}%
3961 }
3962 \@onlypreamble\GlsXtrRecordCounter

```

`docounterrecord`

```

3963 \newcommand*{\@glstr@docounterrecord}[1]{%
3964 \protected@write\@auxout{}{\string\glstr@counterrecord
3965 {\@glstr@label}{#1}{\csuse{the#1}}}%
3966 }

```

`ntunsrtglossary` Similar to `\printnoidxglossary` but it displays all entries defined for the given glossary without sorting.

```

3967 \newcommand*{\printunsrtglossary}{%
3968 \@ifstar\s@printunsrtglossary\@printunsrtglossary
3969 }

```

`ntunsrtglossary` Unstarred version.

```

3970 \newcommand*{\@printunsrtglossary}[1][{}]{%
3971 \@printglossary{type=\glsdefaulttype,#1}{\@print@unsrt@glossary}%
3972 }

```

`ntunsrtglossary` Starred version.

```

3973 \newcommand*{\s@printunsrtglossary}[2][{}]{%
3974 \begingroup
3975 #2%
3976 \@printglossary{type=\glsdefaulttype,#1}{\@print@unsrt@glossary}%
3977 \endgroup
3978 }

```

`\unsrtglossaries` Similar to `\printnoidxglossaries` but it displays all entries defined for the given glossary without sorting.

```
3979 \newcommand*{\printunsrtglossaries}{%
3980   \forallglossaries{\@glo@type}{\printunsrtglossary[type=\@glo@type]}%
3981 }
```

`@unsrt@glossary`

```
3982 \newcommand*{\@print@unsrt@glossary}{%
3983   \glossarysection[\glossarytoctitle]{\glossarytitle}%
3984   \glossarypreamble

   check for empty list
3985   \glstrifemptyglossary{\@glo@type}%
3986   {%
3987     \GlossariesExtraWarning{No entries defined in glossary ‘\@glo@type’}%
3988   }%
3989   {%

3990   \key@ifundefined{glossentry}{group}%
3991   {\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle}%
3992   {\let\@gls@getgrouptitle\@glstr@unsrt@getgrouptitle}%
3993   \def\@gls@currentlettergroup{}}%
```

A loop within the tabular-like styles can cause problems, so move the loop outside.

```
3994   \def\@glstr@doglossary{%
3995     \begin{theglossary}%
3996     \glossaryheader
3997     \glsresetentrylist
3998   }%
3999   \expandafter\@for\expandafter\glscurrententrylabel\expandafter
4000   : \expandafter=\csname glo@list@\@glo@type\endcsname\do{%
4001     \ifdefempty{\glscurrententrylabel}
4002     {}%
4003     {\eappto\@glstr@doglossary{%
4004       \noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}}%
4005   }%
4006   \appto\@glstr@doglossary{\end{theglossary}}%
4007   \@glstr@doglossary
4008 }%
4009 \glossarypostamble
4010 }
```

`glossary@handler`

```
4011 \newcommand{\@printunsrt@glossary@handler}[1]{%
4012   \xdef\glscurrententrylabel{#1}%
4013   \printunsrtglossaryhandler\glscurrententrylabel
4014 }
```

`glossaryhandler`

```

4015 \newcommand{\printunsrtglossaryhandler}[1]{%
4016   \glxtrunsrtdo{#1}%
4017 }

```

srtglossaryunit

```

4018 \newcommand{\print@op@unsrtglossaryunit}[2][ ]{%
4019   \s@printunsrtglossary[type=\gldefaulttype,#1]{%
4020     \printunsrtglossaryunitsetup{#2}%
4021   }%
4022 }

```

ossaryunitsetup

```

4023 \newcommand*{\printunsrtglossaryunitsetup}[1]{%
4024   \renewcommand{\printunsrtglossaryhandler}[1]{%
4025     \glxtrfieldxifinlist{##1}{record.#1}{\csuse{the#1}}
4026     {\glxtrunsrtdo{##1}}%
4027   }%
4028 }%
4029 \ifcsundef{theH#1}%
4030 {%
4031   \renewcommand*{\glolinkprefix}{record.#1.\csuse{the#1}.}%
4032 }%
4033 {%
4034   \renewcommand*{\glolinkprefix}{record.#1.\csuse{theH#1}.}%
4035 }%
4036 \renewcommand*{\glossarysection}[2][ ]{%
4037   \appto\glossarypostamble{\glspar\medskip\glspar}%
4038 }

```

srtglossaryunit

```

4039 \newcommand{\print@noop@unsrtglossaryunit}[2][ ]{%
4040   \PackageError{glossaries-extra}{\string\printunsrtglossaryunit\space
4041     requires the record=only or record=alsoindex package option}{}%
4042 }

```

t@getgrouptitle

```

4043 \newrobustcmd*{\@glxtr@unsrt@getgrouptitle}[2]{%
4044   \protected@edef\@glxtr@titlelabel{\glxtr@grouptitle@#1}%
4045   \@onelevel@sanitize\@glxtr@titlelabel
4046   \ifcsdef{\@glxtr@titlelabel}
4047   {\letcs{#2}{\@glxtr@titlelabel}}%
4048   {\def#2{#1}}%
4049 }

```

\glxtrunsrtdo    Provide a user-level call to \@glxtr@noidx@do to make it easier to define a new handler.

```

4050 \newcommand{\glxtrunsrtdo}{\@glxtr@noidx@do}

```

\glxtr@noidx@do    Minor modification of \@glx@noidx@do to check for location field if present.

```

4051 \newcommand{\@glsxtr@noidx@do}[1]{%
4052   \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
4053   \global\letcs{\@gls@location}{glo@\glsdetoklabel{#1}@location}%
4054   \ifglshasparent{#1}%
4055   {%
4056     \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
4057     \ifdefvoid{\@gls@location}%
4058     {%
4059       \ifdefvoid{\@gls@loclist}%
4060       {%
4061         \subglossentry{\gls@level}{#1}{}%
4062       }%
4063       {%
4064         \subglossentry{\gls@level}{#1}%
4065         {%
4066           \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
4067         }%
4068       }%
4069     }%
4070   }%
4071   \subglossentry{\gls@level}{#1}{\glossaryentrynumbers{\@gls@location}}%
4072 }%
4073 }%
4074 {%
4075   \key@ifundefined{glossentry}{group}%
4076   {%
4077     \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
4078     \expandafter\glo@grabfirst\@gls@sort{}\@nil
4079   }%
4080   {%
4081     \protected@xdef\@glo@thislettergrp{%
4082       \csuse{glo@\glsdetoklabel{#1}@group}}%
4083   }%
4084   \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
4085   {}%
4086   {%
4087     \ifdefempty{\@gls@currentlettergroup}{\@gls@groupskip}%
4088     \expandafter\gls@groupheading\expandafter
4089     {\@glo@thislettergrp}%
4090   }%
4091   \global\let\@gls@currentlettergroup\@glo@thislettergrp
4092   \ifdefvoid{\@gls@location}%
4093   {%
4094     \ifdefvoid{\@gls@loclist}
4095     {%
4096       \glossentry{#1}{}%
4097     }%
4098     {%
4099       \glossentry{#1}%

```

```

4100      {%
4101      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}}%
4102      }%
4103    }%
4104  }%
4105  {%
4106    \glossentry{#1}%
4107    {%
4108      \glossaryentrynumbers{\@gls@location}%
4109    }%
4110  }%
4111 }%
4112 }

```

## 1.4 Integration with glossaries-accsupp

Provide better integration with the glossaries-accsupp package. (Must be loaded before the main code of glossaries-extra either explicitly or through the accsupp package option.)

These commands have their definitions set according to whether or not glossaries-extra has been loaded.

```

4113 \@ifpackageloaded{glossaries-accsupp}
4114 {

```

Define (or redefine) commands to use the accessibility information.

**\glsaccessname** Display the name value (no link and no check for existence).

```

4115 \newcommand*{\glsaccessname}[1]{%
4116   \glsnameaccessdisplay
4117   {%
4118     \glsentryname{#1}%
4119   }%
4120   {#1}%
4121 }

```

**\Glsaccessname** Display the name value (no link and no check for existence) with the first letter converted to upper case.

```

4122 \newcommand*{\Glsaccessname}[1]{%
4123   \glsnameaccessdisplay
4124   {%
4125     \Glsentryname{#1}%
4126   }%
4127   {#1}%
4128 }

```

**\GLSaccessname** Display the name value (no link and no check for existence) converted to upper case.

```

4129 \newcommand*{\GLSaccessname}[1]{%
4130   \glsnameaccessdisplay
4131   {%

```

```

4132     \mfirstucMakeUppercase{\glsentryname{#1}}%
4133   }%
4134   {#1}%
4135 }

```

`\glsaccesstext` Display the text value (no link and no check for existence).

```

4136 \newcommand*{\glsaccesstext}[1]{%
4137   \glstextaccessdisplay
4138   {%
4139     \glsentrytext{#1}%
4140   }%
4141   {#1}%
4142 }

```

`\Glsaccesstext` Display the text value (no link and no check for existence) with the first letter converted to upper case.

```

4143 \newcommand*{\Glsaccesstext}[1]{%
4144   \glstextaccessdisplay
4145   {%
4146     \Glsentrytext{#1}%
4147   }%
4148   {#1}%
4149 }

```

`\GLSaccesstext` Display the text value (no link and no check for existence) converted to upper case.

```

4150 \newcommand*{\GLSaccesstext}[1]{%
4151   \glstextaccessdisplay
4152   {%
4153     \mfirstucMakeUppercase{\glsentrytext{#1}}%
4154   }%
4155   {#1}%
4156 }

```

`glsaccessplural` Display the plural value (no link and no check for existence).

```

4157 \newcommand*{\glsaccessplural}[1]{%
4158   \glspluralaccessdisplay
4159   {%
4160     \glsentryplural{#1}%
4161   }%
4162   {#1}%
4163 }

```

`Glsaccessplural` Display the plural value (no link and no check for existence) with the first letter converted to upper case.

```

4164 \newcommand*{\Glsaccessplural}[1]{%
4165   \glspluralaccessdisplay
4166   {%
4167     \Glsentryplural{#1}%

```



```

4168     }%
4169     {#1}%
4170 }

```

**GLSaccessplural** Display the plural value (no link and no check for existence) converted to upper case.

```

4171 \newcommand*{\GLSaccessplural}[1]{%
4172   \glspluralaccessdisplay
4173   {%
4174     \mfirstucMakeUppercase{\glsentryplural{#1}}%
4175   }%
4176   {#1}%
4177 }

```

**\glsaccessfirst** Display the first value (no link and no check for existence).

```

4178 \newcommand*{\glsaccessfirst}[1]{%
4179   \glsfirstaccessdisplay
4180   {%
4181     \glsentryfirst{#1}%
4182   }%
4183   {#1}%
4184 }

```

**\Glsaccessfirst** Display the first value (no link and no check for existence) with the first letter converted to upper case.

```

4185 \newcommand*{\Glsaccessfirst}[1]{%
4186   \glsfirstaccessdisplay
4187   {%
4188     \Glsentryfirst{#1}%
4189   }%
4190   {#1}%
4191 }

```

**\GLSaccessfirst** Display the first value (no link and no check for existence) converted to upper case.

```

4192 \newcommand*{\GLSaccessfirst}[1]{%
4193   \glsfirstaccessdisplay
4194   {%
4195     \mfirstucMakeUppercase{\glsentryfirst{#1}}%
4196   }%
4197   {#1}%
4198 }

```

**cessfirstplural** Display the firstplural value (no link and no check for existence).

```

4199 \newcommand*{\glsaccessfirstplural}[1]{%
4200   \glsfirstpluralaccessdisplay
4201   {%
4202     \glsentryfirstplural{#1}%
4203   }%
4204   {#1}%
4205 }

```

cessfirstplural Display the firstplural value (no link and no check for existence) with the first letter converted to upper case.

```

4206 \newcommand*\Glsaccessfirstplural}[1]{%
4207   \glsfirstpluralaccessdisplay
4208   {%
4209     \Glsentryfirstplural{#1}%
4210   }%
4211   {#1}%
4212 }
```

cessfirstplural Display the firstplural value (no link and no check for existence) converted to upper case.

```

4213 \newcommand*\GLSaccessfirstplural}[1]{%
4214   \glsfirstpluralaccessdisplay
4215   {%
4216     \mfirstucMakeUppercase{\glsentryfirstplural{#1}}%
4217   }%
4218   {#1}%
4219 }
```

glsaccesssymbol Display the symbol value (no link and no check for existence).

```

4220 \newcommand*\glsaccesssymbol}[1]{%
4221   \glssymbolaccessdisplay
4222   {%
4223     \glsentrysymbol{#1}%
4224   }%
4225   {#1}%
4226 }
```

GLSaccesssymbol Display the symbol value (no link and no check for existence) with the first letter converted to upper case.

```

4227 \newcommand*\GLSaccesssymbol}[1]{%
4228   \glssymbolaccessdisplay
4229   {%
4230     \Glsentrysymbol{#1}%
4231   }%
4232   {#1}%
4233 }
```

GLSaccesssymbol Display the symbol value (no link and no check for existence) converted to upper case.

```

4234 \newcommand*\GLSaccesssymbol}[1]{%
4235   \glssymbolaccessdisplay
4236   {%
4237     \mfirstucMakeUppercase{\glsentrysymbol{#1}}%
4238   }%
4239   {#1}%
4240 }
```

esssymbolplural Display the symbolplural value (no link and no check for existence).

```

4241 \newcommand*{\glsaccesssymbolplural}[1]{%
4242   \glssymbolpluralaccessdisplay
4243   {%
4244     \glsentrysymbolplural{#1}%
4245   }%
4246   {#1}%
4247 }

```

`\esssymbolplural` Display the symbolplural value (no link and no check for existence) with the first letter converted to upper case.

```

4248 \newcommand*{\Glsaccesssymbolplural}[1]{%
4249   \glssymbolpluralaccessdisplay
4250   {%
4251     \Glsentrysymbolplural{#1}%
4252   }%
4253   {#1}%
4254 }

```

`\esssymbolplural` Display the symbolplural value (no link and no check for existence) converted to upper case.

```

4255 \newcommand*{\GLSaccesssymbolplural}[1]{%
4256   \glssymbolpluralaccessdisplay
4257   {%
4258     \mfirstucMakeUppercase{\glsentrysymbolplural{#1}}%
4259   }%
4260   {#1}%
4261 }

```

`\glsaccessdesc` Display the desc value (no link and no check for existence).

```

4262 \newcommand*{\glsaccessdesc}[1]{%
4263   \glsdescriptionaccessdisplay
4264   {%
4265     \glsentrydesc{#1}%
4266   }%
4267   {#1}%
4268 }

```

`\Glsaccessdesc` Display the desc value (no link and no check for existence) with the first letter converted to upper case.

```

4269 \newcommand*{\Glsaccessdesc}[1]{%
4270   \glsdescriptionaccessdisplay
4271   {%
4272     \Glsentrydesc{#1}%
4273   }%
4274   {#1}%
4275 }

```

`\GLSaccessdesc` Display the desc value (no link and no check for existence) converted to upper case.

```

4276 \newcommand*{\GLSaccessdesc}[1]{%

```

```

4277 \glsdescriptionaccessdisplay
4278 {%
4279 \mfirstucMakeUppercase{\glsentrydesc{#1}}%
4280 }%
4281 {#1}%
4282 }

```

`\glsaccessdescplural` Display the descplural value (no link and no check for existence).

```

4283 \newcommand*{\glsaccessdescplural}[1]{%
4284 \glsdescriptionpluralaccessdisplay
4285 {%
4286 \glsentrydescplural{#1}%
4287 }%
4288 {#1}%
4289 }

```

`\Glsaccessdescplural` Display the descplural value (no link and no check for existence) with the first letter converted to upper case.

```

4290 \newcommand*{\Glsaccessdescplural}[1]{%
4291 \glsdescriptionpluralaccessdisplay
4292 {%
4293 \Glsentrydescplural{#1}%
4294 }%
4295 {#1}%
4296 }

```

`\glsaccessdescplural` Display the descplural value (no link and no check for existence) converted to upper case.

```

4297 \newcommand*{\GLSaccessdescplural}[1]{%
4298 \glsdescriptionpluralaccessdisplay
4299 {%
4300 \mfirstucMakeUppercase{\glsentrydescplural{#1}}%
4301 }%
4302 {#1}%
4303 }

```

`\glsaccesssshort` Display the short form (no link and no check for existence).

```

4304 \newcommand*{\glsaccesssshort}[1]{%
4305 \glsshortaccessdisplay
4306 {%
4307 \glsentryshort{#1}%
4308 }%
4309 {#1}%
4310 }

```

`\Glsaccesssshort` Display the short form with first letter converted to uppercase (no link and no check for existence).

```

4311 \newcommand*{\Glsaccesssshort}[1]{%
4312 \glsshortaccessdisplay

```

```

4313     {%
4314         \Glsentryshort{#1}%
4315     }%
4316     {#1}%
4317 }

```

`\GLSaccessshort` Display the short value (no link and no check for existence) converted to upper case.

```

4318 \newcommand*{\GLSaccessshort}[1]{%
4319     \glsshortaccessdisplay
4320     {%
4321         \mfirstucMakeUppercase{\Glsentryshort{#1}}%
4322     }%
4323     {#1}%
4324 }

```

`lsaccessshortpl` Display the short plural form (no link and no check for existence).

```

4325 \newcommand*{\lsaccessshortpl}[1]{%
4326     \glsshortpluralaccessdisplay
4327     {%
4328         \Glsentryshortpl{#1}%
4329     }%
4330     {#1}%
4331 }

```

`lSaccessshortpl` Display the short plural form with first letter converted to uppercase (no link and no check for existence).

```

4332 \newcommand*{\lSaccessshortpl}[1]{%
4333     \glsshortpluralaccessdisplay
4334     {%
4335         \Glsentryshortpl{#1}%
4336     }%
4337     {#1}%
4338 }

```

`LSaccessshortpl` Display the shortplural value (no link and no check for existence) converted to upper case.

```

4339 \newcommand*{\LSaccessshortpl}[1]{%
4340     \glsshortpluralaccessdisplay
4341     {%
4342         \mfirstucMakeUppercase{\Glsentryshortpl{#1}}%
4343     }%
4344     {#1}%
4345 }

```

`\glsaccesslong` Display the long form (no link and no check for existence).

```

4346 \newcommand*{\glsaccesslong}[1]{%
4347     \glslongaccessdisplay{\Glsentrylong{#1}}{#1}%
4348 }

```

```

\GLsaccesslong   Display the long form (no link and no check for existence).
4349
4350   \newcommand*{\GLsaccesslong}[1]{%
4351     \glslongaccessdisplay{\Glsentrylong{#1}}{#1}%
4352   }

\GLSaccesslong   Display the long value (no link and no check for existence) converted to upper case.
4353   \newcommand*{\GLSaccesslong}[1]{%
4354     \glslongaccessdisplay
4355     {%
4356       \mfirstucMakeUppercase{\Glsentrylong{#1}}%
4357     }%
4358     {#1}%
4359   }

glsaccesslongpl   Display the long plural form (no link and no check for existence).
4360   \newcommand*{\glsaccesslongpl}[1]{%
4361     \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}%
4362   }

GLsaccesslongpl   Display the long plural form (no link and no check for existence).
4363
4364   \newcommand*{\GLsaccesslongpl}[1]{%
4365     \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}%
4366   }

GLSaccesslongpl   Display the longplural value (no link and no check for existence) converted to upper case.
4367   \newcommand*{\GLSaccesslongpl}[1]{%
4368     \glslongpluralaccessdisplay
4369     {%
4370       \mfirstucMakeUppercase{\glsentrylongpl{#1}}%
4371     }%
4372     {#1}%
4373   }

      End of if part
4374 }
4375 {

      No accessibility support. Just define these commands to do \glsentry<xxx>

\glsaccessname   Display the name value (no link and no check for existence).
4376   \newcommand*{\glsaccessname}[1]{\glsentryname{#1}}

\GLsaccessname   Display the name value (no link and no check for existence) with the first letter converted to
upper case.
4377   \newcommand*{\GLsaccessname}[1]{\Glsentryname{#1}}

```

`\GLSaccessname` Display the name value (no link and no check for existence). converted to upper case.

```

4378 \newcommand*{\GLSaccessname}[1]{%
4379 \protect\mfirstucMakeUppercase{\glentryname{#1}}}
```

`\glsaccesstext` Display the text value (no link and no check for existence).

```

4380 \newcommand*{\glsaccesstext}[1]{\glentrytext{#1}}
```

`\Glsaccesstext` Display the text value (no link and no check for existence) with the first letter converted to upper case.

```

4381 \newcommand*{\Glsaccesstext}[1]{\Glsentrytext{#1}}
```

`\GLSaccesstext` Display the text value (no link and no check for existence). converted to upper case.

```

4382 \newcommand*{\GLSaccesstext}[1]{%
4383 \protect\mfirstucMakeUppercase{\glentrytext{#1}}}
```

`glsaccessplural` Display the plural value (no link and no check for existence).

```

4384 \newcommand*{\glsaccessplural}[1]{\glentryplural{#1}}
```

`Glsaccessplural` Display the plural value (no link and no check for existence) with the first letter converted to upper case.

```

4385 \newcommand*{\Glsaccessplural}[1]{\Glsentryplural{#1}}
```

`GLSaccessplural` Display the plural value (no link and no check for existence). converted to upper case.

```

4386 \newcommand*{\GLSaccessplural}[1]{%
4387 \protect\mfirstucMakeUppercase{\glentryplural{#1}}}
```

`\glsaccessfirst` Display the first value (no link and no check for existence).

```

4388 \newcommand*{\glsaccessfirst}[1]{\glentryfirst{#1}}
```

`\Glsaccessfirst` Display the first value (no link and no check for existence) with the first letter converted to upper case.

```

4389 \newcommand*{\Glsaccessfirst}[1]{\Glsentryfirst{#1}}
```

`\GLSaccessfirst` Display the first value (no link and no check for existence). converted to upper case.

```

4390 \newcommand*{\GLSaccessfirst}[1]{%
4391 \protect\mfirstucMakeUppercase{\glentryfirst{#1}}}
```

`cessfirstplural` Display the firstplural value (no link and no check for existence).

```

4392 \newcommand*{\glsaccessfirstplural}[1]{\glentryfirstplural{#1}}
```

`Glsfirstplural` Display the firstplural value (no link and no check for existence) with the first letter converted to upper case.

```

4393 \newcommand*{\Glsaccessfirstplural}[1]{\Glsentryfirstplural{#1}}
```

`GLSfirstplural` Display the firstplural value (no link and no check for existence). converted to upper case.

```

4394 \newcommand*{\GLSaccessfirstplural}[1]{%
4395 \protect\mfirstucMakeUppercase{\glentryfirstplural{#1}}}
```

`\glsaccesssymbol` Display the symbol value (no link and no check for existence).  
4396 `\newcommand*{\glsaccesssymbol}[1]{\glsentrysymbol{#1}}`

`\Glsaccesssymbol` Display the symbol value (no link and no check for existence) with the first letter converted to upper case.  
4397 `\newcommand*{\Glsaccesssymbol}[1]{\Glsentrysymbol{#1}}`

`\GLSaccesssymbol` Display the symbol value (no link and no check for existence). converted to upper case.  
4398 `\newcommand*{\GLSaccesssymbol}[1]{%`  
4399 `\protect\mfirstucMakeUppercase{\glsentrysymbol{#1}}}`

`\glsaccesssymbolplural` Display the symbolplural value (no link and no check for existence).  
4400 `\newcommand*{\glsaccesssymbolplural}[1]{\glsentrysymbolplural{#1}}`

`\Glsaccesssymbolplural` Display the symbolplural value (no link and no check for existence) with the first letter converted to upper case.  
4401 `\newcommand*{\Glsaccesssymbolplural}[1]{\Glsentrysymbolplural{#1}}`

`\GLSaccesssymbolplural` Display the symbolplural value (no link and no check for existence). converted to upper case.  
4402 `\newcommand*{\GLSaccesssymbolplural}[1]{%`  
4403 `\protect\mfirstucMakeUppercase{\glsentrysymbolplural{#1}}}`

`\glsaccessdesc` Display the desc value (no link and no check for existence).  
4404 `\newcommand*{\glsaccessdesc}[1]{\glsentrydesc{#1}}`

`\Glsaccessdesc` Display the desc value (no link and no check for existence) with the first letter converted to upper case.  
4405 `\newcommand*{\Glsaccessdesc}[1]{\Glsentrydesc{#1}}`

`\GLSaccessdesc` Display the desc value (no link and no check for existence). converted to upper case.  
4406 `\newcommand*{\GLSaccessdesc}[1]{%`  
4407 `\protect\mfirstucMakeUppercase{\glsentrydesc{#1}}}`

`\glsaccessdescplural` Display the descplural value (no link and no check for existence).  
4408 `\newcommand*{\glsaccessdescplural}[1]{\glsentrydescplural{#1}}`

`\Glsaccessdescplural` Display the descplural value (no link and no check for existence) with the first letter converted to upper case.  
4409 `\newcommand*{\Glsaccessdescplural}[1]{\Glsentrydescplural{#1}}`

`\GLSaccessdescplural` Display the descplural value (no link and no check for existence). converted to upper case.  
4410 `\newcommand*{\GLSaccessdescplural}[1]{%`  
4411 `\protect\mfirstucMakeUppercase{\glsentrydescplural{#1}}}`

`\glsaccessshort` Display the short form (no link and no check for existence).  
4412 `\newcommand*{\glsaccessshort}[1]{\glsentryshort{#1}}`



`\Glsaccessshort` Display the short form with first letter converted to uppercase (no link and no check for existence).

```
4413 \newcommand*{\Glsaccessshort}[1]{\Glsentryshort{#1}}
```

`\GLSaccessshort` Display the short value (no link and no check for existence). converted to upper case.

```
4414 \newcommand*{\GLSaccessshort}[1]{%
4415 \protect\mfirstucMakeUppercase{\Glsentryshort{#1}}}
```

`lsaccessshortpl` Display the short plural form (no link and no check for existence).

```
4416 \newcommand*{\lsaccessshortpl}[1]{\Glsentryshortpl{#1}}
```

`lSaccessshortpl` Display the short plural form with first letter converted to uppercase (no link and no check for existence).

```
4417 \newcommand*{\lSaccessshortpl}[1]{\Glsentryshortpl{#1}}
```

`LSaccessshortpl` Display the shortplural value (no link and no check for existence). converted to upper case.

```
4418 \newcommand*{\LSaccessshortpl}[1]{%
4419 \protect\mfirstucMakeUppercase{\Glsentryshortpl{#1}}}
```

`\glsaccesslong` Display the long form (no link and no check for existence).

```
4420 \newcommand*{\glsaccesslong}[1]{\Glsentrylong{#1}}
```

`\Glsaccesslong` Display the long form (no link and no check for existence).

```
4421 \newcommand*{\Glsaccesslong}[1]{\Glsentrylong{#1}}
```

`\GLSaccesslong` Display the long value (no link and no check for existence). converted to upper case.

```
4422 \newcommand*{\GLSaccesslong}[1]{%
4423 \protect\mfirstucMakeUppercase{\Glsentrylong{#1}}}
```

`glsaccesslongpl` Display the long plural form (no link and no check for existence).

```
4424 \newcommand*{\glsaccesslongpl}[1]{\Glsentrylongpl{#1}}
```

`Glsaccesslongpl` Display the long plural form (no link and no check for existence).

```
4425 \newcommand*{\Glsaccesslongpl}[1]{\Glsentrylongpl{#1}}
```

`GLSaccesslongpl` Display the longplural value (no link and no check for existence). converted to upper case.

```
4426 \newcommand*{\GLSaccesslongpl}[1]{%
4427 \protect\mfirstucMakeUppercase{\Glsentrylongpl{#1}}}
```

End of else part

```
4428 }
```

## 1.5 Categories

`\glscategory` Add a new storage key that can be used to indicate a category. The default category is general.

```
4429 \glsaddstoragekey{category}{general}{\glscategory}
```

`\glsifcategory` Convenient shortcut to determine if an entry has the given category.

```
4430 \newcommand{\glsifcategory}[4]{%
4431   \ifglstfield{#1}{category}{#2}{#3}{#4}%
4432 }
```

Categories can have attributes.

`categoryattribute`

```
\glsssetcategoryattribute{<category>}{<attribute-label>}{<value>}
```

Set (or override if already set) an attribute for the given category.

```
4433 \newcommand*\glsssetcategoryattribute}[3]{%
4434   \csdef{@glstxtr@categoryattr@@#1@#2}{#3}%
4435 }
```

`categoryattribute`

```
\glstgetcategoryattribute{<category>}{<attribute-label>}
```

Get the value of the given attribute for the given category. Does nothing if the attribute isn't defined.

```
4436 \newcommand*\glstgetcategoryattribute}[2]{%
4437   \csuse{@glstxtr@categoryattr@@#1@#2}%
4438 }
```

`categoryattribute`

```
\glshascategoryattribute{<category>}{<attribute-label>}{<true>}{<false>}
```

Tests if the category has the given attribute set.

```
4439 \newcommand*\glshascategoryattribute}[4]{%
4440   \ifcsvoid{@glstxtr@categoryattr@@#1@#2}{#4}{#3}%
4441 }
```

`\glsssetattribute`

```
\glsssetattribute{<entry label>}{<attribute-label>}{<value>}
```

Short cut where the category label is obtained from the entry information.

```
4442 \newcommand*\glsssetattribute}[3]{%
```

```

4443 \glsssetcategoryattribute{\glscategory{#1}}{#2}{#3}%
4444 }

```

`\glsggetattribute` `\glsggetattribute{<entry label>}{<attribute-label>}`

Short cut where the category label is obtained from the entry information.

```

4445 \newcommand*{\glsggetattribute}[2]{%
4446 \glsssetcategoryattribute{\glscategory{#1}}{#2}%
4447 }

```

`\glshasattribute` `\glshasattribute{<entry label>}{<attribute-label>}{<true>}{<false>}`

Short cut to test if the given attribute has been set where the category label is obtained from the entry information.

```

4448 \newcommand*{\glshasattribute}[4]{%
4449 \ifglssentryexists{#1}%
4450 {\glshascategoryattribute{\glscategory{#1}}{#2}{#3}{#4}}%
4451 {#4}%
4452 }

```

`\glssifcategoryattribute` `\glssifcategoryattribute{<category>}{<attribute-label>}{<value>}{<true part>}{<false part>}`

True if category has the attribute with the given value.

```

4453 \newcommand{\glssifcategoryattribute}[5]{%
4454 \ifcsundef{@glssxtr@categoryattr@#1@#2}%
4455 {#5}%
4456 {\ifcsstring{@glssxtr@categoryattr@#1@#2}{#3}{#4}{#5}}%
4457 }

```

`\glssifattribute` `\glssifattribute{<entry label>}{<attribute-label>}{<value>}{<true part>}{<false part>}`

Short cut to determine if the given entry has a category with the given attribute set.

```

4458 \newcommand{\glssifattribute}[5]{%
4459 \ifglssentryexists{#1}%
4460 {\glssifcategoryattribute{\glscategory{#1}}{#2}{#3}{#4}{#5}}%
4461 {#5}%
4462 }

```

Set attributes for the default general category:

```
4463 \glsetcategoryattribute{general}{regular}{true}
```

Acronyms are regular by default, since they're typically just treated like normal words.

```
4464 \glsetcategoryattribute{acronym}{regular}{true}
```

`regularcategory` Convenient shortcut to create add the regular attribute.

```
4465 \newcommand*{\glsetregularcategory}[1]{%
4466   \glsetcategoryattribute{#1}{regular}{true}%
4467 }
```

`ifregularcategory` `\gl@ifregularcategory{<category>}{<true part>}{<false part>}`

Short cut to determine if a category has the regular attribute explicitly set to true.

```
4468 \newcommand{\gl@ifregularcategory}[3]{%
4469   \gl@ifcategoryattribute{#1}{regular}{true}{#2}{#3}%
4470 }
```

`ifnotregularcategory` `\gl@ifnotregularcategory{<category>}{<true part>}{<false part>}`

Short cut to determine if a category has the regular attribute explicitly set to false.

```
4471 \newcommand{\gl@ifnotregularcategory}[3]{%
4472   \gl@ifcategoryattribute{#1}{regular}{false}{#2}{#3}%
4473 }
```

`\gl@ifregular` `\gl@ifregular{<entry label>}{<true part>}{<false part>}`

Short cut to determine if an entry has a regular attribute set to true.

```
4474 \newcommand{\gl@ifregular}[3]{%
4475   \gl@ifregularcategory{\glscategory{#1}}{#2}{#3}%
4476 }
```

`\gl@ifnotregular` `\gl@ifnotregular{<entry label>}{<true part>}{<false part>}`

Short cut to determine if an entry has a regular attribute set to false.

```
4477 \newcommand{\gl@ifnotregular}[3]{%
4478   \gl@ifnotregularcategory{\glscategory{#1}}{#2}{#3}%
4479 }
```

oreachincategory

```
\glsforeachincategory[⟨glossary labels⟩]{⟨category-label⟩}
{⟨glossary-cs⟩}{⟨label-cs⟩}{⟨body⟩}
```

Iterates through all entries in all the glossaries (or just those listed in *⟨glossary labels⟩*) and does *⟨body⟩* if the category matches *⟨category-label⟩*. The control sequences *⟨glossary-cs⟩* and *⟨label-cs⟩* may be used in *⟨body⟩* to access the glossary label and entry label for the current iteration.

```
4480 \newcommand{\glsforeachincategory}[5][\@glo@types]{%
4481   \forallglossaries[#1]{#3}%
4482   {%
4483     \forlgsentries[#3]{#4}%
4484     {%
4485       \glsifcategory{#4}{#2}{#5}{}%
4486     }%
4487   }%
4488 }
```

achwithattribute

```
\glsforeachwithattribute[⟨glossary labels⟩]{⟨attribute-label⟩}
{⟨attribute-value⟩}{⟨glossary-cs⟩}{⟨label-cs⟩}{⟨body⟩}
```

Iterates through all entries in all the glossaries (or just those listed in *⟨glossary labels⟩*) and does *⟨body⟩* if the category attribute *⟨attribute-label⟩* matches *⟨attribute-value⟩*. The control sequences *⟨glossary-cs⟩* and *⟨label-cs⟩* may be used in *⟨body⟩* to access the glossary label and entry label for the current iteration.

```
4489 \newcommand{\glsforeachwithattribute}[6][\@glo@types]{%
4490   \forallglossaries[#1]{#4}%
4491   {%
4492     \forlgsentries[#4]{#5}%
4493     {%
4494       \glsifattribute{#5}{#2}{#3}{#6}{}%
4495     }%
4496   }%
4497 }
```

If `\newterm` has been defined, redefine it so that it automatically sets the category label to `index` and add `\glstrpostdescription`.

```
4498 \ifdef\newterm
4499 {%
```

`\newterm`

```
4500   \renewcommand*{\newterm}[2][ ]{%
4501     \newglossaryentry{#2}%
4502     {type={index},category=index,name={#2},%
```

```

4503     description={\glxtrpostdescription\nopostdesc},#1}%
4504 }

```

Indexed terms are regular by default.

```

4505 \glissetcategoryattribute{index}{regular}{true}

```

trpostdescindex

```

4506 \newcommand*{\glxtrpostdescindex}{}

4507 }
4508 {}

```

If the symbols package option was used, define a similar command for symbols, but set the default sort to the label rather than the name as the symbols will typically contain commands that will confuse makeindex and xindy.

```

4509 \ifdef\printsymbols
4510 {%

```

glxtrnewsymbol Unlike \newterm, this has a separate argument for the label (since the symbol will likely contain commands).

```

4511 \newcommand*{\glxtrnewsymbol}[3] [] {%
4512 \newglossaryentry{#2}{name={#3},sort={#2},type=symbols,category=symbol,#1}%
4513 }

```

Symbols are regular by default.

```

4514 \glissetcategoryattribute{symbol}{regular}{true}

```

rpostdescsymbol

```

4515 \newcommand*{\glxtrpostdescsymbol}{}

4516 }
4517 {}

```

Similar for the numbers option.

```

4518 \ifdef\printnumbers
4519 {%

```

glxtrnewnumber

```

4520 \ifdef\printnumbers
4521 \newcommand*{\glxtrnewnumber}[3] [] {%
4522 \newglossaryentry{#2}{name={#3},sort={#2},type=numbers,category=number,#1}%
4523 }

```

Numbers are regular by default.

```

4524 \glissetcategoryattribute{number}{regular}{true}

```

rpostdescnumber

```

4525 \newcommand*{\glxtrpostdescnumber}{}

```

```

4526 }
4527 {}

```

`\glstrsetcategory` Set the category for all listed labels. The first argument is the list of entry labels and the second argument is the category label.

```

4528 \newcommand*{\glstrsetcategory}[2]{%
4529   \@for\@glstr@label:=#1\do
4530   {%
4531     \glsfieldxdef{\@glstr@label}{category}{#2}%
4532   }%
4533 }

```

`\glstrcategoryforall` Set the category for all entries in the listed glossaries. The first argument is the list of glossary labels and the second argument is the category label.

```

4534 \newcommand*{\glstrcategoryforall}[2]{%
4535   \forallglossaries[#1]{\@glstr@type}{%
4536     \forallsentries[\@glstr@type]{\@glstr@label}%
4537     {%
4538       \glsfieldxdef{\@glstr@label}{category}{#2}%
4539     }%
4540   }%
4541 }

```

`\glstrfieldtitlecase` `\glstrfieldtitlecase{\langle label \rangle}{\langle field \rangle}`

Apply title casing to the contents of the given field.

```

4542 \newcommand*{\glstrfieldtitlecase}[2]{%
4543   \expandafter\glstrfieldtitlecasesecs\expandafter
4544   {\csname glo@glsdetoklabel{#1}@#2\endcsname}%
4545 }

```

`\glstrfieldtitlecasesecs` The command used by `\glstrfieldtitlecase`. May be redefined to use a different command, for example, `\xcapitalisefmtwords`.

```

4546 \newcommand*{\glstrfieldtitlecasesecs}[1]{\xcapitalisewords{#1}}

```

Provide a convenient way to modify glossary styles without having to define a new style just to convert the first letter of fields to upper case.

`\glossentrydesc` If the `glossdesc` attribute is “firstuc” convert first letter to upper case. If the attribute is “title” use title case.

```

4547 \@ifpackageloaded{glossaries-accsupp}
4548 {
4549   \renewcommand*{\glossentrydesc}[1]{%
4550     \glsdoifexistsorwarn{#1}%
4551     {%
4552       \glssetabbrvfmt{\glscategory{#1}}%

```

As from version 1.04, allow the glossdescfont attribute to determine the font applied.

```

4553 \glshasattribute{#1}{glossdescfont}%
4554 {%
4555 \edef\@glxtr@attrval{\glsggetattribute{#1}{glossdescfont}}%
4556 \ifcsdef{\@glxtr@attrval}%
4557 {%
4558 \letcs{\@glxtr@glossdescfont}{\@glxtr@attrval}%
4559 }%
4560 {%
4561 \GlossariesExtraWarning{Unknown control sequence name
4562 '\@glxtr@attrval' supplied in glossdescfont attribute
4563 for entry '#1'. Ignoring}%
4564 \let\@glxtr@glossdescfont\@firstofone
4565 }%
4566 }%
4567 {\let\@glxtr@glossdescfont\@firstofone}%
4568 \glusifattribute{#1}{glossdesc}{firstuc}%
4569 {%
4570 \@glxtr@glossdescfont{\Glsaccessdesc{#1}}%
4571 }%
4572 {%
4573 \glusifattribute{#1}{glossdesc}{title}%
4574 {%
4575 \@glxtr@do@titlecaps@warn
4576 \glsdescriptionaccessdisplay
4577 {%
4578 \@glxtr@glossdescfont{\glxtrfieldtitlecase{#1}{desc}}%
4579 }%
4580 {#1}%
4581 }%
4582 {%
4583 \@glxtr@glossdescfont{\Glsaccessdesc{#1}}%
4584 }%
4585 }%
4586 }%
4587 }
4588 }
4589 {
4590 \renewcommand*\glossentrydesc[1]{%
4591 \glsdoifexistsorwarn{#1}%
4592 {%
4593 \glissetabbrvfmt{\glscategory{#1}}%
4594 \glshasattribute{#1}{glossdescfont}%
4595 {%
4596 \edef\@glxtr@attrval{\glsggetattribute{#1}{glossdescfont}}%
4597 \ifcsdef{\@glxtr@attrval}%
4598 {%
4599 \letcs{\@glxtr@glossdescfont}{\@glxtr@attrval}%
4600 }%

```



```

4601      {%
4602      \GlossariesExtraWarning{Unknown control sequence name
4603      '\@glsxtr@attrval' supplied in glossdescfont attribute
4604      for entry '#1'. Ignoring}%
4605      \let\@glsxtr@glossdescfont\@firstofone
4606      }%
4607  }%
4608  {\let\@glsxtr@glossdescfont\@firstofone}%
4609  \glsifattribute{#1}{glossdesc}{firstuc}%
4610  {%
4611      \@glsxtr@glossdescfont{\Glsentrydesc{#1}}%
4612  }%
4613  {%
4614      \glsifattribute{#1}{glossdesc}{title}%
4615      {%
4616          \@glsxtr@do@titlecaps@warn
4617          \@glsxtr@glossdescfont{\glsxtrfieldtitlecase{#1}{desc}}%
4618      }%
4619      {%
4620          \@glsxtr@glossdescfont{\Glsentrydesc{#1}}%
4621      }%
4622  }%
4623  }%
4624  }
4625  }

```

`\glossentryname` If the glossname attribute is “firstuc” convert first letter to upper case. If the attribute is “title” use title case.

```

4626 \ifpackageloaded{glossaries-accsupp}
4627 {
4628   \renewcommand*{\glossentryname}[1]{%
4629     \@glsdoifexistsorwarn{#1}%
4630     {%
4631       \glssetabbrvfmt{\glscategory{#1}}%

```

As from version 1.04, allow the glossnamefont attribute to determine the font applied.

```

4632   \glshasattribute{#1}{glossnamefont}%
4633   {%
4634     \edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
4635     \ifcsdef{\@glsxtr@attrval}%
4636     {%
4637       \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
4638     }%
4639     {%
4640       \GlossariesExtraWarning{Unknown control sequence name
4641       '\@glsxtr@attrval' supplied in glossnamefont attribute
4642       for entry '#1'. Reverting to default \string\glsnamefont}%
4643       \let\@glsxtr@glossnamefont\glsnamefont
4644     }%
4645   }%

```

```

4646 {\let\@glsxtr@glossnamefont\glsnamefont}%
4647 \glsifattribute{#1}{glossname}{firstuc}%
4648 {%
4649   \glsnameaccessdisplay
4650   {%
4651     \@glsxtr@glossnamefont{\Glsentryname{#1}}%
4652   }%
4653   {#1}%
4654 }%
4655 {%
4656   \glsifattribute{#1}{glossname}{title}%
4657   {%
4658     \@glsxtr@do@titlecaps@warn
4659     \glsnameaccessdisplay
4660     {%
4661       \@glsxtr@glossnamefont{\glsxtrfieldtitlecase{#1}{name}}%
4662     }%
4663     {#1}%
4664   }%
4665   {%
4666     \glsifattribute{#1}{glossname}{uc}%
4667     {%
4668       \glsnameaccessdisplay
4669       {%

```

Hide the label from the upper-casing command.

```

4670       \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
4671       \@glsxtr@glossnamefont{\mfirstucMakeUppercase{\glo@name}}%
4672     }%
4673     {#1}%
4674   }%
4675   {%
4676     \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
4677     \glsnameaccessdisplay
4678     {%
4679       \expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}%
4680     }%
4681     {#1}%
4682   }%
4683 }%
4684 }%

```

Do post-name hook:

```

4685   \glsxtrpostnamehook{#1}%
4686 }%
4687 }
4688 }
4689 {
4690   \renewcommand*{\glossentryname}[1]{%
4691     \@glsdoifexistsorwarn{#1}%

```

```

4692 {%
4693     \glsetabbrvfmt{\glscategory{#1}}%
4694     \glshasattribute{#1}{glossnamefont}%
4695     {%
4696         \edef\@glxtr@attrval{\glsggetattribute{#1}{glossnamefont}}%
4697         \ifcsdef{\@glxtr@attrval}%
4698         {%
4699             \letcs{\@glxtr@glossnamefont}{\@glxtr@attrval}%
4700         }%
4701         {%
4702             \GlossariesExtraWarning{Unknown control sequence name
4703             '\@glxtr@attrval' supplied in glossnamefont attribute
4704             for entry '#1'. Reverting to default \string\glnamefont}%
4705             \let\@glxtr@glossnamefont\glnamefont
4706         }%
4707     }%
4708     {\let\@glxtr@glossnamefont\glnamefont}%
4709     \gl@ifattribute{#1}{glossname}{firstuc}%
4710     {%
4711         \@glxtr@glossnamefont{\Glsentryname{#1}}%
4712     }%
4713     {%
4714         \gl@ifattribute{#1}{glossname}{title}%
4715         {%
4716             \@glxtr@do@titlecaps@warn
4717             \@glxtr@glossnamefont{\glxtrfieldtitlecase{#1}{name}}%
4718         }%
4719         {%
4720             \gl@ifattribute{#1}{glossname}{uc}%
4721             {%

```

Hide the label from the upper-casing command.

```

4722         \letcs{\glo@name}{\glo@glsetoklabel{#1}@name}%
4723         \@glxtr@glossnamefont{\mfirstucMakeUppercase{\glo@name}}%
4724     }%
4725     {%

```

This little trick is used by glossaries to allow the user to redefine \glnamefont to use \makefirstuc. Support it even though they can now use the firstuc attribute.

```

4726         \letcs{\glo@name}{\glo@glsetoklabel{#1}@name}%
4727         \expandafter\@glxtr@glossnamefont\expandafter{\glo@name}%
4728     }%
4729 }%
4730 }%

```

Do post-name hook.

```

4731     \glxtrpostnamehook{#1}%
4732 }%
4733 }
4734 }

```

`\Glossentryname` Redefine to set the abbreviation format and accessibility support.

```
4735 \ifpackageloaded{glossaries-accsupp}
4736 {
4737   \renewcommand*{\Glossentryname}[1]{%
4738     \@glsdoifexistsorwarn{#1}%
4739     {%
4740       \glsetabbrvfmt{\glscategory{#1}}%
```

As from version 1.04, allow the `glossnamefont` attribute to determine the font applied.

```
4741     \glshasattribute{#1}{glossnamefont}%
4742     {%
4743       \edef\@glstr@attrval{\glsetattribute{#1}{glossnamefont}}%
4744       \ifcsdef{\@glstr@attrval}%
4745       {%
4746         \letcs{\@glstr@glossnamefont}{\@glstr@attrval}%
4747       }%
4748       {%
4749         \GlossariesExtraWarning{Unknown control sequence name
4750           '\@glstr@attrval' supplied in glossnamefont attribute
4751           for entry '#1'. Reverting to default \string\glnamefont}%
4752         \let\@glstr@glossnamefont\glnamefont
4753       }%
4754     }%
4755     {\let\@glstr@glossnamefont\glnamefont}%
4756     \glnameaccessdisplay
4757     {%
4758       \@glstr@glossnamefont{\Glsentryname{#1}}%
4759     }%
4760     {#1}%
```

Do post-name hook:

```
4761     \glstrpostnamehook{#1}%
4762   }%
4763 }
4764 }
4765 {
4766   \renewcommand*{\Glossentryname}[1]{%
4767     \@glsdoifexistsorwarn{#1}%
4768     {%
4769       \glsetabbrvfmt{\glscategory{#1}}%
4770       \glshasattribute{#1}{glossnamefont}%
4771       {%
4772         \edef\@glstr@attrval{\glsetattribute{#1}{glossnamefont}}%
4773         \ifcsdef{\@glstr@attrval}%
4774         {%
4775           \letcs{\@glstr@glossnamefont}{\@glstr@attrval}%
4776         }%
4777         {%
4778           \GlossariesExtraWarning{Unknown control sequence name
4779             '\@glstr@attrval' supplied in glossnamefont attribute
```

```

4780         for entry ‘#1’. Reverting to default \string\glsnamefont}%
4781         \let\@glsxtr@glossnamefont\glsnamefont
4782     }%
4783 }%
4784 {\let\@glsxtr@glossnamefont\glsnamefont}%
4785 \@glsxtr@glossnamefont{\Glsentryname{#1}}}%

```

Do post-name hook:

```

4786     \glsxtrpostnamehook{#1}%
4787 }%
4788 }
4789 }

```

Provide a convenient way to also index the entries using the standard `\index` mechanism. This may use different actual, encap and escape characters to those used for the glossaries.

`xtrpostnamehook` Hook to append stuff after the name is displayed in the glossary. The argument is the entry's label.

```

4790 \newcommand*{\glsxtrpostnamehook}[1]{%
4791   \let\@glsnumberformat\@glsxtr@defaultnumberformat
4792   \glsxtrdoautoindexname{#1}{indexname}%

```

Allow categories to hook in here.

```

4793   \csuse{glsxtrpostname\glscategory{#1}}%
4794 }

```

`format@override` Determines if the format key should override the indexing attribute value.

```

4795 \newif\ifglsxtr@format@override
4796 \@glsxtr@format@overridefalse

```

If overriding is enabled, the `\glshypernumber` command will have to be redefined in the index to use `\hyperpage` instead.

`xFormatOverride`

```

4797 \@ifpackageloaded{hyperref}
4798 {

```

If `hyperref`'s `hyperindex` option is on, then `hyperref` will automatically add `\hyperpage`, so don't add it.

```

4799   \ifHy@hyperindex
4800     \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
4801       \@glsxtr@format@override true
4802       \appto\theindex{\let\glshypernumber\@firstofone}%
4803     }
4804   \else
4805     \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
4806       \@glsxtr@format@override true
4807       \appto\theindex{\let\glshypernumber\hyperpage}%
4808     }
4809   \fi

```

```

4810 }
4811 {
4812   \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
4813     \@glsxtr@format@override true
4814   }
4815 }
4816 \@onlypreamble\GlsXtrEnableIndexFormatOverride

```

doautoindexname

```

4817 \newcommand*{\glxtrdoautoindexname}[2]{%
4818   \glshasattribute{#1}{#2}%
4819   {%

```

Escape any makeindex/xindy characters in the value of the name field. Take care with babel as this won't work if the category code has changed for those characters.

```

4820     \@glsxtr@autoindex@setname{#1}%

```

If the attribute value is simply “true” don't add an encap, otherwise use the value as the encap.

```

4821     \protected@edef\@glsxtr@attrval{\glsggetattribute{#1}{#2}}%
4822     \if@glsxtr@format@override

4823       \ifx\@glsnumberformat\@glsxtr@defaultnumberformat
4824       \else
4825         \let\@glsxtr@attrval\@glsnumberformat
4826       \fi
4827     \fi
4828     \ifdefstring{\@glsxtr@attrval}{true}%
4829     {}%
4830     {\eappto\@glo@name{\@glsxtr@autoindex@encap\@glsxtr@attrval}}%
4831     \expandafter\glxtrautoindex\expandafter{\@glo@name}%
4832   }%
4833   {}%
4834 }

```

glxtrautoindex

```

4835 \newcommand*{\glxtrautoindex}{\index}

```

toindex@setname Assign \@glo@name for use with indexname attribute.

```

4836 \newcommand*{\@glsxtr@autoindex@setname}[1]{%
4837   \protected@edef\@glo@name{\glxtrautoindexentry{#1}}%
4838   \glxtrautoindexassignsort{\@glo@sort}{#1}%
4839   \@gls@checkmkidxchars\@glo@sort
4840   \@glsxtr@autoindex@doextra@esc\@glo@sort
4841   \epreto\@glo@name{\@glo@sort\@glsxtr@autoindex@at}%
4842 }

```

rautoindexentry Command used for the actual part when auto-indexing.

```

4843 \newcommand*{\glxtrautoindexentry}[1]{\string\glstryname{#1}}

```

trautoindexsort Used to assign the sort value when auto-indexing.

```
4844 \newcommand*{\glxtrautoindexassignsort}[2]{%
4845   \glslentryfield{#1}{#2}{sort}%
4846 }
```

dex@doextra@esc

```
4847 \newcommand*{\@glxtr@autoindex@doextra@esc}[1]{%
```

Escape the escape character unless it has already been escaped.

```
4848   \ifx\@glxtr@autoindex@esc\@gl@quotechar
4849   \else
4850     \def\@gl@checkedmkidx{}%
4851     \edef\@glxtr@checkspch{%
4852       \noexpand\@glxtr@autoindex@escquote\expandonce{#1}%
4853       \noexpand\@empty\@glxtr@autoindex@esc\noexpand\@nnil
4854       \@glxtr@autoindex@esc\noexpand\@empty\noexpand\@glxtr@endescspch}%
4855     \@glxtr@checkspch
4856     \let#1\@gl@checkedmkidx\relax
4857   \fi
```

Escape actual character unless it has already been escaped.

```
4858   \ifx\@glxtr@autoindex@at\@gl@actualchar
4859   \else
4860     \def\@gl@checkedmkidx{}%
4861     \edef\@glxtr@checkspch{%
4862       \noexpand\@glxtr@autoindex@escat\expandonce{#1}%
4863       \noexpand\@empty\@glxtr@autoindex@at\noexpand\@nnil
4864       \@glxtr@autoindex@at\noexpand\@empty\noexpand\@glxtr@endescspch}%
4865     \@glxtr@checkspch
4866     \let#1\@gl@checkedmkidx\relax
4867   \fi
```

Escape level character unless it has already been escaped.

```
4868   \ifx\@glxtr@autoindex@level\@gl@levelchar
4869   \else
4870     \def\@gl@checkedmkidx{}%
4871     \edef\@glxtr@checkspch{%
4872       \noexpand\@glxtr@autoindex@esclevel\expandonce{#1}%
4873       \noexpand\@empty\@glxtr@autoindex@level\noexpand\@nnil
4874       \@glxtr@autoindex@level\noexpand\@empty\noexpand\@glxtr@endescspch}%
4875     \@glxtr@checkspch
4876     \let#1\@gl@checkedmkidx\relax
4877   \fi
```

Escape encap character unless it has already been escaped.

```
4878   \ifx\@glxtr@autoindex@encap\@gl@encapchar
4879   \else
4880     \def\@gl@checkedmkidx{}%
4881     \edef\@glxtr@checkspch{%
4882       \noexpand\@glxtr@autoindex@escencap\expandonce{#1}%
4883       \noexpand\@empty\@glxtr@autoindex@encap\noexpand\@nnil
```

```

4884      \@glsxtr@autoindex@encap\noexpand\@empty\noexpand\@glsxtr@endescspch}%
4885      \@glsxtr@checkspch
4886      \let#1\@gls@checkedmkidx\relax
4887      \fi
4888 }

```

The user commands here have a preamble-only restriction to ensure they are set before required and also to reduce the chances of complications caused by babel's shorthands.

`\tr@autoindex@at` Actual character for use with `\index`.

```
4889 \newcommand*{\@glsxtr@autoindex@at}{}

```

`\trSetActualChar` Set the actual character.

```

4890 \newcommand*{\GlsXtrSetActualChar}[1]{%
4891   \gdef\@glsxtr@autoindex@at{#1}%
4892   \def\@glsxtr@autoindex@escat##1#1##2#1##3\@glsxtr@endescspch{%
4893     \@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escat}{##1}{##2}{##3}%
4894   }%
4895 }
4896 \@onlypreamble\GlsXtrSetActualChar
4897 \makeatother
4898 \GlsXtrSetActualChar{@}
4899 \makeatletter

```

`\autoindex@encap` Encap character for use with `\index`.

```
4900 \newcommand*{\@glsxtr@autoindex@encap}{}

```

`\XtrSetEncapChar` Set the encap character.

```

4901 \newcommand*{\GlsXtrSetEncapChar}[1]{%
4902   \gdef\@glsxtr@autoindex@encap{#1}%
4903   \def\@glsxtr@autoindex@escencap##1#1##2#1##3\@glsxtr@endescspch{%
4904     \@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escencap}{##1}{##2}{##3}%
4905   }%
4906 }
4907 \GlsXtrSetEncapChar{|}
4908 \@onlypreamble\GlsXtrSetEncapChar

```

`\autoindex@level` Level character for use with `\index`.

```
4909 \newcommand*{\@glsxtr@autoindex@level}{}

```

`\XtrSetLevelChar` Set the encap character.

```

4910 \newcommand*{\GlsXtrSetLevelChar}[1]{%
4911   \gdef\@glsxtr@autoindex@level{#1}%
4912   \def\@glsxtr@autoindex@esclevel##1#1##2#1##3\@glsxtr@endescspch{%
4913     \@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@esclevel}{##1}{##2}{##3}%
4914   }%
4915 }
4916 \GlsXtrSetLevelChar{!}
4917 \@onlypreamble\GlsXtrSetLevelChar

```



r@autoindex@esc Escape character for use with \index.

```
4918 \newcommand*{\@glsxtr@autoindex@esc}{"
```

lsXtrSetEscChar Set the escape character.

```
4919 \newcommand*{\GlsXtrSetEscChar}[1]{%
4920   \gdef\@glsxtr@autoindex@esc{#1}%
4921   \def\@glsxtr@autoindex@escquote##1##2##3\@glsxtr@endescspch{%
4922     \@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escquote}{##1}{##2}{##3}%
4923   }%
4924 }
4925 \GlsXtrSetEscChar{"}
4926 \@onlypreamble\GlsXtrSetEscChar
```

Set if defined. (For example, if doc package has been loaded.) Actual character \actualchar:

```
4927 \ifdef\actualchar
4928 {\expandafter\GlsXtrSetActualChar\expandafter{\actualchar}}
4929 {}
```

Quote character \quotechar:

```
4930 \ifdef\quotechar
4931 {\expandafter\GlsXtrSetEscChar\expandafter{\quotechar}}
4932 {}
```

Level character \levelchar:

```
4933 \ifdef\levelchar
4934 {\expandafter\GlsXtrSetLevelChar\expandafter{\levelchar}}
4935 {}
```

Encap character \encapchar:

```
4936 \ifdef\encapchar
4937 {\expandafter\GlsXtrSetEncapChar\expandafter{\encapchar}}
4938 {}
```

leto@endescspch

```
4939 \def\@glsxtr@gobbleto@endescspch#1\@glsxtr@endescspch{}
```

toindex@esc@spch

|  |
|--|
| <pre>\@glsxtr@autoindex@escspch{\&lt;char&gt;}{\&lt;cs&gt;}{\&lt;pre&gt;}{\&lt;mid&gt;}{\&lt;post&gt;}</pre> |
|--|

```
4940 \newcommand*{\@glsxtr@autoindex@escspch}[5]{%
4941   \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
4942   \toks@={#3}%
4943   \ifx\@nnil#3\relax
4944     \def\@glsxtr@checkspch{\@glsxtr@gobbleto@endescspch#5\@glsxtr@endescspch}%
4945   \else
4946     \ifx\@nnil#4\relax
4947       \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
4948       \def\@glsxtr@checkspch{\@glsxtr@gobbleto@endescspch
```

```

4949      #4#5\@glxtr@endescspch}%
4950    \else
4951      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
4952        \@glxtr@autoindex@esc#1}%
4953      \def\@@glxtr@checkspch{#2#5#1\@nnil#1\@glxtr@endescspch}%
4954    \fi
4955  \fi
4956  \@glxtr@checkspch
4957 }

```

`\Glossentrydesc` Redefine to set the abbreviation format and accessibility support.

```

4958 \renewcommand*{\Glossentrydesc}[1]{%
4959   \glsdoifexistsorwarn{#1}%
4960   {%
4961     \glssetabbrvfmt{\glscategory{#1}}%
4962     \Glsaccessdesc{#1}%
4963   }%
4964 }

```

`\Glossentrysymbol` Redefine to set the abbreviation format and accessibility support.

```

4965 \renewcommand*{\Glossentrysymbol}[1]{%
4966   \glsdoifexistsorwarn{#1}%
4967   {%
4968     \glssetabbrvfmt{\glscategory{#1}}%
4969     \Glsaccesssymbol{#1}%
4970   }%
4971 }

```

`\Glossentrysymbol` Redefine to set the abbreviation format and accessibility support.

```

4972 \renewcommand*{\Glossentrysymbol}[1]{%
4973   \glsdoifexistsorwarn{#1}%
4974   {%
4975     \glssetabbrvfmt{\glscategory{#1}}%
4976     \Glsaccesssymbol{#1}%
4977   }%
4978 }

```

Allow initials to be marked but only use the formatting for the tag in the glossary.

`\GlsXtrEnableInitialTagging` Allow initial tagging. The first argument is a list of categories to apply this to. The second argument is the name of the command to use to tag the initials. This can't already be defined for safety unless the starred version is used.

```

4979 \newcommand*{\GlsXtrEnableInitialTagging}{%
4980   \@ifstar\s@glxtr@enabletagging\@glxtr@enabletagging
4981 }
4982 \@onlypreamble\GlsXtrEnableInitialTagging

```

`\GlsXtrEnableInitialTagging` Starred version undefines command.

```

4983 \newcommand*{\s@glxtr@enabletagging}[2]{%
4984   \undef#2%
4985   \@glxtr@enabletagging{#1}{#2}%
4986 }

```

r@enabletagging Internal command.

```

4987 \newcommand*{\@glxtr@enabletagging}[2]{%
    Set attributes for categories given in the first argument.
4988   \@for\@glxtr@cat:=#1\do
4989   {%
4990     \ifdefempty\@glxtr@cat
4991     {}%
4992     {\glsssetcategoryattribute{\@glxtr@cat}{tagging}{true}}%
4993   }%
4994   \newrobustcmd*#2[1]{##1}%
4995   \def\@glxtr@taggingcs{#2}%
4996   \renewcommand*{\@glxtr@activate@initialtagging{%
4997     \let#2\@glxtr@tag
4998   }%
4999   \ifundef\@glx@preglossaryhook
5000   {\GlossariesExtraWarning{Initial tagging requires at least
5001     glossaries.sty v4.19 to work correctly}}%
5002   {}%
5003 }

```

Are we using an old version of mfirstuc that has a bug in \capitalisewords? If so, patch it so we don't have a problem with a combination of tagging and title case.

fu@checkword@do If this command hasn't been defined, then we have pre v2.02 of mfirstuc

```

5004 \ifundef\mfu@checkword@do
5005 {
5006   \newcommand*{\mfu@checkword@do}[1]{%
5007     \ifdefstring{\mfu@checkword@arg}{#1}%
5008     {%
5009       \let\@mfu@domakefirstuc\@firstofone
5010       \listbreak
5011     }%
5012     {}%
5013   }

```

\mfu@checkword \capitalisewords was introduced in mfirstuc v1.06. If \mfu@checkword hasn't been defined mfirstuc is too old to support the title case attribute.

```

5014 \ifundef\mfu@checkword
5015 {
5016   \newcommand{\@glxtr@do@titlecaps@warn}{%
5017     \GlossariesExtraWarning{mfirstuc.sty too old. Title Caps
5018       support not available}%

```

One warning should suffice.

```
5019 \let\@glxtr@do@titlecaps@warn\relax
5020 }
5021 }
5022 {
5023 \renewcommand*{\mfu@checkword}[1]{%
5024 \def\mfu@checkword@arg{#1}%
5025 \let\@mfu@domakefirstuc\makefirstuc
5026 \forlistloop\mfu@checkword@do\@mfu@nocaplist
5027 }
5028 }
5029 }
5030 {}% no patch required
```

@titlecaps@warn Do warning if title case not supported.

```
5031 \newcommand*{\@glxtr@do@titlecaps@warn}{}%
```

@initialtagging Used in \printglossary but at least v4.19 of glossaries required.

```
5032 \newcommand*{\@glxtr@activate@initialtagging}{}%
```

\@glxtr@tag Definition of tagging command when used in glossary.

```
5033 \newrobustcmd*{\@glxtr@tag}[1]{%
5034 \glsifattribute{\glscurrententrylabel}{tagging}{true}%
5035 {\glxtrtagfont{#1}}{#1}%
5036 }
```

\glxtrtagfont Used in the glossary.

```
5037 \newcommand*{\glxtrtagfont}[1]{\underline{#1}}
```

preglossaryhook This macro was introduced in glossaries version 4.19, so it may not be defined. If it hasn't been defined this feature is unavailable. A check is added for the entry's existence to prevent errors from occurring if the user removes an entry or changes the label, which can interrupt the build process.

```
5038 \ifdef\@gls@preglossaryhook
5039 {
5040 \renewcommand*{\@gls@preglossaryhook}{%
5041 \@glxtr@activate@initialtagging
```

Since the glossaries are automatically scoped, \@glxtr@org@postdescription shouldn't already be defined, but check anyway just as a precautionary measure.

```
5042 \ifundef\@glxtr@org@postdescription
5043 {%
5044 \let\@glxtr@org@postdescription\glspostdescription
5045 \renewcommand*{\glspostdescription}{%
5046 \ifgl Sentryexists{\glscurrententrylabel}%
5047 {%
5048 \glxtrpostdescription
5049 \@glxtr@org@postdescription
```

```

5050      }%
5051      {}%
5052      }%
5053      }%
5054      {}%

    Enable the options used by \@glsxtrp:
5055      \glossxtrsetpopts
5056      }%
5057  }
5058 {}

postdescription  This command will only be used if \@gls@preglossaryhook is available and the glossary
                  style uses \glspostdescription without modifying it. (\nopostdesc will suppress this.)
                  The glossaries-extra-stylemods package will add the post description hook to all the prede-
                  fined styles that don't include it.
5059 \newcommand*{\glsxtrpostdescription}{%
5060   \csuse{glsxtrpostdesc\glscategory{\glscurrententrylabel}}%
5061 }

postdescgeneral
5062 \newcommand*{\glsxtrpostdescgeneral}{}

xtrpostdescterm
5063 \newcommand*{\glsxtrpostdescterm}{}

postdescacronym
5064 \newcommand*{\glsxtrpostdescacronym}{}

descabbreviation
5065 \newcommand*{\glsxtrpostdescabbreviation}{}

glspostlinkhook  Redefine the post link hook used by commands like \gls to make it easier for categories
                  or attributes to modify this action. Since this hook occurs outside the existence check of
                  commands like \gls, this needs to be checked again here. Do nothing if the entry hasn't
                  been defined.
5066 \renewcommand*{\glspostlinkhook}{%
5067   \ifglsentryexists{\glslabel}{\glsxtrpostlinkhook}{}%
5068 }

xtrpostlinkhook  The entry label should already be stored in \glslabel by \@gls@link.
5069 \newcommand*{\glsxtrpostlinkhook}{%
5070   \glsxtrdiscardperiod{\glslabel}%
5071   {\glsxtrpostlinkendsentence}%
5072   {\glsxtrpostlink}%
5073 }

```

`\glxtrpostlink`

```
5074 \newcommand*{\glxtrpostlink}{%
5075   \csuse{\glxtrpostlink\glscategory{\glslabel}}%
5076 }
```

`linkendsentence` Done by `\glxtrpostlinkhook` if a full stop is discarded.

```
5077 \newcommand*{\glxtrpostlinkendsentence}{%
5078   \ifcsdef{\glxtrpostlink\glscategory{\glslabel}}
5079   {%
5080     \csuse{\glxtrpostlink\glscategory{\glslabel}}%
5081     .\spacefactor\sfcode'\. \relax
5082   }%
5083   {%
5084     \spacefactor\sfcode'\. \relax
5085   }%
5086 }
```

Put the full stop back.

```
5081 .\spacefactor\sfcode'\. \relax
5082 }%
5083 {%
5084 \spacefactor\sfcode'\. \relax
5085 }%
5086 }
```

Assume the full stop was discarded because the entry ends with a period, so adjust the spacefactor.

```
5084 \spacefactor\sfcode'\. \relax
5085 }%
5086 }
```

`addDescOnFirstUse` Provide a command for appending the description in parentheses on first use, for the convenience of users wanting to add this to the post link hook.

```
5087 \newcommand*{\glxtrpostlinkAddDescOnFirstUse}{%
5088   \glxtrifwasfirstuse{\space(\glssaccessdesc{\glslabel})}{}%
5089 }
```

`addSymbolOnFirstUse` Provide a command for appending the symbol (if defined) in parentheses on first use, for the convenience of users wanting to add this to the post link hook.

```
5090 \newcommand*{\glxtrpostlinkAddSymbolOnFirstUse}{%
5091   \glxtrifwasfirstuse
5092   {%
5093     \ifglshassymbol{\glslabel}{\space(\glssaccesssymbol{\glslabel})}{}%
5094   }%
5095   {%
5096 }
```

`trdiscardperiod` Discard following period (if present) if the `discardperiod` attribute is true. If a period is discarded, do the second argument otherwise do the third argument. The entry label is in the first argument. Since this is designed for abbreviations that end with a period, check if the plural form was used (which typically won't end with a period).

```
5097 \newcommand*{\glxtrdiscardperiod}[3]{%
5098   \glxtrifwasfirstuse
5099   {%
5100     \glusifattribute{#1}{retainfirstuseperiod}{true}%
5101     {#3}%
5102     {%
5103     }%
5104   }%
5105   {%
5106     }%
5107   }%
5108 }
```

```

5103 \glsifattribute{#1}{discardperiod}{true}%
5104 {%
5105 \glsifplural
5106 {%
5107 \glsifattribute{#1}{pluraldiscardperiod}{true}%
5108 {\glsxtrifperiod{#2}{#3}}%
5109 {#3}%
5110 }%
5111 {%
5112 \glsxtrifperiod{#2}{#3}%
5113 }%
5114 }%
5115 {#3}%
5116 }%
5117 }%
5118 {%
5119 \glsifattribute{#1}{discardperiod}{true}%
5120 {%
5121 \glsifplural
5122 {%
5123 \glsifattribute{#1}{pluraldiscardperiod}{true}%
5124 {\glsxtrifperiod{#2}{#3}}%
5125 {#3}%
5126 }%
5127 {%
5128 \glsxtrifperiod{#2}{#3}%
5129 }%
5130 }%
5131 {#3}%
5132 }%
5133 }

```

`\glsxtrifperiod` Make a convenient user command to check if the next character is a full stop (period). Works like `\@ifstar` but uses `\new@ifnextchar` rather than `\@ifnextchar`

```
5134 \newcommand*{\glsxtrifperiod}[1]{\new@ifnextchar.{\@firstoftwo{#1}}}
```

Sometimes it's useful to test if there's a punctuation character following the glossary entry.

`glsxtr@punclist` List of characters identified as punctuation marks. (Be careful of babel shorthands!) This doesn't allow for punctuation marks made up from multiple characters (such as ' ').

```
5135 \newcommand*{\glsxtr@punclist}{.,;?!}
```

`punctuationmark` Add character to punctuation list.

```
5136 \newcommand*{\glsxtraddpunctuationmark}[1]{\appto\glsxtr@punclist{#1}}
```

`unctuationmarks` Reset the punctuation list.

```
5137 \newcommand*{\glsxtrsetpunctuationmarks}[1]{\def\glsxtr@punclist{#1}}
```

`\glxtrifpunc` `\glxtrifnextpunc{<true part>}{<false part>}`

Test if this is followed by a punctuation mark. (Adapted from `\new@ifnextchar`.)

```
5138 \newcommand*{\glxtrifnextpunc}[2]{%
5139   \def\reserved@a{#1}%
5140   \def\reserved@b{#2}%
5141   \futurelet\@glxpunc@token\glxtr@ifnextpunc
5142 }
```

`glxtr@ifnextpunc`

```
5143 \newcommand*{\glxtr@ifnextpunc}{%
5144   \glxtr@ifpunctoken{\@glxpunc@token}{\let\reserved@b\reserved@a}{}%
5145   \reserved@b
5146 }
```

`glxtr@ifpunctoken` Test if the token given in the first argument is in the punctuation list.

```
5147 \newcommand*{\glxtr@ifpunctoken}[1]{%
5148   \expandafter\@glxtr@ifpunctoken\expandafter#1\glxtr@punclist\@nnil
5149 }
```

`glxtr@ifpunctoken`

```
5150 \def\@glxtr@ifpunctoken#1#2{%
5151   \let\reserved@d=#2%
5152   \ifx\reserved@d\@nnil
5153     \let\glxtr@next\@glxtr@notfoundinlist
5154   \else
5155     \ifx#1\reserved@d
5156       \let\glxtr@next\@glxtr@foundinlist
5157     \else
5158       \let\glxtr@next\@glxtr@ifpunctoken
5159     \fi
5160   \fi
5161   \glxtr@next#1%
5162 }
```

`glxtr@foundinlist`

```
5163 \def\@glxtr@foundinlist#1\@nnil{\@firstoftwo}
```

`@notfoundinlist`

```
5164 \def\@glxtr@notfoundinlist#1{\@secondoftwo}
```

`glxtr@dopostpunc`

`\glxtrdopostpunc{<code>}`



If this is followed by a punctuation character, do *<code>* after the character otherwise do *<code>* before whatever comes next.

```
5165 \newcommand{\glxtrdopostpunc}[1]{%
5166   \glxtrifnextpunc{\@glxtr@swaptwo{#1}}{#1}%
5167 }
```

@glxtr@swaptwo

```
5168 \newcommand{\@glxtr@swaptwo}[2]{#2#1}
```

## 1.6 Abbreviations

The “acronym” code from glossaries is misnamed as it’s more often used for other forms of abbreviations. This code corrects this inconsistency, but rather than just having synonyms, provide commands for abbreviations that have a similar, but not identical, underlying mechanism to acronyms.

If there’s a style for the given category, apply it.

```
5169 \define@key{glxtrabbrv}{category}{%
5170   \edef\glscategorylabel{#1}%
5171   \ifcsdef{@glxtr@current@#1}%
5172   {%
```

Warning should already have been issued.

```
5173   \let\@glxtr@orgwarndep\GlsXtrWarnDeprecatedAbbrStyle
5174   \let\GlsXtrWarnDeprecatedAbbrStyle\@gobbletwo
5175   \glxtr@applyabbrvstyle{\csname @glxtr@current@#1\endcsname}%
5176   \let\GlsXtrWarnDeprecatedAbbrStyle\@glxtr@orgwarndep
5177 }%
5178 {}%
5179 }
```

Save the short plural form. This may be needed before the entry is defined.

```
5180 \define@key{glxtrabbrv}{shortplural}{%
5181   \def\@glxtr@shortpl{#1}%
5182 }
```

Similarly for the long plural form.

```
5183 \define@key{glxtrabbrv}{longplural}{%
5184   \def\@glxtr@longpl{#1}%
5185 }
```

Token registers for the short plural and long plural, provided for use in the abbreviation style definitions.

\glsshortpltok

```
5186 \newtoks\glsshortpltok
```

\glslongpltok

```
5187 \newtoks\glslongpltok
```

`sxtr@insertdots` Provided in case user wants to automatically insert dots between each letter of the abbreviation. This should be applied before defining the abbreviation to optimise the document build. (Otherwise, it would have to be done each time the short form is required, which is an unnecessary waste of time.) For this to work the short form must be expanded when passed to `\newabbreviation`. Note that explicitly using the short or shortplural keys will override this.

```
5188 \newcommand*{\@glxstr@insertdots}[2]{%
5189   \def#1{}%
5190   \@glxstr@insert@dots#1#2\@nnil
5191 }
```

`xtr@insert@dots`

```
5192 \newcommand*{\@glxstr@insert@dots}[2]{%
5193   \ifx\@nnil#2\relax
5194   \let\@glxstr@insert@dots@next\@gobble
5195   \else
5196   \ifx\relax#2\relax
5197   \else
5198     \appto#1{#2.}%
5199   \fi
5200   \let\@glxstr@insert@dots@next\@glxstr@insert@dots
5201   \fi
5202   \@glxstr@insert@dots@next#1%
5203 }
```

Similarly provide a way of replacing spaces with `\glxstrwordsep`, which first needs to be defined:

`\glxstrwordsep`

```
5204 \newcommand*{\glxstrwordsep}{\space}
```

Each word is marked with

`\glxstrword`

```
5205 \newcommand*{\glxstrword}[1]{#1}
```

`tr@markwordseps`

```
5206 \newcommand*{\@glxstr@markwordseps}[2]{%
5207   \def#1{}%
5208   \@glxstr@mark@wordseps#1#2 \@nnil
5209 }
```

`r@mark@wordseps`

```
5210 \def\@glxstr@mark@wordseps#1#2 #3{%
5211   \ifdefempty{#1}%
5212   {\def#1{\protect\glxstrword{#2}}}%
5213   {\appto#1{\protect\glxstrwordsep\protect\glxstrword{#2}}}%
5214   \ifx\@nnil#3\relax
```

```

5215 \let\@glxstr@mark@wordseps@next\relax
5216 \else
5217 \def\@glxstr@mark@wordseps@next{%
5218 \@glxstr@mark@wordseps#1#3}%
5219 \fi
5220 \@glxstr@mark@wordseps@next
5221 }

```

**newabbreviation** Define a new generic abbreviation.

```

5222 \newcommand*\@newabbreviation[4][\@%
5223 \glxstr@newabbreviation{#1}{#2}{#3}{#4}%
5224 }

```

**newabbreviation** Internal macro. (bib2gls has an option that needs to temporarily redefine \@newabbreviation. This just makes it easier to save and restore the original definition.)

```

5225 \newcommand*\@glxstr@newabbreviation[4]{%
5226 \glskeylisttok{#1}%
5227 \glslabeltok{#2}%
5228 \glsshorttok{#3}%
5229 \glslongtok{#4}%

```

Save the original short and long values (before attribute settings modify them).

```

5230 \def\glxstrorgshort{#3}%
5231 \def\glxstrorglong{#4}%

```

Get the category.

```

5232 \def\glscategorylabel{abbreviation}%
5233 \glxstr@applyabbrvstyle{\@glsabbrv@current@abbreviation}%

```

Ignore the shortplural and longplural keys.

```

5234 \setkeys*\glxstrabbrv[shortplural,longplural]{#1}%

```

Set the default long plural

```

5235 \def\@glslongpl{#4\glspluralsuffix}%
5236 \let\@glslongpl\@glslongpl

```

Has the markwords attribute been set?

```

5237 \gl@ifcategoryattribute{\glscategorylabel}{markwords}{true}%
5238 {%
5239 \@glxstr@markwordseps\@glslong{#4}%
5240 \expandafter\def\expandafter\@glslongpl\expandafter
5241 {\@glslong\glspluralsuffix}%
5242 \let\@glslongpl\@glslongpl

```

Update \glslongtok.

```

5243 \expandafter\glslongtok\expandafter{\@glslong}%
5244 }%
5245 {}%

```

Has the markshortwords attribute been set? (Not compatible with insertdots.)

```

5246 \gl@ifcategoryattribute{\glscategorylabel}{markshortwords}{true}%
5247 {%

```

```

5248 \glxtr@keywordseps\@gls@short{#3}%
5249 }%
5250 {%

Has the insertdots attribute been set?
5251 \glsifcategoryattribute{\glscategorylabel}{insertdots}{true}%
5252 {%
5253 \glxtr@insertdots\@gls@short{#3}%
5254 \expandafter\glsshorttok\expandafter{\@gls@short\spacefactor1000 \relax}%
5255 }%
5256 {\def\@gls@short{#3}}%
5257 }%

Has the aposplural attribute been set? (Not compatible with noshortplural.)
5258 \glsifcategoryattribute{\glscategorylabel}{aposplural}{true}%
5259 {%
5260 \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
5261 '\abbrvpluralsuffix}%
5262 }%
5263 {%

Has the noshortplural attribute been set?
5264 \glsifcategoryattribute{\glscategorylabel}{noshortplural}{true}%
5265 {%
5266 \let\@gls@shortpl\@gls@short
5267 }%
5268 {%
5269 \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
5270 \abbrvpluralsuffix}%
5271 }%
5272 }%

Update \glsshorttok:
5273 \expandafter\glsshorttok\expandafter{\@gls@short}%

Hook for further customisation if required:
5274 \glxtrnewabbrevpresetkeyhook{#1}{#2}{#3}%

Get the short and long plurals provided by user in optional argument to override defaults, if
necessary. Ignore the category key (already obtained).
5275 \setkeys*{glxtrabbrev}[category]{#1}%

Has the plural been explicitly set?
5276 \ifx\@gls@default@longpl\@gls@longpl
5277 \else

Has the markwords attribute been set?
5278 \glsifcategoryattribute{\glscategorylabel}{markwords}{true}%
5279 {%
5280 \expandafter\glxtr@keywordseps\expandafter\@gls@longpl\expandafter
5281 {\@gls@longpl}%
5282 }%

```

```

5283     {}%
5284 \fi

```

Set the plural token registers so the values can be accessed by the abbreviation styles.

```

5285 \expandafter\glsshortpltok\expandafter{\@gls@shortpl}%
5286 \expandafter\glslongpltok\expandafter{\@gls@longpl}%

```

Do any extra setup provided by hook:

```

5287 \newabbreviationhook

```

Define this entry:

```

5288 \protected@edef\@do@newglossaryentry{%
5289   \noexpand\newglossaryentry{\the\glslabeltok}%
5290   {%
5291     type=\glstrabbrvtype,%
5292     category=abbreviation,%
5293     short={\the\glsshorttok},%
5294     shortplural={\the\glsshortpltok},%
5295     long={\the\glslongtok},%
5296     longplural={\the\glslongpltok},%
5297     name={\the\glsshorttok},%
5298     \CustomAbbreviationFields,%
5299     \the\glskeylisttok
5300   }%
5301 }%
5302 \@do@newglossaryentry
5303 \GlsXtrPostNewAbbreviation
5304 }

```

`evpresetkeyhook` Hook for extra stuff in `\newabbreviation`

```

5305 \newcommand*{\glstrnewabbspresetkeyhook}[3]{}

```

`NewAbbreviation` Hook used by abbreviation styles.

```

5306 \newcommand*{\GlsXtrPostNewAbbreviation}{}

```

`bbreviationhook` Hook for use with `\newabbreviation`.

```

5307 \newcommand*{\newabbreviationhook}{}

```

`reviationFields`

```

5308 \newcommand*{\CustomAbbreviationFields}{}

```

`\glstrparen` For the parenthetical styles.

```

5309 \newcommand*{\glstrparen}[1]{(#1)}

```

`lsxtrfullformat` Full format without case change.

```

5310 \newcommand*{\glsxtrfullformat}[2]{%
5311   \glsfirstlongfont{\glsaccesslong{#1}}#2\glsxtrfullsep{#1}%
5312   \glstrparen{\protect\glsfirstabbrvfont{\glsaccessshort{#1}}}%
5313 }

```

`\lsxtrfullformat` Full format with case change.

```
5314 \newcommand*{\Glsxtrfullformat}[2]{%
5315   \glsfirstlongfont{\Glsaccesslong{#1}}#2\gsxtrfullsep{#1}%
5316   \gsxtrparen{\protect\glsfirstabbrvfont{\Glsaccessshort{#1}}}%
5317 }
```

`\xtrfullplformat` Plural full format without case change.

```
5318 \newcommand*{\Glsxtrfullplformat}[2]{%
5319   \glsfirstlongfont{\Glsaccesslongpl{#1}}#2\gsxtrfullsep{#1}%
5320   \gsxtrparen{\protect\glsfirstabbrvfont{\Glsaccessshortpl{#1}}}%
5321 }
```

`\xtrfullplformat` Plural full format with case change.

```
5322 \newcommand*{\Glsxtrfullplformat}[2]{%
5323   \glsfirstlongfont{\Glsaccesslongpl{#1}}#2\gsxtrfullsep{#1}%
5324   \gsxtrparen{\protect\glsfirstabbrvfont{\Glsaccessshortpl{#1}}}%
5325 }
```

`\glxtrfullsep` Separator used by full format is a space by default. The argument is the entry's label.

```
5326 \newcommand*{\glxtrfullsep}[1]{\space}
```

In-line formats in case first use isn't compatible with `\glentryfull` (for example, first use suppresses the long form or uses a footnote).

`\nlinefullformat` Full format without case change.

```
5327 \newcommand*{\Glsxtrinlinefullformat}{\glxtrfullformat}
```

`\nlinefullformat` Full format with case change.

```
5328 \newcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}
```

`\xtrfullplformat` Plural full format without case change.

```
5329 \newcommand*{\Glsxtrinlinefullplformat}{\glxtrfullplformat}
```

`\inefullplformat` Plural full format with case change.

```
5330 \newcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}
```

Redefine `\glentryfull` etc to use the inline format. Since these commands as supposed to be expandable, they can only use the currently applied style. If there are mixed styles, you'll need to use the `\glxtrfull` set of commands instead.

`\glentryfull`

```
5331 \renewcommand*{\glentryfull}[1]{\Glsxtrinlinefullformat{#1}{}}
```

`\Glsentryfull`

```
5332 \renewcommand*{\Glsentryfull}[1]{\Glsxtrinlinefullformat{#1}{}}
```

`\glentryfullpl`

```
5333 \renewcommand*{\glentryfullpl}[1]{\Glsxtrinlinefullplformat{#1}{}}
```

`\Glsentryfullpl` 5334 `\renewcommand*{\Glsentryfullpl}[1]{\Glsxtrinlinefullplformat{#1}{}}`

`\gsfirstabbrvfont` Font changing command used for the abbreviation on first use or in the full format.  
5335 `\newcommand*{\gsfirstabbrvfont}[1]{\gsfirstabbrvdefaultfont{#1}}`

`\gsabbrvdefaultfont` Font changing command used for the abbreviation on first use or in the full format.  
5336 `\newcommand*{\gsfirstabbrvdefaultfont}[1]{\gsabbrvfont{#1}}`

`\gsabbrvfont` Font changing command used for the abbreviation on subsequent use.  
5337 `\newcommand*{\gsabbrvfont}[1]{\gsabbrvdefaultfont{#1}}`

`\gsabbrvdefaultfont`  
5338 `\newcommand*{\gsabbrvdefaultfont}[1]{#1}`

`\gslongfont` Font changing command used for the long form in commands like `\glsxtrlong`.  
5339 `\newcommand*{\gslongfont}[1]{\gslongdefaultfont{#1}}`

`\gslongdefaultfont` Default font changing command used for the long form in commands like `\glsxtrlong`.  
5340 `\newcommand*{\gslongdefaultfont}[1]{#1}`

`\gsfirstlongfont` Font changing command used for the long form on first use or in the full format.  
5341 `\newcommand*{\gsfirstlongfont}[1]{\gslongfont{#1}}`

`\gsfirstlongdefaultfont`  
5342 `\newcommand*{\gsfirstlongdefaultfont}[1]{\gslongdefaultfont{#1}}`

`\gsbrvpluralsuffix` Default plural suffix. Allow an alternative default suffix for abbreviations.  
5343 `\newcommand*{\gsxtrabbrvpluralsuffix}{\glspluralsuffix}`

`\gsbrvpluralsuffix` Default plural suffix.  
5344 `\newcommand*{\abbrvpluralsuffix}{\gsxtrabbrvpluralsuffix}`

`\glsxtrfull` Full form (no case-change).  
5345 `\newrobustcmd*{\glsxtrfull}{\@gls@hyp@opt\ns@glsxtrfull}`  
5346 `\newcommand*\ns@glsxtrfull[2][{}]{%`  
5347 `\new@ifnextchar[{\@glsxtr@full{#1}{#2}}{%`  
5348 `{\@glsxtr@full{#1}{#2}[]}%`  
5349 `}`

`\@glsxtr@full` Low-level macro:  
5350 `\def\@glsxtr@full#1#2[#3]{%`  
5351 `\glsdoifexists{#2}%`  
5352 `{%`  
5353 `\glssetabbrvfmt{\glscategory{#2}}%`  
5354 `\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper`  
5355 `\let\glsifplural\@secondoftwo`  
5356 `\let\glsupcase\@firstofthree`  
5357 `\let\glsinsert\@empty`  
5358 `\def\glscustomtext{\glsxtrinlinefullformat{#2}{#3}}%`

What should `\glxtrifwasfirstuse` be set to here? Where the inline and display full forms are the same, this is essentially emulating first use, to it make sense for the postlink hook to pretend it was a first use instance. It makes less sense if the inline and display forms are different. Provide a hook to make it easier to reconfigure.

```

5359 \glxtrsetupfulldefs
5360 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5361 }%
5362 \glspostlinkhook
5363 }

```

`trsetupfulldefs`

```

5364 \newcommand*\glxtrsetupfulldefs{%
5365 \let\glxtrifwasfirstuse\@firstoftwo
5366 }

```

`\Glsxtrfull` Full form (first letter uppercase).

```

5367 \newrobustcmd*\Glsxtrfull{\@gls@hyp@opt\@ns@Glsxtrfull}
5368 \newcommand*\ns@Glsxtrfull[2][{}]{%
5369 \new@ifnextchar[\@Glsxtr@full{#1}{#2}}%
5370 {\@Glsxtr@full{#1}{#2}[]}%
5371 }

```

`\@Glsxtr@full` Low-level macro:

```

5372 \def\@Glsxtr@full#1#2[#3]{%
5373 \glsdoifexists{#2}%
5374 {%
5375 \glssetabbrvfmt{\glscategory{#2}}%
5376 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5377 \let\glsifplural\@secondoftwo
5378 \let\glsupcase\@secondofthree
5379 \let\glsinsert\@empty
5380 \def\glscustomtext{\Glsxtrinlinefullformat{#2}{#3}}%
5381 \glxtrsetupfulldefs
5382 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5383 }%
5384 \glspostlinkhook
5385 }

```

`\GLSxtrfull` Full form (all uppercase).

```

5386 \newrobustcmd*\GLSxtrfull{\@gls@hyp@opt\@ns@GLSxtrfull}
5387 \newcommand*\ns@GLSxtrfull[2][{}]{%
5388 \new@ifnextchar[\@GLSxtr@full{#1}{#2}}%
5389 {\@GLSxtr@full{#1}{#2}[]}%
5390 }

```

`\@GLSxtr@full` Low-level macro:

```

5391 \def\@GLSxtr@full#1#2[#3]{%
5392 \glsdoifexists{#2}%

```



```

5393 {%
5394   \glssetabbrvfmt{\glscategory{#2}}%
5395   \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5396   \let\glsifplural\@secondoftwo
5397   \let\gls caps case\@thirdofthree
5398   \let\glsinsert\@empty
5399   \def\gls custom text{\mfirstucMakeUppercase{\glsxtr inline full format{#2}{#3}}}%
5400   \glsxtr setup full defs
5401   \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5402 }%
5403 \gls post link hook
5404 }

```

`\glsxtrfullpl` Plural full form (no case-change).

```

5405 \newrobustcmd*{\glsxtrfullpl}{\@gls@hyp@opt\@ns@glsxtrfullpl}
5406 \newcommand*\ns@glsxtrfullpl[2][{}]{%
5407   \new@ifnextchar[{\@glsxtr@fullpl{#1}{#2}}%
5408     {\@glsxtr@fullpl{#1}{#2}[]}%
5409 }

```

`\@glsxtr@fullpl` Low-level macro:

```

5410 \def\@glsxtr@fullpl#1#2[#3]{%
5411   \glsdoifexists{#2}%
5412   {%
5413     \glssetabbrvfmt{\glscategory{#2}}%
5414     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5415     \let\glsifplural\@firstoftwo
5416     \let\gls caps case\@firstofthree
5417     \let\glsinsert\@empty
5418     \def\gls custom text{\glsxtr inline full pl format{#2}{#3}}%
5419     \glsxtr setup full defs
5420     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5421   }%
5422   \gls post link hook
5423 }

```

`\Glsxtrfullpl` Plural full form (first letter uppercase).

```

5424 \newrobustcmd*{\Glsxtrfullpl}{\@gls@hyp@opt\@ns@Glsxtrfullpl}
5425 \newcommand*\ns@Glsxtrfullpl[2][{}]{%
5426   \new@ifnextchar[{\@Glsxtr@fullpl{#1}{#2}}%
5427     {\@Glsxtr@fullpl{#1}{#2}[]}%
5428 }

```

`\@Glsxtr@fullpl` Low-level macro:

```

5429 \def\@Glsxtr@fullpl#1#2[#3]{%
5430   \glsdoifexists{#2}%
5431   {%
5432     \glssetabbrvfmt{\glscategory{#2}}%
5433     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```

```

5434 \let\glsifplural\@firstoftwo
5435 \let\glsifscapscase\@secondofthree
5436 \let\glsinsert\@empty
5437 \def\glscustomtext{\Glsxtrinlinefullplformat{#2}{#3}}%
5438 \glsxtrsetupfulldefs
5439 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5440 }%
5441 \glspostlinkhook
5442 }

```

`\Glsxtrfullpl` Plural full form (all upper case).

```

5443 \newrobustcmd*{\Glsxtrfullpl}{\@gls@hyp@opt\@ns@Glsxtrfullpl}
5444 \newcommand*\ns@Glsxtrfullpl[2][]{%
5445 \new@ifnextchar[{\@Glsxtr@fullpl{#1}{#2}}%
5446 {\@Glsxtr@fullpl{#1}{#2}[]}%
5447 }

```

`\@Glsxtr@fullpl` Low-level macro:

```

5448 \def\@Glsxtr@fullpl#1#2[#3]{%
5449 \glsdoifexists{#2}%
5450 {%
5451 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5452 \let\glsifplural\@firstoftwo
5453 \let\glsifscapscase\@thirdofthree
5454 \let\glsinsert\@empty
5455 \def\glscustomtext{%
5456 \mfirstucMakeUppercase{\glsxtrinlinefullplformat{#2}{#3}}}%
5457 \glsxtrsetupfulldefs
5458 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5459 }%
5460 \glspostlinkhook
5461 }

```

The short and long forms work in a similar way to acronyms.

`\glsxtrshort`

```

5462 \newrobustcmd*{\glsxtrshort}{\@gls@hyp@opt\@ns@glsxtrshort}

```

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

```

5463 \newcommand*\ns@glsxtrshort[2][]{%
5464 \new@ifnextchar[{\@glsxtrshort{#1}{#2}}{\@glsxtrshort{#1}{#2}[]}%
5465 }

```

Read in the final optional argument:

```

5466 \def\@glsxtrshort#1#2[#3]{%
5467 \glsdoifexists{#2}%
5468 {%

```

Need to make sure `\glsabbrvfont` is set correctly.

```

5469 \glssetabbrvfmt{\glscategory{#2}}%

```

```

5470 \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
5471 \let\glxtrifwasfirstuse\@secondoftwo
5472 \let\gl@sifplural\@secondoftwo
5473 \let\glscapscase\@firstofthree
5474 \let\gl$insert\@empty
5475 \def\glscustomtext{%
5476   \gl@sabbrvfont{\gl@saccessshort{#2}\ifglxtrininsertinside#3\fi}%
5477   \ifglxtrininsertinside\else#3\fi
5478 }%
5479 \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5480 }%
5481 \glspostlinkhook
5482 }

```

\Glsxtrshort

```

5483 \newrobustcmd*{\Glsxtrshort}{\@gl@hyp@opt\ns@Glsxtrshort}
    Define the un-starred form. Need to determine if there is a final optional argument
5484 \newcommand*{\ns@Glsxtrshort}[2][\glscapscase]{%
5485   \new@ifnextchar[\@Glsxtrshort{#1}{#2}]{\@Glsxtrshort{#1}{#2}[\glscapscase]}%
5486 }
    Read in the final optional argument:
5487 \def\@Glsxtrshort#1#2[#3]{%
5488   \gl@sdoifexists{#2}%
5489   {%
5490     \gl@ssetabbrvfmt{\glscategory{#2}}%
5491     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
5492     \let\glxtrifwasfirstuse\@secondoftwo
5493     \let\gl@sifplural\@secondoftwo
5494     \let\glscapscase\@secondofthree
5495     \let\gl$insert\@empty
5496     \def\glscustomtext{%
5497       \gl@sabbrvfont{\Glsaccessshort{#2}\ifglxtrininsertinside#3\fi}%
5498       \ifglxtrininsertinside\else#3\fi
5499     }%
5500     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5501   }%
5502   \glspostlinkhook
5503 }

```

\GLSxtrshort

```

5504 \newrobustcmd*{\GLSxtrshort}{\@gl@hyp@opt\ns@GLSxtrshort}
    Define the un-starred form. Need to determine if there is a final optional argument
5505 \newcommand*{\ns@GLSxtrshort}[2][\glscapscase]{%
5506   \new@ifnextchar[\@GLSxtrshort{#1}{#2}]{\@GLSxtrshort{#1}{#2}[\glscapscase]}%
5507 }
    Read in the final optional argument:
5508 \def\@GLSxtrshort#1#2[#3]{%

```

```

5509 \glsdoifexists{#2}%
5510 {%
5511   \glssetabbrvfmt{\glscategory{#2}}%
5512   \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
5513   \let\glxtrifwasfirstuse\@secondoftwo
5514   \let\glsifplural\@secondoftwo
5515   \let\glscapscase\@thirdofthree
5516   \let\glsinsert\@empty
5517   \def\glscustomtext{%
5518     \mfirstucMakeUppercase
5519     {\glsabbrvfont{\glsaccessshort{#2}\ifglxtrininsertinside#3\fi}%
5520     \ifglxtrininsertinside\else#3\fi
5521   }%
5522   }%
5523   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5524 }%
5525 \glspostlinkhook
5526 }

```

`\glxtrlong`

```

5527 \newrobustcmd*{\glxtrlong}{\@gl@hyp@opt\@ns@glxtrlong}
    Define the un-starred form. Need to determine if there is a final optional argument
5528 \newcommand*{\ns@glxtrlong}[2] []{%
5529   \new@ifnextchar[{\@glxtrlong{#1}{#2}}{\@glxtrlong{#1}{#2} []}%
5530 }

    Read in the final optional argument:
5531 \def\@glxtrlong#1#2[#3]{%
5532   \glsdoifexists{#2}%
5533   {%
5534     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
5535     \let\glxtrifwasfirstuse\@secondoftwo
5536     \let\glsifplural\@secondoftwo
5537     \let\glscapscase\@firstofthree
5538     \let\glsinsert\@empty
5539     \def\glscustomtext{%
5540       \glsfont{\glsaccesslong{#2}\ifglxtrininsertinside#3\fi}%
5541       \ifglxtrininsertinside\else#3\fi
5542     }%
5543     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5544   }%
5545   \glspostlinkhook
5546 }

```

`\Glsxtrlong`

```

5547 \newrobustcmd*{\Glsxtrlong}{\@gl@hyp@opt\@ns@Glsxtrlong}
    Define the un-starred form. Need to determine if there is a final optional argument
5548 \newcommand*{\ns@Glsxtrlong}[2] []{%

```

```

5549 \new@ifnextchar[{\@Glsxtrlong{#1}{#2}}{\@Glsxtrlong{#1}{#2} []}%
5550 }

```

Read in the final optional argument:

```

5551 \def\@Glsxtrlong#1#2[#3]{%
5552   \glsdoifexists{#2}%
5553   {%
5554     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
5555     \let\glxtrifwasfirstuse\@secondoftwo
5556     \let\gl@sifplural\@secondoftwo
5557     \let\glscapscase\@secondofthree
5558     \let\gl$insert\@empty
5559     \def\glscustomtext{%
5560       \glslongfont{\gl@saccesslong{#2}\ifglxtrininsertinside#3\fi}%
5561       \ifglxtrininsertinside\else#3\fi
5562     }%
5563     \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5564   }%
5565   \glspostlinkhook
5566 }

```

\GLSxtrlong

```

5567 \newrobustcmd*{\GLSxtrlong}{\@gl@hyp@opt\@ns@GLSxtrlong}

```

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

```

5568 \newcommand*{\ns@GLSxtrlong}[2] [] {%
5569   \new@ifnextchar[{\@GLSxtrlong{#1}{#2}}{\@GLSxtrlong{#1}{#2} []}%
5570 }

```

Read in the final optional argument:

```

5571 \def\@GLSxtrlong#1#2[#3]{%
5572   \glsdoifexists{#2}%
5573   {%
5574     \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
5575     \let\glxtrifwasfirstuse\@secondoftwo
5576     \let\gl@sifplural\@secondoftwo
5577     \let\glscapscase\@thirdofthree
5578     \let\gl$insert\@empty
5579     \def\glscustomtext{%
5580       \mfirstucMakeUppercase
5581       {\glslongfont{\gl@saccesslong{#2}\ifglxtrininsertinside#3\fi}%
5582       \ifglxtrininsertinside\else#3\fi
5583     }%
5584   }%
5585   \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5586 }%
5587 \glspostlinkhook
5588 }

```

Plural short forms:

\glxtrshortpl

```
5589 \newrobustcmd*{\glxtrshortpl}{\@glx@hyp@opt\ns@glxtrshortpl}
```

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

```
5590 \newcommand*{\ns@glxtrshortpl}[2] [] {%
```

```
5591 \new@ifnextchar[{\@glsxtrshortpl{#1}{#2}}{\@glsxtrshortpl{#1}{#2}[]}%
```

5592 }

Read in the final optional argument:

```
5593 \def\@glxtrshortpl#1#2[#3]{%
```

5594 \glsdoifexists{#2}%

5595 { %

5596 \glssetabbrvfmt{\glscategory{#2}}%

```
5597 \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
```

```
5598 \let\glxtrifwasfirstuse\@secondoftwo
```

5599 \let\glsifplural\@firstoftwo

5600 \let\glscapscase\@firstofthree

```
5601 \let\glsinsert\@empty
```

```
5602 \def\glscustomtext{%
```

5603 \glsabbrvfont{\glsaccessshortpl{#2}\ifglxtrinsertinside#3\fi}%

5604 \ifglxtrinsertinside\else#3\fi

5605 }%

```
5606 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
```

5607 }%

5608 \glspostlinkhook

5609 }

\Glsxtrshortpl

```
5610 \newrobustcmd*{\Glsxtrshortpl}{\@gls@hyp@opt\ns@Glsxtrshortpl}
```

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

```
5611 \newcommand*{\ns@Glsxtrshortpl}[2] [] {%
```

```
5612 \new@ifnextchar[{\@Glsxtrshortpl{#1}{#2}}{\@Glsxtrshortpl{#1}{#2}[]}%
```

5613 }

Read in the final optional argument:

```
5614 \def\@Glsxtrshortpl#1#2[#3]{%
```

5615 \glsdoifexists{#2}%

5616 { %

5617 \glssetabbrvfmt{\glscategory{#2}}%

```
5618 \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
```

5619 \let\glxtrifwasfirstuse\@secondoftwo

5620 \let\glsifplural\@firstoftwo

5621 \let\glscapscase\@secondofthree

```
5622 \let\glsinsert\@empty
```

```
5623 \def\glscustomtext{%
```

5624 \glsabbrvfont{\Glsaccessshortpl{#2}\ifglxtrinsertinside#3\fi}%

5625 \ifglxtrinsertinside\else#3\fi

5626 }%

5627 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%

5628 }%

```

5629 \glspostlinkhook
5630 }

```

\GLSxtrshortpl

```

5631 \newrobustcmd*{\GLSxtrshortpl}{\@gls@hyp@opt\@ns@GLSxtrshortpl}
    Define the un-starred form. Need to determine if there is a final optional argument
5632 \newcommand*{\ns@GLSxtrshortpl}[2] [] {%
5633   \new@ifnextchar[{\@GLSxtrshortpl{#1}{#2}}{\@GLSxtrshortpl{#1}{#2} []}%
5634 }

    Read in the final optional argument:
5635 \def\@GLSxtrshortpl#1#2[#3] {%
5636   \glsdoidexists{#2}%
5637   {%
5638     \glsetabbrvfmt{\gls@category{#2}}%
5639     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5640     \let\glsxtrifwasfirstuse\@secondoftwo
5641     \let\glsifplural\@firstoftwo
5642     \let\gls@caps@case\@thirdofthree
5643     \let\glsinsert\@empty
5644     \def\gls@customtext{%
5645       \mfirstucMakeUppercase
5646       {\glsabbrvfont{\gls@accessshortpl{#2}}\ifglsxtrininsertinside#3\fi}%
5647       \ifglsxtrininsertinside\else#3\fi
5648     }%
5649   }%
5650   \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
5651 }%
5652 \glspostlinkhook
5653 }

```

Plural long forms:

\glsxtrlongpl

```

5654 \newrobustcmd*{\glsxtrlongpl}{\@gls@hyp@opt\@ns@glsxtrlongpl}
    Define the un-starred form. Need to determine if there is a final optional argument
5655 \newcommand*{\ns@glsxtrlongpl}[2] [] {%
5656   \new@ifnextchar[{\@glsxtrlongpl{#1}{#2}}{\@glsxtrlongpl{#1}{#2} []}%
5657 }

    Read in the final optional argument:
5658 \def\@glsxtrlongpl#1#2[#3] {%
5659   \glsdoidexists{#2}%
5660   {%
5661     \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5662     \let\glsxtrifwasfirstuse\@secondoftwo
5663     \let\glsifplural\@firstoftwo
5664     \let\gls@caps@case\@firstofthree
5665     \let\glsinsert\@empty

```

```

5666 \def\glscustomtext{%
5667 \glslongfont{\glssaccesslongpl{#2}\ifglxtrinsertinside#3\fi}%
5668 \ifglxtrinsertinside\else#3\fi
5669 }%
5670 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5671 }%
5672 \glspostlinkhook
5673 }

```

\Glsxtrlongpl

```

5674 \newrobustcmd*{\Glsxtrlongpl}{\@gls@hyp@opt\@ns@Glsxtrlongpl}
    Define the un-starred form. Need to determine if there is a final optional argument
5675 \newcommand*{\ns@Glsxtrlongpl}[2] [] {%
5676 \new@ifnextchar[{\@Glsxtrlongpl{#1}{#2}}{\@Glsxtrlongpl{#1}{#2} []}%
5677 }
    Read in the final optional argument:
5678 \def\@Glsxtrlongpl#1#2[#3]{%
5679 \glsdofexists{#2}%
5680 {%
5681 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5682 \let\glxtrifwasfirstuse\@secondoftwo
5683 \let\glsifplural\@firstoftwo
5684 \let\glscapscase\@secondofthree
5685 \let\glsinsert\@empty
5686 \def\glscustomtext{%
5687 \glslongfont{\glssaccesslongpl{#2}\ifglxtrinsertinside#3\fi}%
5688 \ifglxtrinsertinside\else#3\fi
5689 }%
5690 \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5691 }%
5692 \glspostlinkhook
5693 }

```

\GLSxtrlongpl

```

5694 \newrobustcmd*{\GLSxtrlongpl}{\@gls@hyp@opt\@ns@GLSxtrlongpl}
    Define the un-starred form. Need to determine if there is a final optional argument
5695 \newcommand*{\ns@GLSxtrlongpl}[2] [] {%
5696 \new@ifnextchar[{\@GLSxtrlongpl{#1}{#2}}{\@GLSxtrlongpl{#1}{#2} []}%
5697 }
    Read in the final optional argument:
5698 \def\@GLSxtrlongpl#1#2[#3]{%
5699 \glsdofexists{#2}%
5700 {%
5701 \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
5702 \let\glxtrifwasfirstuse\@secondoftwo
5703 \let\glsifplural\@firstoftwo
5704 \let\glscapscase\@thirdofthree

```



```

5705 \let\glsinsert\@empty
5706 \def\glscustomtext{%
5707     \mfirstucMakeUppercase
5708     {\glslongfont{\glsaccesslongpl{#2}\ifglxtrinsertinside#3\fi}%
5709     \ifglxtrinsertinside\else#3\fi
5710     }%
5711     }%
5712     \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
5713 }%
5714 \glspostlinkhook
5715 }

```

`\glssetabbrvfmt` Set the current format for the given category (or the abbreviation category if unset).

```

5716 \newcommand*{\glssetabbrvfmt}[1]{%
5717     \ifcsdef{@glsabbrv@current@#1}%
5718     {\glxtr@applyabbrvfmt{\csname @glsabbrv@current@#1\endcsname}}%
5719     {\glxtr@applyabbrvfmt{\@glsabbrv@current@abbreviation}}%
5720 }

```

`sxtrgenabbrvfmt` Similar to `\glsgenacfmt`, but for abbreviations.

```

5721 \newcommand*{\glxtrgenabbrvfmt}{%
5722     \ifdefempty\glscustomtext
5723     {%
5724         \ifglused\glslabel
5725         {%

```

Subsequent use:

```

5726         \glsifplural
5727         {%

```

Subsequent plural form:

```

5728         \glscapscase
5729         {%

```

Subsequent plural form, don't adjust case:

```

5730         \glxtrsubsequentplfmt{\glslabel}{\glsinsert}%
5731         }%
5732         {%

```

Subsequent plural form, make first letter upper case:

```

5733         \Glsxtrsubsequentplfmt{\glslabel}{\glsinsert}%
5734         }%
5735         {%

```

Subsequent plural form, all caps:

```

5736         \mfirstucMakeUppercase
5737         {\glxtrsubsequentplfmt{\glslabel}{\glsinsert}}%
5738         }%
5739         }%
5740         {%

```

Subsequent singular form

```
5741      \glscapscase  
5742      {%
```

Subsequent singular form, don't adjust case:

```
5743      \glxtrsubsequentfmt{\glslabel}{\glsinsert}%  
5744      }%  
5745      {%
```

Subsequent singular form, make first letter upper case:

```
5746      \Glsxtrsubsequentfmt{\glslabel}{\glsinsert}%  
5747      }%  
5748      {%
```

Subsequent singular form, all caps:

```
5749      \mfirstucMakeUppercase  
5750      {\glxtrsubsequentfmt{\glslabel}{\glsinsert}}%  
5751      }%  
5752      }%  
5753      }%  
5754      {%
```

First use:

```
5755      \glsifplural  
5756      {%
```

First use plural form:

```
5757      \glscapscase  
5758      {%
```

First use plural form, don't adjust case:

```
5759      \glxtrfullplformat{\glslabel}{\glsinsert}%  
5760      }%  
5761      {%
```

First use plural form, make first letter upper case:

```
5762      \Glsxtrfullplformat{\glslabel}{\glsinsert}%  
5763      }%  
5764      {%
```

First use plural form, all caps:

```
5765      \mfirstucMakeUppercase  
5766      {\glxtrfullplformat{\glslabel}{\glsinsert}}%  
5767      }%  
5768      }%  
5769      {%
```

First use singular form

```
5770      \glscapscase  
5771      {%
```

First use singular form, don't adjust case:

```
5772      \glxtrfullformat{\glslabel}{\glsinsert}%
```

```

5773     }%
5774     {%

```

First use singular form, make first letter upper case:

```

5775     \Glsxtrfullformat{\glslabel}{\glsinsert}%
5776     }%
5777     {%

```

First use singular form, all caps:

```

5778     \mfirstucMakeUppercase
5779     {\glsxtrfullformat{\glslabel}{\glsinsert}}%
5780     }%
5781     }%
5782     }%
5783     }%
5784     {%

```

User supplied text.

```

5785     \glscustomtext
5786     }%
5787 }

```

trsubsequentfmt Subsequent use format (singular no case change).

```

5788 \newcommand*{\glsxtrsubsequentfmt}[2]{%
5789   \glsabbrvfont{\glsaccessshort{#1}\ifglsxtrininsertinside #2\fi}%
5790   \ifglsxtrininsertinside \else#2\fi
5791 }
5792 \let\glsxtrdefaultsubsequentfmt\glsxtrsubsequentfmt

```

subsequentplfmt Subsequent use format (plural no case change).

```

5793 \newcommand*{\glsxtrsubsequentplfmt}[2]{%
5794   \glsabbrvfont{\glsaccessshortpl{#1}\ifglsxtrininsertinside #2\fi}%
5795   \ifglsxtrininsertinside \else#2\fi
5796 }
5797 \let\glsxtrdefaultsubsequentplfmt\glsxtrsubsequentplfmt

```

trsubsequentfmt Subsequent use format (singular, first letter uppercase).

```

5798 \newcommand*{\Glsxtrsubsequentfmt}[2]{%
5799   \glsabbrvfont{\Glsaccessshort{#1}\ifglsxtrininsertinside #2\fi}%
5800   \ifglsxtrininsertinside \else#2\fi
5801 }
5802 \let\Glsxtrdefaultsubsequentfmt\Glsxtrsubsequentfmt

```

subsequentplfmt Subsequent use format (plural, first letter uppercase).

```

5803 \newcommand*{\Glsxtrsubsequentplfmt}[2]{%
5804   \glsabbrvfont{\Glsaccessshortpl{#1}\ifglsxtrininsertinside #2\fi}%
5805   \ifglsxtrininsertinside \else#2\fi
5806 }
5807 \let\Glsxtrdefaultsubsequentplfmt\Glsxtrsubsequentplfmt

```

## 1.6.1 Abbreviation Styles Setup

`\abbreviationstyle`

```
5808 \newcommand*{\setabbreviationstyle}[2][abbreviation]{%
5809   \ifcsundef{@glsabbrv@dispstyle@setup@#2}%
5810   {%
5811     \PackageError{glossaries-extra}{Undefined abbreviation style '#2'}{}}%
5812   }%
5813   {%
```

Have abbreviations already been defined for this category?

```
5814   \ifcsstring{@glsabbrv@current@#1}{#2}%
5815   {%
```

Style already set.

```
5816   }%
5817   {%
5818     \def\@glsxtr@dostylewarn{%
5819       \glsforeachincategory{#1}{\@gls@type}{\@gls@label}%
5820       {%
5821         \def\@glsxtr@dostylewarn{\GlossariesWarning{Abbreviation
5822           style has been switched \MessageBreak
5823           for category '#1', \MessageBreak
5824           but there have already been entries \MessageBreak
5825           defined for this category. Unwanted \MessageBreak
5826           side-effects may result}}}%
5827       \@endfortrue
5828     }%
5829     \@glsxtr@dostylewarn
```

Set up the style for the given category.

```
5830     \csdef{@glsabbrv@current@#1}{#2}%
5831     \glsxtr@applyabbrvstyle{#2}%
5832   }%
5833 }%
5834 }
```

`\applyabbrvstyle` Apply the abbreviation style without existence check.

```
5835 \newcommand*{\glsxtr@applyabbrvstyle}[1]{%
5836   \csuse{@glsabbrv@dispstyle@setup@#1}%
5837   \csuse{@glsabbrv@dispstyle@fmts@#1}%
5838 }
```

`\r@applyabbrvfmt` Only apply the style formats.

```
5839 \newcommand*{\glsxtr@applyabbrvfmt}[1]{%
5840   \csuse{@glsabbrv@dispstyle@fmts@#1}%
5841 }
```

`\abbreviationstyle` This is different from `\newacronymstyle`. The first argument is the label, the second argument sets the information required when defining the new abbreviation and the third argument sets the commands used to display the full format.

```

5842 \newcommand*{\newabbreviationstyle}[3]{%
5843   \ifcsdef{@glsabbrv@dispstyle@setup@#1}
5844   {%
5845     \PackageError{glossaries-extra}{Abbreviation style ‘#1’ already
5846       defined}{}%
5847   }%
5848   {%
5849     \csdef{@glsabbrv@dispstyle@setup@#1}{%
      Initialise hook to do nothing. The style may change this.
5850       \renewcommand*{\GlsXtrPostNewAbbreviation}{}%
5851       #2}%
5852     \csdef{@glsabbrv@dispstyle@fmts@#1}{%
      Assume in-line form is the same as first use. The style may change this.
5853       \renewcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}%
5854       \renewcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}%
5855       \renewcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}%
5856       \renewcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}%
      Reset \glsxtrsubsequentfmt etc in case a style changes this.
5857       \let\glsxtrsubsequentfmt\glsxtrdefaultsubsequentfmt
5858       \let\glsxtrsubsequentplfmt\glsxtrdefaultsubsequentplfmt
5859       \let\Glsxtrsubsequentfmt\Glsxtrdefaultsubsequentfmt
5860       \let\Glsxtrsubsequentplfmt\Glsxtrdefaultsubsequentplfmt
5861       #3}%
5862     }%
5863 }

```

breivationstyle

```

5864 \newcommand*{\renewabbreviationstyle}[3]{%
5865   \ifcsundef{@glsabbrv@dispstyle@setup@#1}
5866   {%
5867     \PackageError{glossaries-extra}{Abbreviation style ‘#1’ not defined}{}%
5868   }%
5869   {%
5870     \csdef{@glsabbrv@dispstyle@setup@#1}{%
      Initialise hook to do nothing. The style may change this.
5871       \renewcommand*{\GlsXtrPostNewAbbreviation}{}%
5872       #2}%
5873     \csdef{@glsabbrv@dispstyle@fmts@#1}{%
      Assume in-line form is the same as first use. The style may change this.
5874       \renewcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}%
5875       \renewcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}%
5876       \renewcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}%
5877       \renewcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}%
5878       #3}%
5879     }%
5880 }

```

abbreviationstyle Define a synonym for an abbreviation style. The first argument is the new name. The second argument is the original style's name.

```
5881 \newcommand*{\letabbreviationstyle}[2]{%
5882   \csletcs{@glsabbrv@dispstyle@setup@#1}{@glsabbrv@dispstyle@setup@#2}%
5883   \csletcs{@glsabbrv@dispstyle@fmts@#1}{@glsabbrv@dispstyle@fmts@#2}%
5884 }
```

deprecated@abbrstyle `\@glsxtr@deprecated@abbrstyle{<old-name>}{<new-name>}`

Define a synonym for a deprecated abbreviation style.

```
5885 \newcommand*{\@glsxtr@deprecated@abbrstyle}[2]{%
5886   \csdef{@glsabbrv@dispstyle@setup@#1}{%
5887     \GlsXtrWarnDeprecatedAbbrStyle{#1}{#2}%
5888     \csuse{@glsabbrv@dispstyle@setup@#2}%
5889   }%
5890   \csletcs{@glsabbrv@dispstyle@fmts@#1}{@glsabbrv@dispstyle@fmts@#2}%
5891 }
```

deprecatedAbbrStyle Generate warning for deprecated style use.

```
5892 \newcommand*{\GlsXtrWarnDeprecatedAbbrStyle}[2]{%
5893   \GlossariesExtraWarning{Deprecated abbreviation style name ‘#1’,
5894     use ‘#2’ instead}%
5895 }
```

useAbbrStyleSetup

```
5896 \newcommand*{\GlsXtrUseAbbrStyleSetup}[1]{%
5897   \ifcsundef{@glsabbrv@dispstyle@setup@#1}%
5898   {%
5899     \PackageError{glossaries-extra}%
5900     {Unknown abbreviation style definitions ‘#1’}{}%
5901   }%
5902   {%
5903     \csname @glsabbrv@dispstyle@setup@#1\endcsname
5904   }%
5905 }
```

useAbbrStyleFmt

```
5906 \newcommand*{\GlsXtrUseAbbrStyleFmt}[1]{%
5907   \ifcsundef{@glsabbrv@dispstyle@fmts@#1}%
5908   {%
5909     \PackageError{glossaries-extra}%
5910     {Unknown abbreviation style formats ‘#1’}{}%
5911   }%
5912   {%
5913     \csname @glsabbrv@dispstyle@fmts@#1\endcsname
5914   }%
5915 }
```

## 1.6.2 Predefined Styles (Default Font)

Define some common styles. These will set the first, firstplural, text and plural keys, even if the regular attribute isn't set to "true". If this attribute is set, commands like `\gls` will use them as per a regular entry, otherwise those keys will be ignored unless explicitly invoked by the user with commands like `\glsfirst`. In order for the first letter uppercase versions to work correctly, `\glstrfullformat` needs to be expanded when those keys are set. The final optional argument of `\glsfirst` will behave differently to the final optional argument of `\gls` with some styles.

`xtrinsertinside` Switch to determine if the insert text should be inside or outside the font changing command. The default is outside.

```
5916 \newif\ifglstrinsertinside
5917 \glstrinsertinsidefalse
```

`long-short`

```
5918 \newabbreviationstyle{long-short}%
5919 {%
5920   \renewcommand*{\CustomAbbreviationFields}{%
5921     name={\protect\glsabbrvfont{\the\glsshorttok}},
5922     sort={\the\glsshorttok},
5923     first={\protect\glsfirstlongfont{\the\glslongtok}%
5924       \protect\glstrfullsep{\the\glslabeltok}%
5925       \glstrparen{\protect\glsfirstabbrvfont{\the\glsshorttok}}},%
5926     firstplural={\protect\glsfirstlongfont{\the\glslongpltok}%
5927       \protect\glstrfullsep{\the\glslabeltok}%
5928       \glstrparen{\protect\glsfirstabbrvfont{\the\glsshortpltok}}},%
5929     plural={\protect\glsabbrvfont{\the\glsshortpltok}}},%
5930     description={\the\glslongtok}}%
```

Unset the regular attribute if it has been set.

```
5931 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
5932   \glshasattribute{\the\glslabeltok}{regular}%
5933   {%
5934     \glissetattribute{\the\glslabeltok}{regular}{false}%
5935   }%
5936   {}%
5937 }%
5938 }%
5939 {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
5940 \renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
5941 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
5942 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
5943 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
5944 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline form are the same for this style.

```

5945 \renewcommand*{\glxtrfullformat}[2]{%
5946   \glsfirstlongfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
5947   \ifglxtrinsertinside\else##2\fi
5948   \glxtrfullsep{##1}%
5949   \glxtrparen{\glsfirstabbrvfont{\glsaccessshort{##1}}}%
5950 }%
5951 \renewcommand*{\glxtrfullplformat}[2]{%
5952   \glsfirstlongfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
5953   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
5954   \glxtrparen{\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
5955 }%
5956 \renewcommand*{\Glsxtrfullformat}[2]{%
5957   \glsfirstlongfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
5958   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
5959   \glxtrparen{\glsfirstabbrvfont{\glsaccessshort{##1}}}%
5960 }%
5961 \renewcommand*{\Glsxtrfullplformat}[2]{%
5962   \glsfirstlongfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
5963   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
5964   \glxtrparen{\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
5965 }%
5966 }

```

Set this as the default style for general abbreviations:

```

5967 \setabbreviationstyle{long-short}

```

ngshortdescsort

```

5968 \newcommand*{\glxtrlongshortdescsort}{%
5969   \expandonce\glxtrorglongspace (\expandonce\glxtrorgshort)%
5970 }

```

ngshortdescname

```

5971 \newcommand*{\glxtrlongshortdescname}{%
5972   \protect\glslongfont{\the\glslongtok}
5973   \glxtrparen{\protect\glsabbrvfont{\the\glsshorttok}}%
5974 }

```

long-short-desc User supplies description. The long form is included in the name.

```

5975 \newabbreviationstyle{long-short-desc}%
5976 {%
5977   \renewcommand*{\CustomAbbreviationFields}{%
5978     name={\glxtrlongshortdescname},
5979     sort={\glxtrlongshortdescsort},%
5980     first={\protect\glsfirstlongfont{\the\glslongtok}%
5981       \protect\glxtrfullsep{\the\glslabeltok}%
5982       \glxtrparen{\protect\glsfirstabbrvfont{\the\glsshorttok}}},%
5983     firstplural={\protect\glsfirstlongfont{\the\glslongtok}%
5984       \protect\glxtrfullsep{\the\glslabeltok}%

```



```
5985 \glstrparen{\protect\glsfirstabbrvfont{\the\glsshortpltok}}},%
```

The text key should only have the short form.

```
5986 text={\protect\glsabbrvfont{\the\glsshorttok}},%
```

```
5987 plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
5988 }%
```

Unset the regular attribute if it has been set.

```
5989 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
5990 \glshasattribute{\the\glslabeltok}{regular}%
5991 {%
5992 \glissetattribute{\the\glslabeltok}{regular}{false}%
5993 }%
5994 {}%
5995 }%
5996 }%
5997 {%
5998 \GlsXtrUseAbbrStyleFmts{long-short}%
5999 }
```

**short-long** Short form followed by long form in parenthesis on first use.

```
6000 \newabbreviationstyle{short-long}%
6001 {%
6002 \renewcommand*{\CustomAbbreviationFields}{%
6003 name={\protect\glsabbrvfont{\the\glsshorttok}},
6004 sort={\the\glsshorttok},
6005 description={\the\glslongtok},%
6006 first={\protect\glsfirstabbrvfont{\the\glsshorttok}}%
6007 \protect\glstrfullsep{\the\glslabeltok}%
6008 \glstrparen{\protect\glsfirstlongfont{\the\glslongtok}}},%
6009 firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}}%
6010 \protect\glstrfullsep{\the\glslabeltok}%
6011 \glstrparen{\protect\glsfirstlongfont{\the\glslongpltok}}},%
6012 plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%

```

Unset the regular attribute if it has been set.

```
6013 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6014 \glshasattribute{\the\glslabeltok}{regular}%
6015 {%
6016 \glissetattribute{\the\glslabeltok}{regular}{false}%
6017 }%
6018 {}%
6019 }%
6020 }%
6021 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```
6022 \renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
6023 \renewcommand*{\glsabbrvfont[1]}{\glsabbrvdefaultfont{##1}}%
```

```

6024 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
6025 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6026 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

6027 \renewcommand*{\glsxtrfullformat}[2]{%
6028   \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
6029   \ifglsxtrininsertinside\else##2\fi
6030   \glsxtrfullsep{##1}%
6031   \glsxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
6032 }%
6033 \renewcommand*{\glsxtrfullplformat}[2]{%
6034   \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
6035   \ifglsxtrininsertinside\else##2\fi
6036   \glsxtrfullsep{##1}%
6037   \glsxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
6038 }%
6039 \renewcommand*{\Glsxtrfullformat}[2]{%
6040   \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
6041   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
6042   \glsxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
6043 }%
6044 \renewcommand*{\Glsxtrfullplformat}[2]{%
6045   \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
6046   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
6047   \glsxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
6048 }%
6049 }

```

ortlongdescsort

```

6050 \newcommand*{\glsxtrshortlongdescsort}{\the\glsshorttok}

```

ortlongdescname

```

6051 \newcommand*{\glsxtrshortlongdescname}{%
6052   \protect\glsabbrvfont{\the\glsshorttok}
6053   \glsxtrparen{\protect\glslongfont{\the\glslongtok}}}%
6054 }

```

short-long-desc User supplies description. The long form is included in the name.

```

6055 \newabbreviationstyle{short-long-desc}%
6056 {%
6057   \renewcommand*{\CustomAbbreviationFields}{%
6058     name={\glsxtrshortlongdescname},
6059     sort={\glsxtrshortlongdescsort},
6060     first={\protect\glsfirstabbrvfont{\the\glsshorttok}}%
6061     \protect\glsxtrfullsep{\the\glslabeltok}}%
6062     \glsxtrparen{\protect\glsfirstlongfont{\the\glslongtok}}},%
6063     firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}}%
6064     \protect\glsxtrfullsep{\the\glslabeltok}}%
6065     \glsxtrparen{\protect\glsfirstlongfont{\the\glslongpltok}}},%

```

```

6066     text={\protect\glsabbrvfont{\the\glsshorttok}},%
6067     plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
6068 }%

Unset the regular attribute if it has been set.
6069 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6070   \glshasattribute{\the\glslabeltok}{regular}%
6071   {%
6072     \glissetattribute{\the\glslabeltok}{regular}{false}%
6073   }%
6074   {}%
6075 }%
6076 }%
6077 {%
6078   \GlsXtrUseAbbrStyleFmts{short-long}%
6079 }

```

ongfootnotefont Only used by the “footnote” styles.

```
6080 \newcommand*{\glsfirstlongfootnotefont}[1]{\glslongfootnotefont{#1}}%
```

ongfootnotefont Only used by the “footnote” styles.

```
6081 \newcommand*{\glslongfootnotefont}[1]{\glslongdefaultfont{#1}}%
```

xtrabbrvfootnote `\glsxtrabbrvfootnote{<label>}{<long>}`

Command used by footnote abbreviation styles. The default definition ignores the first argument. The second argument *<long>* includes the font changing command and may be the singular or plural form, depending on the command that was used (for example, `\gls` or `\glspl`).

```
6082 \newcommand*{\glsxtrabbrvfootnote}[2]{\footnote{#2}}
```

footnote Short form followed by long form in footnote on first use.

```

6083 \newabbreviationstyle{footnote}%
6084 {%
6085   \renewcommand*{\CustomAbbreviationFields}{%
6086     name={\protect\glsabbrvfont{\the\glsshorttok}},
6087     sort={\the\glsshorttok},
6088     description={\the\glslongtok},%

6089     first={\protect\glsfirstabbrvfont{\the\glsshorttok}}%
6090     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
6091     {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
6092     firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}}%
6093     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
6094     {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%

```

6095 plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the regular attribute if it has been set.

```
6096 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6097   \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
6098   \glsattribute{\the\glslabeltok}{regular}%
6099   {%
6100     \glssetattribute{\the\glslabeltok}{regular}{false}%
6101   }%
6102   {}%
6103 }%
6104 }%
6105 {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
6106 \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
6107 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
6108 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
6109 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
6110 \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form followed by the long form as a footnote.

```
6111 \renewcommand*{\glsxtrfullformat}[2]{%
6112   \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
6113   \ifglsxtrininsertinside\else##2\fi
6114   \protect\glsxtrabbrvfootnote{##1}%
6115   {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
6116 }%
6117 \renewcommand*{\glsxtrfullplformat}[2]{%
6118   \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
6119   \ifglsxtrininsertinside\else##2\fi
6120   \protect\glsxtrabbrvfootnote{##1}%
6121   {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6122 }%
6123 \renewcommand*{\Glsxtrfullformat}[2]{%
6124   \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
6125   \ifglsxtrininsertinside\else##2\fi
6126   \protect\glsxtrabbrvfootnote{##1}%
6127   {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
6128 }%
6129 \renewcommand*{\Glsxtrfullplformat}[2]{%
6130   \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
6131   \ifglsxtrininsertinside\else##2\fi
6132   \protect\glsxtrabbrvfootnote{##1}%
6133   {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6134 }%
```

The first use full form and the inline full form use the short (long) style.

```
6135 \renewcommand*{\glsxtrinlinefullformat}[2]{%
6136   \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
6137 }
```

```

6137     \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6138     \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
6139 }%
6140 \renewcommand*{\glxtrinlinefullplformat}[2]{%
6141     \glsfirstabbrvfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6142     \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6143     \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
6144 }%
6145 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6146     \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6147     \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6148     \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
6149 }%
6150 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6151     \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6152     \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6153     \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
6154 }%
6155 }

```

#### short-footnote

```
6156 \letabbreviationstyle{short-footnote}{footnote}
```

postfootnote Similar to the above but the footnote is placed afterwards, outside the link. This avoids nested links and can also move the footnote marker after any following punctuation mark. Pre v1.07 included `\footnote` in the first keys, which was incorrect as it caused duplicate footnotes.

```

6157 \newabbreviationstyle{postfootnote}%
6158 {%
6159     \renewcommand*{\CustomAbbreviationFields}{%
6160         name={\protect\glsabbrvfont{\the\glssshorttok}},
6161         sort={\the\glssshorttok},
6162         description={\the\glslongtok},%
6163         first={\protect\glsfirstabbrvfont{\the\glssshorttok}},%
6164         firstplural={\protect\glsfirstabbrvfont{\the\glssshortpltok}},%
6165         plural={\protect\glsabbrvfont{\the\glssshortpltok}}}%

```

Make this category insert a footnote after the link if this was the first use, and unset the regular attribute if it has been set.

```

6166 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6167     \csdef{glxtrpostlink\glscategorylabel}{%
6168         \glxtrifwasfirstuse
6169         {%

```

Needs the specific font command here as the style may have been lost by the time the footnote occurs.

```

6170         \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}%
6171             {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}%
6172         }%

```

```

6173     {}%
6174   }%
6175   \glshasattribute{\the\glslabeltok}{regular}%
6176   {%
6177     \glsssetAttribute{\the\glslabeltok}{regular}{false}%
6178   }%
6179   {}%
6180 }%

```

The footnote needs to be suppressed in the inline form, so `\glxtrfull` must set the first use switch off.

```

6181 \renewcommand*{\glxtrsetupfulldefs}{%
6182   \let\glxtrifwasfirstuse\@secondoftwo
6183 }%
6184 }%
6185 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

6186 \renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
6187 \renewcommand*{\glssabbrvfont}[1]{\glssabbrvdefaultfont{##1}}%
6188 \renewcommand*{\glssfirstabbrvfont}[1]{\glssfirstabbrvdefaultfont{##1}}%
6189 \renewcommand*{\glssfirstlongfont}[1]{\glssfirstlongfootnotefont{##1}}%
6190 \renewcommand*{\glsslongfont}[1]{\glsslongfootnotefont{##1}}%

```

The full format displays the short form. The long form is deferred.

```

6191 \renewcommand*{\glxtrfullformat}[2]{%
6192   \glssfirstabbrvfont{\glssaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6193   \ifglxtrininsertinside\else##2\fi
6194 }%
6195 \renewcommand*{\glxtrfullplformat}[2]{%
6196   \glssfirstabbrvfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6197   \ifglxtrininsertinside\else##2\fi
6198 }%
6199 \renewcommand*{\Glsxtrfullformat}[2]{%
6200   \glssfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6201   \ifglxtrininsertinside\else##2\fi
6202 }%
6203 \renewcommand*{\Glsxtrfullplformat}[2]{%
6204   \glssfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6205   \ifglxtrininsertinside\else##2\fi
6206 }%

```

The first use full form and the inline full form use the short (long) style.

```

6207 \renewcommand*{\glxtrinlinefullformat}[2]{%
6208   \glssfirstabbrvfont{\glssaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6209   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6210   \glxtrparen{\glssfirstlongfootnotefont{\glssaccesslong{##1}}}%
6211 }%
6212 \renewcommand*{\glxtrinlinefullplformat}[2]{%
6213   \glssfirstabbrvfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6214   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%

```

```

6215 \glstrparen{\glfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6216 }%
6217 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6218 \glfirstabbrvfont{\Glsaccessshort{##1}\ifglstrinsertinside##2\fi}%
6219 \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
6220 \glstrparen{\glfirstlongfootnotefont{\glsaccesslong{##1}}}%
6221 }%
6222 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6223 \glfirstabbrvfont{\Glsaccessshortpl{##1}\ifglstrinsertinside##2\fi}%
6224 \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
6225 \glstrparen{\glfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6226 }%
6227 }

```

rt-postfootnote

```
6228 \letabbreviationstyle{short-postfootnote}{postfootnote}
```

**short** Provide a style that only displays the short form on first use, but the short and long form can be displayed with the “full” commands that use the inline format. If the user supplies a description, the long form won’t be displayed in the predefined glossary styles, but the post description hook can be employed to automatically insert it.

```

6229 \newabbreviationstyle{short}%
6230 {%
6231 \renewcommand*{\CustomAbbreviationFields}{%
6232 name={\protect\glssabbrvfont{\the\glssshorttok}},
6233 sort={\the\glssshorttok},
6234 first={\protect\glssfirstabbrvfont{\the\glssshorttok}},
6235 firstplural={\protect\glssfirstabbrvfont{\the\glssshortpltok}},
6236 text={\protect\glssabbrvfont{\the\glssshorttok}},
6237 plural={\protect\glssabbrvfont{\the\glssshortpltok}},
6238 description={\the\glslongtok}}%
6239 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6240 \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
6241 }%
6242 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

6243 \renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
6244 \renewcommand*{\glssabbrvfont}[1]{\glssabbrvdefaultfont{##1}}%
6245 \renewcommand*{\glssfirstabbrvfont}[1]{\glssfirstabbrvdefaultfont{##1}}%
6246 \renewcommand*{\glssfirstlongfont}[1]{\glssfirstlongdefaultfont{##1}}%
6247 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The inline full form displays the short form followed by the long form in parentheses.

```

6248 \renewcommand*{\glstrinlinefullformat}[2]{%
6249 \protect\glssfirstabbrvfont{\glsaccessshort{##1}}%
6250 \ifglstrinsertinside##2\fi}%
6251 \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
6252 \glstrparen{\glssfirstlongfont{\glsaccesslong{##1}}}%
6253 }%

```

```

6254 \renewcommand*{\glxtrinlinefullplformat}[2]{%
6255   \protect\glsfirstabbrvfont{\glsaccessshortpl{##1}%
6256     \ifglxtrininsertinside##2\fi}%
6257   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6258   \glxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
6259 }%
6260 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6261   \protect\glsfirstabbrvfont{\glsaccessshort{##1}%
6262     \ifglxtrininsertinside##2\fi}%
6263   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6264   \glxtrparen{\glsfirstlongfont{\Glsaccesslong{##1}}}%
6265 }%
6266 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6267   \protect\glsfirstabbrvfont{\glsaccessshortpl{##1}%
6268     \ifglxtrininsertinside##2\fi}%
6269   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6270   \glxtrparen{\glsfirstlongfont{\Glsaccesslongpl{##1}}}%
6271 }%

```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

6272 \renewcommand*{\glxtrfullformat}[2]{%
6273   \glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6274   \ifglxtrininsertinside\else##2\fi
6275 }%
6276 \renewcommand*{\glxtrfullplformat}[2]{%
6277   \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6278   \ifglxtrininsertinside\else##2\fi
6279 }%
6280 \renewcommand*{\Glsxtrfullformat}[2]{%
6281   \glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6282   \ifglxtrininsertinside\else##2\fi
6283 }%
6284 \renewcommand*{\Glsxtrfullplformat}[2]{%
6285   \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6286   \ifglxtrininsertinside\else##2\fi
6287 }%
6288 }

```

Set this as the default style for acronyms:

```

6289 \setabbreviationstyle[acronym]{short}

```

short-nolong

```

6290 \letabbreviationstyle{short-nolong}{short}

```

rt-nolong-noreg Like short-nolong but doesn't set the regular attribute.

```

6291 \newabbreviationstyle{short-nolong-noreg}%
6292 {%
6293   \GlsXtrUseAbbrStyleSetup{short-nolong}%

```



Unset the regular attribute if it has been set.

```

6294 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6295   \glshasattribute{\the\glslabeltok}{regular}%
6296   {%
6297     \glsselattribute{\the\glslabeltok}{regular}{false}%
6298   }%
6299   {}%
6300 }%
6301 }%
6302 {%
6303   \GlsXtrUseAbbrStyleFmts{short-nolong}%
6304 }

```

trshortdescname

```

6305 \newcommand*{\glxtrshortdescname}{%
6306   \protect\glsabbrvfont{\the\glsshorttok}%
6307 }

```

short-desc The user must supply the description in this style. The long form is added to the name. The short style (possibly with the post-description hooks set) might be a better option.

```

6308 \newabbreviationstyle{short-desc}%
6309 {%
6310   \renewcommand*{\CustomAbbreviationFields}{%
6311     name={\glxtrshortdescname},
6312     sort={\the\glsshorttok},
6313     first={\protect\glsfirstabbrvfont{\the\glsshorttok}},
6314     firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}},
6315     text={\protect\glsabbrvfont{\the\glsshorttok}},
6316     plural={\protect\glsabbrvfont{\the\glsshortpltok}},
6317     description={\the\glslongtok}}%
6318   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6319     \glsselattribute{\the\glslabeltok}{regular}{true}}%
6320 }%
6321 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

6322 \renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
6323 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
6324 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
6325 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6326 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The inline full form displays the short format followed by the long form in parentheses.

```

6327 \renewcommand*{\glxtrinlinefullformat}[2]{%
6328   \glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6329   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6330   \glxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
6331 }%
6332 \renewcommand*{\glxtrinlinefullplformat}[2]{%
6333   \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%

```

```

6334 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6335 \glxtrparen{\glsfirstlongfont{\glssaccesslongpl{##1}}}%
6336 }%
6337 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6338 \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6339 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6340 \glxtrparen{\glsfirstlongfont{\glssaccesslong{##1}}}%
6341 }%
6342 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6343 \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6344 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6345 \glxtrparen{\glsfirstlongfont{\glssaccesslongpl{##1}}}%
6346 }%

```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

6347 \renewcommand*{\glxtrfullformat}[2]{%
6348 \glsfirstabbrvfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6349 \ifglxtrinsertinside\else##2\fi
6350 }%
6351 \renewcommand*{\glxtrfullplformat}[2]{%
6352 \glsfirstabbrvfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6353 \ifglxtrinsertinside\else##2\fi
6354 }%
6355 \renewcommand*{\Glsxtrfullformat}[2]{%
6356 \glsfirstabbrvfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6357 \ifglxtrinsertinside\else##2\fi
6358 }%
6359 \renewcommand*{\Glsxtrfullplformat}[2]{%
6360 \glsfirstabbrvfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6361 \ifglxtrinsertinside\else##2\fi
6362 }%
6363 }

```

ort-nolong-desc

```
6364 \letabbreviationstyle{short-nolong-desc}{short-desc}
```

long-desc-noreg Like short-nolong-desc but doesn't set the regular attribute.

```

6365 \newabbreviationstyle{short-nolong-desc-noreg}%
6366 {%
6367 \GlsXtrUseAbbrStyleSetup{short-nolong-desc}%

```

Unset the regular attribute if it has been set.

```

6368 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6369 \glshasattribute{\the\glslabeltok}{regular}%
6370 {%
6371 \glissetattribute{\the\glslabeltok}{regular}{false}%
6372 }%
6373 {}%
6374 }%

```

```

6375 }%
6376 {%
6377   \GlsXtrUseAbbrStyleFmts{short-nolong-desc}%
6378 }

```

long-desc Provide a style that only displays the long form, but the long and short form can be displayed with the “full” commands that use the inline format. The predefined glossary styles won’t show the short form. The user must supply a description for this style.

```

6379 \newabbreviationstyle{long-desc}%
6380 {%
6381   \renewcommand*{\CustomAbbreviationFields}{%
6382     name={\protect\protect\glslongfont{\the\glslongtok}},
6383     sort={\the\glslongtok},
6384     first={\protect\glsfirstlongfont{\the\glslongtok}},
6385     firstplural={\protect\glsfirstlongfont{\the\glslongpltok}},
6386     text={\glslongfont{\the\glslongtok}},
6387     plural={\glslongfont{\the\glslongpltok}}}%
6388   }%
6389   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6390     \glssetattribute{\the\glslabeltok}{regular}{true}}%
6391 }%
6392 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

6393 \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
6394 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
6395 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
6396 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6397 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

6398 \renewcommand*{\glsxtrsubsequentfmt}[2]{%
6399   \glslongfont{\glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
6400   \ifglsxtrininsertinside \else##2\fi
6401 }%
6402 \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
6403   \glslongfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
6404   \ifglsxtrininsertinside \else##2\fi
6405 }%
6406 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
6407   \glslongfont{\Glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
6408   \ifglsxtrininsertinside \else##2\fi
6409 }%
6410 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
6411   \glslongfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
6412   \ifglsxtrininsertinside \else##2\fi
6413 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

6414 \renewcommand*{\glsxtrinlinefullformat}[2]{%

```

```

6415 \glsfirstlongfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6416 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6417 \glxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
6418 }%
6419 \renewcommand*{\glxtrinlinefullplformat}[2]{%
6420 \glsfirstlongfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6421 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6422 \glxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
6423 }%
6424 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6425 \glsfirstlongfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6426 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6427 \glxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
6428 }%
6429 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6430 \glsfirstlongfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6431 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6432 \glxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
6433 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

6434 \renewcommand*{\glxtrfullformat}[2]{%
6435 \glsfirstlongfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6436 \ifglxtrinsertinside\else##2\fi
6437 }%
6438 \renewcommand*{\glxtrfullplformat}[2]{%
6439 \glsfirstlongfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6440 \ifglxtrinsertinside\else##2\fi
6441 }%
6442 \renewcommand*{\Glsxtrfullformat}[2]{%
6443 \glsfirstlongfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6444 \ifglxtrinsertinside\else##2\fi
6445 }%
6446 \renewcommand*{\Glsxtrfullplformat}[2]{%
6447 \glsfirstlongfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6448 \ifglxtrinsertinside\else##2\fi
6449 }%
6450 }

```

`ng-noshort-desc` Provide a synonym that matches similar styles.

```
6451 \letabbreviationstyle{long-noshort-desc}{long-desc}
```

`hort-desc-noreg` Like `long-noshort-desc` but doesn't set the regular attribute.

```

6452 \newabbreviationstyle{long-noshort-desc-noreg}%
6453 {%
6454 \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%

```

Unset the regular attribute if it has been set.

```
6455 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
```

```

6456 \glshasattribute{\the\glslabelltok}{regular}%
6457 {%
6458 \glissetattribute{\the\glslabelltok}{regular}{false}%
6459 }%
6460 {}%
6461 }%
6462 }%
6463 {%
6464 \GlsXtrUseAbbrStyleFmts{long-noshort-desc}%
6465 }

```

**long** It doesn't really make a great deal of sense to have a long-only style that doesn't have a description (unless no glossary is required), but the best course of action here is to use the short form as the name and the long form as the description.

```

6466 \newabbreviationstyle{long}%
6467 {%
6468 \renewcommand*{\CustomAbbreviationFields}{%
6469 name={\protect\glsabbrvfont{\the\glsshorttok}},
6470 sort={\the\glsshorttok},
6471 first={\protect\glsfirstlongfont{\the\glslongtok}},
6472 firstplural={\protect\glsfirstlongfont{\the\glslongpltok}},
6473 text={\glslongfont{\the\glslongtok}},
6474 plural={\glslongfont{\the\glslongpltok}},%
6475 description={\the\glslongtok}%
6476 }%
6477 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6478 \glissetattribute{\the\glslabelltok}{regular}{true}}%
6479 }%
6480 {%
6481 \GlsXtrUseAbbrStyleFmts{long-desc}%
6482 }

```

**long-noshort** Provide a synonym that matches similar styles.

```

6483 \letabbreviationstyle{long-noshort}{long}

```

**g-noshort-noreg** Like long-noshort but doesn't set the regular attribute.

```

6484 \newabbreviationstyle{long-noshort-noreg}%
6485 {%
6486 \GlsXtrUseAbbrStyleSetup{long-noshort}%

```

Unset the regular attribute if it has been set.

```

6487 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6488 \glshasattribute{\the\glslabelltok}{regular}%
6489 {%
6490 \glissetattribute{\the\glslabelltok}{regular}{false}%
6491 }%
6492 {}%
6493 }%
6494 }%

```

```

6495 {%
6496   \GlsXtrUseAbbrStyleFmts{long-noshort}%
6497 }

```

### 1.6.3 Predefined Styles (Small Capitals)

These styles use `\textsc` for the short form.

```

\glstrscfont   Maintained for backward-compatibility.
6498 \newcommand*{\glstrscfont}[1]{\textsc{#1}}

```

```

\glsabbrvscfont   Added for consistent naming.
6499 \newcommand*{\glsabbrvscfont}{\glstrscfont}

```

```

\glstrfirstscfont   Maintained for backward-compatibility.
6500 \newcommand*{\glstrfirstscfont}[1]{\glsabbrvscfont{#1}}

```

```

\glstrfirstabbrvscfont   Added for consistent naming.
6501 \newcommand*{\glstrfirstabbrvscfont}{\glstrfirstscfont}

```

and for the default short form suffix:

```

\glstrscsuffix
6502 \newcommand*{\glstrscsuffix}{\glstextup{\glstrabbrvpluralsuffix}}

```

```

long-short-sc
6503 \newabbreviationstyle{long-short-sc}%
6504 {%
6505   \renewcommand*{\CustomAbbreviationFields}{%
6506     name={\protect\glsabbrvscfont{\the\glsshorttok}},
6507     sort={\the\glsshorttok},
6508     first={\protect\glstrfirstlongdefaultfont{\the\glslongtok}%
6509       \protect\glstrfullsep{\the\glslabeltok}%
6510       \glstrparen{\protect\glstrfirstabbrvscfont{\the\glsshorttok}}},%
6511     firstplural={\protect\glstrfirstlongdefaultfont{\the\glslongpltok}%
6512       \protect\glstrfullsep{\the\glslabeltok}%
6513       \glstrparen{\protect\glstrfirstabbrvscfont{\the\glsshortpltok}}},%
6514     plural={\protect\glsabbrvscfont{\the\glsshortpltok}},%
6515     description={\the\glslongtok}}%
6516   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6517     \glshasattribute{\the\glslabeltok}{regular}%
6518     {%
6519       \glsssetattribute{\the\glslabeltok}{regular}{false}%
6520     }%
6521   }%
6522 }%
6523 }%
6524 {%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```
6525 \renewcommand*{\abbrvpluralsuffix}{\protect\glxtrscsuffix}%
6526 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
6527 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
```

Use the default long fonts.

```
6528 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6529 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
6530 \renewcommand*{\glxtrfullformat}[2]{%
6531   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6532   \ifglxtrinsertinside\else##2\fi
6533   \glxtrfullsep{##1}%
6534   \glxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
6535 }%
6536 \renewcommand*{\glxtrfullplformat}[2]{%
6537   \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6538   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6539   \glxtrparen{\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
6540 }%
6541 \renewcommand*{\Glsxtrfullformat}[2]{%
6542   \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6543   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6544   \glxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
6545 }%
6546 \renewcommand*{\Glsxtrfullplformat}[2]{%
6547   \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6548   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6549   \glxtrparen{\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
6550 }%
6551 }
```

g-short-sc-desc

```
6552 \newabbreviationstyle{long-short-sc-desc}%
6553 {%
6554   \renewcommand*{\CustomAbbreviationFields}{%
6555     name={\glxtrlongshortdescname},
6556     sort={\glxtrlongshortdescsort},%
6557     first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
6558       \protect\glxtrfullsep{\the\glslabeltok}%
6559       \glxtrparen{\protect\glsfirstabbrvscfont{\the\glsshorttok}}},%
6560     firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
6561       \protect\glxtrfullsep{\the\glslabeltok}%
6562       \glxtrparen{\protect\glsfirstabbrvscfont{\the\glsshortpltok}}},%
6563     text={\protect\glsabbrvscfont{\the\glsshorttok}},%
6564     plural={\protect\glsabbrvscfont{\the\glsshortpltok}}%
6565   }%
```

Unset the regular attribute if it has been set.

```

6566 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6567   \glshasattribute{\the\glslabeltok}{regular}%
6568   {%
6569     \glissetattribute{\the\glslabeltok}{regular}{false}%
6570   }%
6571   {}%
6572 }%
6573 }%
6574 {%

```

As long-short-sc style:

```

6575 \GlsXtrUseAbbrStyleFmts{long-short-sc}%
6576 }

```

Now the short (long) version

```

6577 \newabbreviationstyle{short-sc-long}%
6578 {%
6579   \renewcommand*{\CustomAbbreviationFields}{%
6580     name={\protect\glsabbrvscfont{\the\glsshorttok}},
6581     sort={\the\glsshorttok},
6582     description={\the\glslongtok},%
6583     first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
6584       \protect\glsxtrfullsep{\the\glslabeltok}%
6585       \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
6586     firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
6587       \protect\glsxtrfullsep{\the\glslabeltok}%
6588       \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
6589     plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}

```

Unset the regular attribute if it has been set.

```

6590 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6591   \glshasattribute{\the\glslabeltok}{regular}%
6592   {%
6593     \glissetattribute{\the\glslabeltok}{regular}{false}%
6594   }%
6595   {}%
6596 }%
6597 }%
6598 {%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

6599 \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtrscsuffix}%
6600 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
6601 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
6602 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6603 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

6604 \renewcommand*{\glsxtrfullformat}[2]{%
6605   \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
6606   \ifglsxtrininsertinside\else##2\fi

```



```

6607 \glxtrfullsep{##1}%
6608 \glxtrparen{\glsfirstlongdefaultfont{\glssaccesslong{##1}}}%
6609 }%
6610 \renewcommand*{\glxtrfullplformat}[2]{%
6611 \glsfirstabbrvscfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6612 \ifglxtrininsertinside\else##2\fi
6613 \glxtrfullsep{##1}%
6614 \glxtrparen{\glsfirstlongdefaultfont{\glssaccesslongpl{##1}}}%
6615 }%
6616 \renewcommand*{\Glsxtrfullformat}[2]{%
6617 \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6618 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6619 \glxtrparen{\glsfirstlongdefaultfont{\glssaccesslong{##1}}}%
6620 }%
6621 \renewcommand*{\Glsxtrfullplformat}[2]{%
6622 \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6623 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6624 \glxtrparen{\glsfirstlongdefaultfont{\glssaccesslongpl{##1}}}%
6625 }%
6626 }

```

As before but user provides description

```

6627 \newabbreviationstyle{short-sc-long-desc}%
6628 {%
6629 \renewcommand*{\CustomAbbreviationFields}{%
6630 name={\glxtrshortlongdescname},
6631 sort={\glxtrshortlongdescsort},
6632 first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
6633 \protect\glxtrfullsep{\the\glslabeltok}%
6634 \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
6635 firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
6636 \protect\glxtrfullsep{\the\glslabeltok}%
6637 \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
6638 text={\protect\glssabbrvscfont{\the\glsshorttok}},%
6639 plural={\protect\glssabbrvscfont{\the\glsshortpltok}}}%
6640 }%

```

Unset the regular attribute if it has been set.

```

6641 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6642 \glshasattribute{\the\glslabeltok}{regular}%
6643 {%
6644 \glissetattribute{\the\glslabeltok}{regular}{false}%
6645 }%
6646 {}%
6647 }%
6648 }%
6649 {%

```

As short-sc-long style:

```

6650 \GlsXtrUseAbbrStyleFmts{short-sc-long}%

```

6651 }

short-sc

```
6652 \newabbreviationstyle{short-sc}%
6653 {%
6654   \renewcommand*{\CustomAbbreviationFields}{%
6655     name={\protect\glsabbrvscfont{\the\glsshorttok}},
6656     sort={\the\glsshorttok},
6657     first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},
6658     firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},
6659     text={\protect\glsabbrvscfont{\the\glsshorttok}},
6660     plural={\protect\glsabbrvscfont{\the\glsshortpltok}},
6661     description={\the\glslongtok}}%
6662   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6663     \glssetattribute{\the\glslabeltok}{regular}{true}}%
6664 }%
6665 {%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
6666   \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtrscsuffix}%
6667   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
6668   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
6669   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6670   \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short form followed by the long form in parentheses.

```
6671   \renewcommand*{\glsxtrinlinefullformat}[2]{%
6672     \protect\glsfirstabbrvscfont{\glsaccesssshort{##1}}%
6673     \ifglsxtrininsertinside##2\fi}%
6674     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
6675     \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
6676   }%
6677   \renewcommand*{\glsxtrinlinefullplformat}[2]{%
6678     \protect\glsfirstabbrvscfont{\glsaccesssshortpl{##1}}%
6679     \ifglsxtrininsertinside##2\fi}%
6680     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
6681     \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
6682   }%
6683   \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6684     \protect\glsfirstabbrvscfont{\glsaccesssshort{##1}}%
6685     \ifglsxtrininsertinside##2\fi}%
6686     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
6687     \glsxtrparen{\glsfirstlongdefaultfont{\Glsaccesslong{##1}}}%
6688   }%
6689   \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6690     \protect\glsfirstabbrvscfont{\glsaccesssshortpl{##1}}%
6691     \ifglsxtrininsertinside##2\fi}%
6692     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
6693     \glsxtrparen{\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}}}%
6694   }%
```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

6695 \renewcommand*{\glxtrfullformat}[2]{%
6696   \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6697   \ifglxtrinsertinside\else##2\fi
6698 }%
6699 \renewcommand*{\glxtrfullplformat}[2]{%
6700   \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6701   \ifglxtrinsertinside\else##2\fi
6702 }%
6703 \renewcommand*{\Glsxtrfullformat}[2]{%
6704   \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6705   \ifglxtrinsertinside\else##2\fi
6706 }%
6707 \renewcommand*{\Glsxtrfullplformat}[2]{%
6708   \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6709   \ifglxtrinsertinside\else##2\fi
6710 }%
6711 }

```

short-sc-nolong

```

6712 \letabbreviationstyle{short-sc-nolong}{short-sc}

```

short-sc-desc

```

6713 \newabbreviationstyle{short-sc-desc}{%
6714 {%
6715   \renewcommand*{\CustomAbbreviationFields}{%
6716     name={\glxtrshortdescname},
6717     sort={\the\glsshorttok},
6718     first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},
6719     firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},
6720     text={\protect\glsabbrvscfont{\the\glsshorttok}},
6721     plural={\protect\glsabbrvscfont{\the\glsshortpltok}},
6722     description={\the\glslongtok}}%
6723   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6724     \glssetattribute{\the\glslabeltok}{regular}{true}}%
6725 }%
6726 {%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

6727 \renewcommand*{\abbrvpluralsuffix}{\protect\glxtrscsuffix}%
6728 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
6729 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
6730 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6731 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The inline full form displays the short format followed by the long form in parentheses.

```

6732 \renewcommand*{\glxtrinlinfullformat}[2]{%
6733   \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6734   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%

```

```

6735 \glxtrparen{\glfirstlongdefaultfont{\glaccesslong{##1}}}%
6736 }%
6737 \renewcommand*{\glxtrinlinefullplformat}[2]{%
6738 \glfirstabbrvscfont{\glaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6739 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6740 \glxtrparen{\glfirstlongdefaultfont{\glaccesslongpl{##1}}}%
6741 }%
6742 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6743 \glfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6744 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6745 \glxtrparen{\glfirstlongdefaultfont{\glaccesslong{##1}}}%
6746 }%
6747 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6748 \glfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6749 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6750 \glxtrparen{\glfirstlongdefaultfont{\glaccesslongpl{##1}}}%
6751 }%

```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

6752 \renewcommand*{\glxtrfullformat}[2]{%
6753 \glfirstabbrvscfont{\glaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6754 \ifglxtrininsertinside\else##2\fi
6755 }%
6756 \renewcommand*{\glxtrfullplformat}[2]{%
6757 \glfirstabbrvscfont{\glaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6758 \ifglxtrininsertinside\else##2\fi
6759 }%
6760 \renewcommand*{\Glsxtrfullformat}[2]{%
6761 \glfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
6762 \ifglxtrininsertinside\else##2\fi
6763 }%
6764 \renewcommand*{\Glsxtrfullplformat}[2]{%
6765 \glfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
6766 \ifglxtrininsertinside\else##2\fi
6767 }%
6768 }

```

-sc-nolong-desc

```
6769 \letabbreviationstyle{short-sc-nolong-desc}{short-sc-desc}
```

long-noshort-sc The smallcaps font will only be used if the short form is explicitly invoked through commands like `\glxtrshort`.

```

6770 \newabbreviationstyle{long-noshort-sc}%
6771 {%
6772 \renewcommand*{\CustomAbbreviationFields}{%
6773 name={\protect\glabbrvscfont{\the\glsshorttok}},
6774 sort={\the\glsshorttok},
6775 first={\protect\glfirstlongdefaultfont{\the\glslongtok}},

```

```

6776   firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},
6777   text={\protect\glslongdefaultfont{\the\glslongtok}},
6778   plural={\protect\glslongdefaultfont{\the\glslongpltok}},%
6779   description={\the\glslongtok}%
6780 }%
6781 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6782   \glssetattribute{\the\glslabeltok}{regular}{true}}%
6783 }%
6784 {%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

6785 \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtrscsuffix}%
6786 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
6787 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
6788 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6789 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

6790 \renewcommand*{\glxtrsubsequentfmt}[2]{%
6791   \glslongdefaultfont{\glsaccesslong{##1}\ifglxtrininsertinside ##2\fi}%
6792   \ifglxtrininsertinside \else##2\fi
6793 }%
6794 \renewcommand*{\glxtrsubsequentplfmt}[2]{%
6795   \glslongdefaultfont{\glsaccesslongpl{##1}\ifglxtrininsertinside ##2\fi}%
6796   \ifglxtrininsertinside \else##2\fi
6797 }%
6798 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
6799   \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrininsertinside ##2\fi}%
6800   \ifglxtrininsertinside \else##2\fi
6801 }%
6802 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
6803   \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrininsertinside ##2\fi}%
6804   \ifglxtrininsertinside \else##2\fi
6805 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

6806 \renewcommand*{\glxtrinlinefullformat}[2]{%
6807   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrininsertinside##2\fi}%
6808   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6809   \glxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
6810 }%
6811 \renewcommand*{\glxtrinlinefullplformat}[2]{%
6812   \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrininsertinside##2\fi}%
6813   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6814   \glxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
6815 }%
6816 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6817   \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrininsertinside##2\fi}%
6818   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
6819   \glxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
6820 }%

```

```

6821 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6822   \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6823   \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6824   \glxtrparen{\protect\glsfirstabbrvscfont{\Glsaccessshortpl{##1}}}%
6825 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

6826 \renewcommand*{\glxtrfullformat}[2]{%
6827   \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6828   \ifglxtrinsertinside\else##2\fi
6829 }%
6830 \renewcommand*{\glxtrfullplformat}[2]{%
6831   \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6832   \ifglxtrinsertinside\else##2\fi
6833 }%
6834 \renewcommand*{\Glsxtrfullformat}[2]{%
6835   \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6836   \ifglxtrinsertinside\else##2\fi
6837 }%
6838 \renewcommand*{\Glsxtrfullplformat}[2]{%
6839   \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6840   \ifglxtrinsertinside\else##2\fi
6841 }%
6842 }

```

long-sc Backward compatibility:

```

6843 \@glxtr@deprecated@abbrstyle{long-sc}{long-noshort-sc}

```

noshort-sc-desc The smallcaps font will only be used if the short form is explicitly invoked through commands like `\glsshort`.

```

6844 \newabbreviationstyle{long-noshort-sc-desc}%
6845 {%
6846   \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
6847 }%
6848 {%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

6849 \renewcommand*{\abbrvpluralsuffix}{\protect\glxtrscsuffix}%
6850 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
6851 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
6852 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
6853 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

6854 \renewcommand*{\glxtrsubsequentfmt}[2]{%
6855   \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
6856   \ifglxtrinsertinside \else##2\fi
6857 }%
6858 \renewcommand*{\glxtrsubsequentplfmt}[2]{%

```

```

6859 \glslongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
6860 \ifglxtrinsertinside \else##2\fi
6861 }%
6862 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
6863 \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
6864 \ifglxtrinsertinside \else##2\fi
6865 }%
6866 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
6867 \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
6868 \ifglxtrinsertinside \else##2\fi
6869 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

6870 \renewcommand*{\glsxtrinlinefullformat}[2]{%
6871 \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6872 \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
6873 \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
6874 }%
6875 \renewcommand*{\glsxtrinlinefullplformat}[2]{%
6876 \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6877 \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
6878 \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
6879 }%
6880 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6881 \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6882 \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
6883 \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
6884 }%
6885 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6886 \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6887 \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
6888 \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
6889 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

6890 \renewcommand*{\glsxtrfullformat}[2]{%
6891 \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6892 \ifglxtrinsertinside\else##2\fi
6893 }%
6894 \renewcommand*{\glsxtrfullplformat}[2]{%
6895 \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
6896 \ifglxtrinsertinside\else##2\fi
6897 }%
6898 \renewcommand*{\Glsxtrfullformat}[2]{%
6899 \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
6900 \ifglxtrinsertinside\else##2\fi
6901 }%
6902 \renewcommand*{\Glsxtrfullplformat}[2]{%
6903 \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%

```

```

6904 \ifglxtrinsertinside\else##2\fi
6905 }%
6906 }

```

long-desc-sc Backward compatibility:

```

6907 \@glxtr@deprecated@abbrstyle{long-desc-sc}{long-noshort-sc-desc}

```

ort-sc-footnote

```

6908 \newabbreviationstyle{short-sc-footnote}%
6909 {%
6910 \renewcommand*{\CustomAbbreviationFields}{%
6911 name={\protect\glsabbrvscfont{\the\glsshorttok}},
6912 sort={\the\glsshorttok},
6913 description={\the\glslongtok},%
6914 first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
6915 \protect\glxtrabbrvfootnote{\the\glslabeltok}%
6916 {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
6917 firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
6918 \protect\glxtrabbrvfootnote{\the\glslabeltok}%
6919 {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
6920 plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%

```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the regular attribute if it has been set.

```

6921 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6922 \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
6923 \glsattribute{\the\glslabeltok}{regular}%
6924 {%
6925 \glssetattribute{\the\glslabeltok}{regular}{false}%
6926 }%
6927 }%
6928 }%
6929 }%
6930 {%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

6931 \renewcommand*{\abbrvpluralsuffix}{\protect\glxtrscsuffix}%
6932 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
6933 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
6934 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
6935 \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%

```

The full format displays the short form followed by the long form as a footnote.

```

6936 \renewcommand*{\glxtrfullformat}[2]{%
6937 \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6938 \ifglxtrinsertinside\else##2\fi
6939 \protect\glxtrabbrvfootnote{##1}%
6940 {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
6941 }%
6942 \renewcommand*{\glxtrfullplformat}[2]{%

```



```

6943 \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6944 \ifglxtrinsertinside\else##2\fi
6945 \protect\glxtrabbrvfootnote{##1}%
6946 {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6947 }%
6948 \renewcommand*{\Glsxtrfullformat}[2]{%
6949 \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6950 \ifglxtrinsertinside\else##2\fi
6951 \protect\glxtrabbrvfootnote{##1}%
6952 {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
6953 }%
6954 \renewcommand*{\Glsxtrfullplformat}[2]{%
6955 \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6956 \ifglxtrinsertinside\else##2\fi
6957 \protect\glxtrabbrvfootnote{##1}%
6958 {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6959 }%

```

The first use full form and the inline full form use the short (long) style.

```

6960 \renewcommand*{\glsxtrinlinefullformat}[2]{%
6961 \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6962 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6963 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
6964 }%
6965 \renewcommand*{\glsxtrinlinefullplformat}[2]{%
6966 \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6967 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6968 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6969 }%
6970 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
6971 \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
6972 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6973 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
6974 }%
6975 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
6976 \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
6977 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
6978 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
6979 }%
6980 }

```

footnote-sc Backward compatibility:

```

6981 \@glxtr@deprecated@abbrstyle{footnote-sc}{short-sc-footnote}

```

sc-postfootnote

```

6982 \newabbreviationstyle{short-sc-postfootnote}%
6983 {%
6984 \renewcommand*{\CustomAbbreviationFields}{%
6985 name={\protect\glsabbrvscfont{\the\glsshorttok}},
6986 sort={\the\glsshorttok},

```

```

6987   description={\the\glslongtok},%
6988   first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},%
6989   firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},%
6990   plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%

```

Make this category insert a footnote after the link if this was the first use, and unset the regular attribute if it has been set.

```

6991   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
6992     \csdef{glsxtrpostlink\glscategorylabel}{%
6993       \glsxtrifwasfirstuse
6994       {%

```

Needs the specific font command here as the style may have been lost by the time the footnote occurs.

```

6995         \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}}%
6996         {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}}%
6997     }%
6998   }%
6999 }%
7000   \glsattribute{\the\glslabeltok}{regular}%
7001   {%
7002     \glssetattribute{\the\glslabeltok}{regular}{false}%
7003   }%
7004   {}%
7005 }%

```

The footnote needs to be suppressed in the inline form, so `\glsxtrfull` must set the first use switch off.

```

7006   \renewcommand*{\glsxtrsetupfulldefs}{%
7007     \let\glsxtrifwasfirstuse\@secondoftwo
7008   }%
7009 }%
7010 {%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

7011   \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtrscsuffix}%
7012   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
7013   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
7014   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
7015   \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%

```

The full format displays the short form. The long form is deferred.

```

7016   \renewcommand*{\glsxtrfullformat}[2]{%
7017     \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsetinside##2\fi}%
7018     \ifglsxtrinsetinside\else##2\fi
7019   }%
7020   \renewcommand*{\glsxtrfullplformat}[2]{%
7021     \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsetinside##2\fi}%
7022     \ifglsxtrinsetinside\else##2\fi
7023   }%
7024   \renewcommand*{\Glsxtrfullformat}[2]{%

```

```

7025 \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7026 \ifglxtrinsertinside\else##2\fi
7027 }%
7028 \renewcommand*{\Glsxtrfullplformat}[2]{%
7029 \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7030 \ifglxtrinsertinside\else##2\fi
7031 }%

```

The first use full form and the inline full form use the short (long) style.

```

7032 \renewcommand*{\glxtrinlinefullformat}[2]{%
7033 \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7034 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7035 \glxtrparen{\glsfirstlongfootnotefont{\Glsaccesslong{##1}}}%
7036 }%
7037 \renewcommand*{\glxtrinlinefullplformat}[2]{%
7038 \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7039 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7040 \glxtrparen{\glsfirstlongfootnotefont{\Glsaccesslongpl{##1}}}%
7041 }%
7042 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
7043 \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7044 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7045 \glxtrparen{\glsfirstlongfootnotefont{\Glsaccesslong{##1}}}%
7046 }%
7047 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
7048 \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7049 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7050 \glxtrparen{\glsfirstlongfootnotefont{\Glsaccesslongpl{##1}}}%
7051 }%
7052 }

```

postfootnote-sc Backward compatibility:

```

7053 \@glxtr@deprecated@abbrstyle{postfootnote-sc}{short-sc-postfootnote}

```

## 1.6.4 Predefined Styles (Fake Small Capitals)

These styles require the relsize package, which must be loaded by the user. These styles all use:

```

\glxtrsmfont Maintained for backward compatibility.
7054 \newcommand*{\glxtrsmfont}[1]{\textsmaller{#1}}

\glsabbrvsmfont Added for consistent naming.
7055 \newcommand*{\glsabbrvsmfont}{\glxtrsmfont}

sxtrfirstsmfont Maintained for backward compatibility.
7056 \newcommand*{\glxtrfirstsmfont}[1]{\glsabbrvsmfont{#1}}

irstabbrvsmfont Added for consistent naming.
7057 \newcommand*{\glsfirstabbrvsmfont}{\glxtrfirstsmfont}

```

and for the default short form suffix:

`\glxtrmsuffix`

```
7058 \newcommand*{\glxtrmsuffix}{\glxtrabbrvpluralsuffix}
```

`long-short-sm`

```
7059 \newabbreviationstyle{long-short-sm}%
7060 {%
7061   \renewcommand*{\CustomAbbreviationFields}{%
7062     name={\protect\glsabbrvsmfont{\the\glsshorttok}},
7063     sort={\the\glsshorttok},
7064     first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
7065       \protect\glxtrfullsep{\the\glslabeltok}%
7066       \glxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshorttok}}},%
7067     firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
7068       \protect\glxtrfullsep{\the\glslabeltok}%
7069       \glxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshortpltok}}},%
7070     plural={\protect\glsabbrvsmfont{\the\glsshortpltok}},%
7071     description={\the\glslongtok}}%
7072   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7073     \glshasattribute{\the\glslabeltok}{regular}%
7074     {%
7075       \glsssetAttribute{\the\glslabeltok}{regular}{false}%
7076     }%
7077   }%
7078 }%
7079 }%
7080 {%
7081   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvsmfont{##1}}%
7082   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvsmfont{##1}}%
7083   \renewcommand*{\abbrvpluralsuffix}{\protect\glxtrmsuffix}%

```

Use the default long fonts.

```
7084 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
7085 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
7086 \renewcommand*{\glxtrfullformat}[2]{%
7087   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrininsertinside##2\fi}%
7088   \ifglxtrininsertinside\else##2\fi
7089   \glxtrfullsep{##1}%
7090   \glxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
7091 }%
7092 \renewcommand*{\glxtrfullplformat}[2]{%
7093   \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrininsertinside##2\fi}%
7094   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7095   \glxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
7096 }%
7097 \renewcommand*{\Glsxtrfullformat}[2]{%
7098   \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrininsertinside##2\fi}%

```

```

7099 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7100 \glxtrparen{\glsfirstabbrvsmfont{\glssaccessshort{##1}}}%
7101 }%
7102 \renewcommand*{\Glsxtrfullplformat}[2]{%
7103 \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7104 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7105 \glxtrparen{\glsfirstabbrvsmfont{\glssaccessshortpl{##1}}}%
7106 }%
7107 }

```

g-short-sm-desc

```

7108 \newabbreviationstyle{long-short-sm-desc}%
7109 {%
7110 \renewcommand*{\CustomAbbreviationFields}{%
7111 name={\glxtrlongshortdescname},
7112 sort={\glxtrlongshortdescsort},%
7113 first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
7114 \protect\glxtrfullsep{\the\glslabeltok}%
7115 \glxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshorttok}}},%
7116 firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
7117 \protect\glxtrfullsep{\the\glslabeltok}%
7118 \glxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshortpltok}}},%
7119 text={\protect\glssabbrvsmfont{\the\glsshorttok}},%
7120 plural={\protect\glssabbrvsmfont{\the\glsshortpltok}}%
7121 }%

```

Unset the regular attribute if it has been set.

```

7122 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7123 \glshasattribute{\the\glslabeltok}{regular}%
7124 {%
7125 \glissetattribute{\the\glslabeltok}{regular}{false}%
7126 }%
7127 {}%
7128 }%
7129 }%
7130 {%

```

As long-short-sm style:

```

7131 \GlsXtrUseAbbrStyleFmts{long-short-sm}%
7132 }

```

short-sm-long Now the short (long) version

```

7133 \newabbreviationstyle{short-sm-long}%
7134 {%
7135 \renewcommand*{\CustomAbbreviationFields}{%
7136 name={\protect\glssabbrvsmfont{\the\glsshorttok}},
7137 sort={\the\glsshorttok},
7138 description={\the\glslongtok},%
7139 first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}%
7140 \protect\glxtrfullsep{\the\glslabeltok}%

```

```

7141 \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
7142 firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}%
7143 \protect\glxtrfullsep{\the\glslabeltok}%
7144 \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
7145 plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%

```

Unset the regular attribute if it has been set.

```

7146 \renewcommand*\GlsXtrPostNewAbbreviation{%
7147 \glshasattribute{\the\glslabeltok}{regular}%
7148 {%
7149 \glssetattribute{\the\glslabeltok}{regular}{false}%
7150 }%
7151 {}%
7152 }%
7153 }%
7154 {%
7155 \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
7156 \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
7157 \renewcommand*\abbrvpluralsuffix{\protect\glxtrsmsuffix}%
7158 \renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
7159 \renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

7160 \renewcommand*\glxtrfullformat[2]{%
7161 \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
7162 \ifglxtrininsertinside\else##2\fi
7163 \glxtrfullsep{##1}%
7164 \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7165 }%
7166 \renewcommand*\glxtrfullplformat[2]{%
7167 \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
7168 \ifglxtrininsertinside\else##2\fi
7169 \glxtrfullsep{##1}%
7170 \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
7171 }%
7172 \renewcommand*\Glsxtrfullformat[2]{%
7173 \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
7174 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7175 \glxtrparen{\glsfirstlongdefaultfont{\Glsaccesslong{##1}}}%
7176 }%
7177 \renewcommand*\Glsxtrfullplformat[2]{%
7178 \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
7179 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7180 \glxtrparen{\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}}}%
7181 }%
7182 }

```

rt-sm-long-desc As before but user provides description

```

7183 \newabbreviationstyle{short-sm-long-desc}%
7184 {%

```

```

7185 \renewcommand*{\CustomAbbreviationFields}{%
7186   name={\glxtrshortlongdescname},
7187   sort={\glxtrshortlongdescsort},
7188   first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}%
7189     \protect\glxtrfullsep{\the\glslabeltok}%
7190     \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
7191   firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}%
7192     \protect\glxtrfullsep{\the\glslabeltok}%
7193     \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
7194   text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
7195   plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
7196 }%

```

Unset the regular attribute if it has been set.

```

7197 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7198   \glshasattribute{\the\glslabeltok}{regular}%
7199   {%
7200     \glssetattribute{\the\glslabeltok}{regular}{false}%
7201   }%
7202   {}%
7203 }%
7204 }%
7205 {%

```

As short-sm-long style:

```

7206 \GlsXtrUseAbbrStyleFmts{short-sm-long}%
7207 }

```

short-sm

```

7208 \newabbreviationstyle{short-sm}%
7209 {%
7210   \renewcommand*{\CustomAbbreviationFields}{%
7211     name={\protect\glsabbrvsmfont{\the\glsshorttok}},
7212     sort={\the\glsshorttok},
7213     first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}},
7214     firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}},
7215     text={\protect\glsabbrvsmfont{\the\glsshorttok}},
7216     plural={\protect\glsabbrvsmfont{\the\glsshortpltok}},
7217     description={\the\glslongtok}}%
7218   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7219     \glssetattribute{\the\glslabeltok}{regular}{true}}%
7220 }%
7221 {%
7222   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvsmfont{##1}}%
7223   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvsmfont{##1}}%
7224   \renewcommand*{\abbrvpluralsuffix}{\protect\glxtrsmsuffix}%
7225   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
7226   \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The inline full form displays the short form followed by the long form in parentheses.

```

7227 \renewcommand*{\glxtrinlinefullformat}[2]{%
7228   \protect\glsfirstabbrvsmfont{\glsaccessshort{##1}%
7229     \ifglxtrininsertinside##2\fi}%
7230   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7231   \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7232 }%
7233 \renewcommand*{\glxtrinlinefullplformat}[2]{%
7234   \protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}%
7235     \ifglxtrininsertinside##2\fi}%
7236   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7237   \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
7238 }%
7239 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
7240   \protect\glsfirstabbrvsmfont{\glsaccessshort{##1}%
7241     \ifglxtrininsertinside##2\fi}%
7242   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7243   \glxtrparen{\glsfirstlongdefaultfont{\Glsaccesslong{##1}}}%
7244 }%
7245 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
7246   \protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}%
7247     \ifglxtrininsertinside##2\fi}%
7248   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7249   \glxtrparen{\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}}}%
7250 }%

```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

7251 \renewcommand*{\glxtrfullformat}[2]{%
7252   \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
7253   \ifglxtrininsertinside\else##2\fi
7254 }%
7255 \renewcommand*{\glxtrfullplformat}[2]{%
7256   \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
7257   \ifglxtrininsertinside\else##2\fi
7258 }%
7259 \renewcommand*{\Glsxtrfullformat}[2]{%
7260   \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
7261   \ifglxtrininsertinside\else##2\fi
7262 }%
7263 \renewcommand*{\Glsxtrfullplformat}[2]{%
7264   \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
7265   \ifglxtrininsertinside\else##2\fi
7266 }%
7267 }

```

short-sm-nolong

```
7268 \letabbreviationstyle{short-sm-nolong}{short-sm}
```

short-sm-desc



```

7269 \newabbreviationstyle{short-sm-desc}%
7270 {%
7271   \renewcommand*{\CustomAbbreviationFields}{%
7272     name={\glstrshortdescname},
7273     sort={\the\glsshorttok},
7274     first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}},
7275     firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}},
7276     text={\protect\glsabbrvsmfont{\the\glsshorttok}},
7277     plural={\protect\glsabbrvsmfont{\the\glsshortpltok}},
7278     description={\the\glslongtok}}%
7279   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7280     \glssetattribute{\the\glslabeltok}{regular}{true}}%
7281 }%
7282 {%
7283   \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
7284   \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
7285   \renewcommand*{\abbrvpluralsuffix}{\protect\glstrsmsuffix}%
7286   \renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
7287   \renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%

```

The inline full form displays the short format followed by the long form in parentheses.

```

7288   \renewcommand*\glsxtrinlinefullformat}[2]{%
7289     \glsfirstabbrvsmfont{\glsaccessshort{##1}}\ifglxtrininsertinside##2\fi}%
7290     \ifglxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7291     \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7292   }%
7293   \renewcommand*\glsxtrinlinefullplformat}[2]{%
7294     \glsfirstabbrvsmfont{\glsaccessshortpl{##1}}\ifglxtrininsertinside##2\fi}%
7295     \ifglxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7296     \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
7297   }%
7298   \renewcommand*\Glsxtrinlinefullformat}[2]{%
7299     \glsfirstabbrvsmfont{\Glsaccessshort{##1}}\ifglxtrininsertinside##2\fi}%
7300     \ifglxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7301     \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7302   }%
7303   \renewcommand*\Glsxtrinlinefullplformat}[2]{%
7304     \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}}\ifglxtrininsertinside##2\fi}%
7305     \ifglxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7306     \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
7307   }%

```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

7308   \renewcommand*\glsxtrfullformat}[2]{%
7309     \glsfirstabbrvsmfont{\glsaccessshort{##1}}\ifglxtrininsertinside##2\fi}%
7310     \ifglxtrininsertinside\else##2\fi
7311   }%
7312   \renewcommand*\glsxtrfullplformat}[2]{%
7313     \glsfirstabbrvsmfont{\glsaccessshortpl{##1}}\ifglxtrininsertinside##2\fi}%

```

```

7314     \ifglxtrinsertinside\else##2\fi
7315 }%
7316 \renewcommand*{\Glsxtrfullformat}[2]{%
7317     \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7318     \ifglxtrinsertinside\else##2\fi
7319 }%
7320 \renewcommand*{\Glsxtrfullplformat}[2]{%
7321     \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7322     \ifglxtrinsertinside\else##2\fi
7323 }%
7324 }

```

-sm-nolong-desc

```

7325 \letabbreviationstyle{short-sm-nolong-desc}{short-sm-desc}

```

long-noshort-sm The smallcaps font will only be used if the short form is explicitly invoked through commands like \glsshort.

```

7326 \newabbreviationstyle{long-noshort-sm}%
7327 {%
7328     \renewcommand*{\CustomAbbreviationFields}{%
7329         name={\protect\glsabbrvsmfont{\the\glsshorttok}},
7330         sort={\the\glsshorttok},
7331         first={\protect\glsfirstlongdefaultfont{\the\glslongtok}},
7332         firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},
7333         text={\protect\glslongdefaultfont{\the\glslongtok}},
7334         plural={\protect\glslongdefaultfont{\the\glslongpltok}},%
7335         description={\the\glslongtok}%
7336     }%
7337     \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7338         \glssetattribute{\the\glslabeltok}{regular}{true}}%
7339 }%
7340 {%
7341     \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
7342     \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
7343     \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtrsmsuffix}%
7344     \renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
7345     \renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

7346 \renewcommand*{\glsxtrsubsequentfmt}[2]{%
7347     \glslongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
7348     \ifglxtrinsertinside \else##2\fi
7349 }%
7350 \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
7351     \glslongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
7352     \ifglxtrinsertinside \else##2\fi
7353 }%
7354 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
7355     \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%

```

```

7356 \ifglxtrinsertinside \else##2\fi
7357 }%
7358 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
7359 \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
7360 \ifglxtrinsertinside \else##2\fi
7361 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

7362 \renewcommand*{\glxtrinlinefullformat}[2]{%
7363 \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
7364 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7365 \glxtrparen{\protect\glslfirstabbrvsmfont{\Glsaccessshort{##1}}}%
7366 }%
7367 \renewcommand*{\glxtrinlinefullplformat}[2]{%
7368 \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7369 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7370 \glxtrparen{\protect\glslfirstabbrvsmfont{\Glsaccessshortpl{##1}}}%
7371 }%
7372 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
7373 \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
7374 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7375 \glxtrparen{\protect\glslfirstabbrvsmfont{\Glsaccessshort{##1}}}%
7376 }%
7377 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
7378 \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7379 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7380 \glxtrparen{\protect\glslfirstabbrvsmfont{\Glsaccessshortpl{##1}}}%
7381 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

7382 \renewcommand*{\glxtrfullformat}[2]{%
7383 \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
7384 \ifglxtrinsertinside\else##2\fi
7385 }%
7386 \renewcommand*{\glxtrfullplformat}[2]{%
7387 \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7388 \ifglxtrinsertinside\else##2\fi
7389 }%
7390 \renewcommand*{\Glsxtrfullformat}[2]{%
7391 \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
7392 \ifglxtrinsertinside\else##2\fi
7393 }%
7394 \renewcommand*{\Glsxtrfullplformat}[2]{%
7395 \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7396 \ifglxtrinsertinside\else##2\fi
7397 }%
7398 }

```

long-sm Backward compatibility:

7399 \@glxtr@deprecated@abbrstyle{long-sm}{long-noshort-sm}

noshort-sm-desc The smaller font will only be used if the short form is explicitly invoked through commands like \glsshort.

```
7400 \newabbreviationstyle{long-noshort-sm-desc}%
7401 {%
7402   \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
7403 }%
7404 {%
7405   \renewcommand*\glabbrvfont[1]{\glabbrvsmfont{##1}}%
7406   \renewcommand*\glfirstabbrvfont[1]{\glfirstabbrvsmfont{##1}}%
7407   \renewcommand*\abbrvpluralsuffix{\protect\glxtrsmsuffix}%
7408   \renewcommand*\glfirstlongfont[1]{\glfirstlongdefaultfont{##1}}%
7409   \renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
7410 \renewcommand*\glxtrsubsequentfmt}[2]{%
7411   \glslongdefaultfont{\glaccesslong{##1}\ifglxtrininsertinside ##2\fi}%
7412   \ifglxtrininsertinside \else##2\fi
7413 }%
7414 \renewcommand*\glxtrsubsequentplfmt}[2]{%
7415   \glslongdefaultfont{\glaccesslongpl{##1}\ifglxtrininsertinside ##2\fi}%
7416   \ifglxtrininsertinside \else##2\fi
7417 }%
7418 \renewcommand*\Glsxtrsubsequentfmt}[2]{%
7419   \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrininsertinside ##2\fi}%
7420   \ifglxtrininsertinside \else##2\fi
7421 }%
7422 \renewcommand*\Glsxtrsubsequentplfmt}[2]{%
7423   \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrininsertinside ##2\fi}%
7424   \ifglxtrininsertinside \else##2\fi
7425 }%
```

The inline full form displays the long format followed by the short form in parentheses.

```
7426 \renewcommand*\glxtrinlinefullformat}[2]{%
7427   \glfirstlongdefaultfont{\glaccesslong{##1}\ifglxtrininsertinside##2\fi}%
7428   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7429   \glxtrparen{\protect\glfirstabbrvsmfont{\glaccessshort{##1}}}%
7430 }%
7431 \renewcommand*\glxtrinlinefullplformat}[2]{%
7432   \glfirstlongdefaultfont{\glaccesslongpl{##1}\ifglxtrininsertinside##2\fi}%
7433   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7434   \glxtrparen{\protect\glfirstabbrvsmfont{\glaccessshortpl{##1}}}%
7435 }%
7436 \renewcommand*\Glsxtrinlinefullformat}[2]{%
7437   \glfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrininsertinside##2\fi}%
7438   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7439   \glxtrparen{\protect\glfirstabbrvsmfont{\glaccessshort{##1}}}%
7440 }%
7441 \renewcommand*\Glsxtrinlinefullplformat}[2]{%

```

```

7442 \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7443 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7444 \glxtrparen{\protect\glsfirstabbrvsmfont{\Glsaccessshortpl{##1}}}%
7445 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

7446 \renewcommand*{\glxtrfullformat}[2]{%
7447 \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
7448 \ifglxtrinsertinside\else##2\fi
7449 }%
7450 \renewcommand*{\glxtrfullplformat}[2]{%
7451 \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7452 \ifglxtrinsertinside\else##2\fi
7453 }%
7454 \renewcommand*{\Glsxtrfullformat}[2]{%
7455 \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
7456 \ifglxtrinsertinside\else##2\fi
7457 }%
7458 \renewcommand*{\Glsxtrfullplformat}[2]{%
7459 \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7460 \ifglxtrinsertinside\else##2\fi
7461 }%
7462 }

```

long-desc-sm Backward compatibility:

```

7463 \@glxtr@deprecated@abbrstyle{long-desc-sm}{long-noshort-sm-desc}

```

ort-sm-footnote

```

7464 \newabbreviationstyle{short-sm-footnote}%
7465 {%
7466 \renewcommand*{\CustomAbbreviationFields}{%
7467 name={\protect\glsabbrvsmfont{\the\glsshorttok}},
7468 sort={\the\glsshorttok},
7469 description={\the\glslongtok},%
7470 first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}}%
7471 \protect\glxtrabbrvfootnote{\the\glslabeltok}%
7472 {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
7473 firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}}%
7474 \protect\glxtrabbrvfootnote{\the\glslabeltok}%
7475 {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
7476 plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%

```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the regular attribute if it has been set.

```

7477 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7478 \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
7479 \glsattribute{\the\glslabeltok}{regular}%
7480 }%
7481 \glssetattribute{\the\glslabeltok}{regular}{false}%

```

```

7482 }%
7483 {}%
7484 }%
7485 }%
7486 {%
7487 \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
7488 \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
7489 \renewcommand*\abbrvpluralsuffix{\protect\glsxtrmsuffix}%
7490 \renewcommand*\glsfirstlongfont[1]{\glsfirstlongfootnotefont{##1}}%
7491 \renewcommand*\glslongfont[1]{\glslongfootnotefont{##1}}%

```

The full format displays the short form followed by the long form as a footnote.

```

7492 \renewcommand*\glsxtrfullformat[2]{%
7493   \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7494   \ifglsxtrininsertinside\else##2\fi
7495   \protect\glsxtrabbrvfootnote{##1}%
7496   {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
7497 }%
7498 \renewcommand*\glsxtrfullplformat[2]{%
7499   \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7500   \ifglsxtrininsertinside\else##2\fi
7501   \protect\glsxtrabbrvfootnote{##1}%
7502   {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
7503 }%
7504 \renewcommand*\Glsxtrfullformat[2]{%
7505   \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7506   \ifglsxtrininsertinside\else##2\fi
7507   \protect\glsxtrabbrvfootnote{##1}%
7508   {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
7509 }%
7510 \renewcommand*\Glsxtrfullplformat[2]{%
7511   \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7512   \ifglsxtrininsertinside\else##2\fi
7513   \protect\glsxtrabbrvfootnote{##1}%
7514   {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
7515 }%

```

The first use full form and the inline full form use the short (long) style.

```

7516 \renewcommand*\glsxtrininlinefullformat[2]{%
7517   \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7518   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7519   \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
7520 }%
7521 \renewcommand*\glsxtrininlinefullplformat[2]{%
7522   \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7523   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7524   \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
7525 }%
7526 \renewcommand*\Glsxtrininlinefullformat[2]{%
7527   \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%

```

```

7528     \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7529     \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
7530 }%
7531 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
7532     \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7533     \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7534     \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
7535 }%
7536 }

```

footnote-sm Backward compatibility:

```

7537 \@glxtr@deprecated@abbrstyle{footnote-sm}{short-sm-footnote}

```

sm-postfootnote

```

7538 \newabbreviationstyle{short-sm-postfootnote}%
7539 {%
7540     \renewcommand*{\CustomAbbreviationFields}{%
7541         name={\protect\glssabbrvsmfont{\the\glssshorttok}},
7542         sort={\the\glssshorttok},
7543         description={\the\glslongtok},%
7544         first={\protect\glsfirstabbrvsmfont{\the\glssshorttok}},%
7545         firstplural={\protect\glsfirstabbrvsmfont{\the\glssshortpltok}},%
7546         plural={\protect\glssabbrvsmfont{\the\glssshortpltok}}}%

```

Make this category insert a footnote after the link if this was the first use, and unset the regular attribute if it has been set.

```

7547 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7548     \csdef{glxtrpostlink\glscategorylabel}{%
7549         \glxtrifwasfirstuse
7550         {%

```

Needs the specific font command here as the style may have been lost by the time the footnote occurs.

```

7551         \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}%
7552             {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
7553     }%
7554     {}%
7555 }%
7556 \glshasattribute{\the\glslabeltok}{regular}%
7557 {%
7558     \glissetattribute{\the\glslabeltok}{regular}{false}%
7559 }%
7560 {}%
7561 }%

```

The footnote needs to be suppressed in the inline form, so `\glxtrfull` must set the first use switch off.

```

7562 \renewcommand*{\glxtrsetupfulldefs}{%
7563     \let\glxtrifwasfirstuse\@secondoftwo
7564 }%

```

```

7565 }%
7566 {%
7567   \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
7568   \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
7569   \renewcommand*\abbrvpluralsuffix{\protect\glsxrtrmsuffix}%
7570   \renewcommand*\glsfirstlongfont[1]{\glsfirstlongfootnotefont{##1}}%
7571   \renewcommand*\glslongfont[1]{\glslongfootnotefont{##1}}%

```

The full format displays the short form. The long form is deferred.

```

7572   \renewcommand*\glsxtrfullformat}[2]{%
7573     \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7574     \ifglsxtrininsertinside\else##2\fi
7575   }%
7576   \renewcommand*\glsxtrfullplformat}[2]{%
7577     \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7578     \ifglsxtrininsertinside\else##2\fi
7579   }%
7580   \renewcommand*\Glsxtrfullformat}[2]{%
7581     \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7582     \ifglsxtrininsertinside\else##2\fi
7583   }%
7584   \renewcommand*\Glsxtrfullplformat}[2]{%
7585     \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7586     \ifglsxtrininsertinside\else##2\fi
7587   }%

```

The first use full form and the inline full form use the short (long) style.

```

7588   \renewcommand*\glsxtrinlinefullformat}[2]{%
7589     \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7590     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7591     \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
7592   }%
7593   \renewcommand*\glsxtrinlinefullplformat}[2]{%
7594     \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7595     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7596     \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
7597   }%
7598   \renewcommand*\Glsxtrinlinefullformat}[2]{%
7599     \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7600     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7601     \glsxtrparen{\glsfirstlongfootnotefont{\Glsaccesslong{##1}}}%
7602   }%
7603   \renewcommand*\Glsxtrinlinefullplformat}[2]{%
7604     \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7605     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7606     \glsxtrparen{\glsfirstlongfootnotefont{\Glsaccesslongpl{##1}}}%
7607   }%
7608 }

```

postfootnote-sm Backward compatibility:



7609 \@glxtr@deprecated@abbrstyle{postfootnote-sm}{short-sm-postfootnote}

## 1.6.5 Predefined Styles (Emphasized)

These styles use \emph for the short form.

\glsabbrvemfont

7610 \newcommand\*{\glsabbrvemfont}[1]{\emph{#1}}%

firstabbrvemfont

7611 \newcommand\*{\glsfirstabbrvemfont}[1]{\glsabbrvemfont{#1}}%

The default short form suffix:

\glxtremsuffix

7612 \newcommand\*{\glxtremsuffix}{\glxtrabbrvpluralsuffix}

firstlongemfont Only used by the “long-em” styles.

7613 \newcommand\*{\glsfirstlongemfont}[1]{\glslongemfont{#1}}%

\glslongemfont Only used by the “long-em” styles.

7614 \newcommand\*{\glslongemfont}[1]{\emph{#1}}%

long-short-em The long form is just set in the default long font.

7615 \newabbreviationstyle{long-short-em}%

7616 {%

7617 \renewcommand\*{\CustomAbbreviationFields}{%

7618 name={\protect\glsabbrvemfont{\the\glsshorttok}},

7619 sort={\the\glsshorttok},

7620 first={\protect\glsfirstlongdefaultfont{\the\glslongtok}}%

7621 \protect\glxtrfullsep{\the\glslabeltok}}%

7622 \glxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%

7623 firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}}%

7624 \protect\glxtrfullsep{\the\glslabeltok}}%

7625 \glxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%

7626 plural={\protect\glsabbrvemfont{\the\glsshortpltok}}},%

7627 description={\the\glslongtok}}%

7628 \renewcommand\*{\GlsXtrPostNewAbbreviation}{%

7629 \glshasattribute{\the\glslabeltok}{regular}}%

7630 {%

7631 \glsselattribute{\the\glslabeltok}{regular}{false}}%

7632 }%

7633 {}%

7634 }%

7635 }%

7636 {%

7637 \renewcommand\*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%

7638 \renewcommand\*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%

7639 \renewcommand\*{\abbrvpluralsuffix}{\protect\glxtremsuffix}}%

Use the default long fonts.

```
7640 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
7641 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
7642 \renewcommand*{\glsxtrfullformat}[2]{%
7643   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
7644   \ifglsxtrinsertinside\else##2\fi
7645   \glsxtrfullsep{##1}%
7646   \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
7647 }%
7648 \renewcommand*{\glsxtrfullplformat}[2]{%
7649   \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
7650   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
7651   \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
7652 }%
7653 \renewcommand*{\Glsxtrfullformat}[2]{%
7654   \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
7655   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
7656   \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
7657 }%
7658 \renewcommand*{\Glsxtrfullplformat}[2]{%
7659   \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
7660   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
7661   \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
7662 }%
7663 }
```

g-short-em-desc

```
7664 \newabbreviationstyle{long-short-em-desc}%
7665 {%
7666   \renewcommand*{\CustomAbbreviationFields}{%
7667     name={\glsxtrlongshortdescname},
7668     sort={\glsxtrlongshortdescsort},%
7669     first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
7670       \protect\glsxtrfullsep{\the\glslabeltok}%
7671       \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
7672     firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
7673       \protect\glsxtrfullsep{\the\glslabeltok}%
7674       \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
7675     text={\protect\glsabbrvemfont{\the\glsshorttok}},%
7676     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}%
7677   }%
```

Unset the regular attribute if it has been set.

```
7678 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7679   \glshasattribute{\the\glslabeltok}{regular}%
7680   {%
7681     \glssetattribute{\the\glslabeltok}{regular}{false}%
7682   }%
```

```

7683     {}%
7684   }%
7685 }%
7686 {%

```

As long-short-em style:

```

7687   \GlsXtrUseAbbrStyleFmts{long-short-em}%
7688 }

```

ong-em-short-em

```

7689 \newabbreviationstyle{long-em-short-em}%
7690 {%

```

\glslongemfont is used in the description since \glsdesc doesn't set the style.

```

7691   \renewcommand*{\CustomAbbreviationFields}{%
7692     name={\protect\glsabbrvemfont{\the\glsshorttok}},
7693     sort={\the\glsshorttok},
7694     first={\protect\glsfirstlongemfont{\the\glslongtok}}%
7695     \protect\glsxtrfullsep{\the\glslabeltok}%
7696     \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
7697     firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}}%
7698     \protect\glsxtrfullsep{\the\glslabeltok}%
7699     \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%

7700     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}},%
7701     description={\protect\glslongemfont{\the\glslongtok}}}%

```

Unset the regular attribute if it has been set.

```

7702   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7703     \glsattribute{\the\glslabeltok}{regular}%
7704     {%
7705       \glssetattribute{\the\glslabeltok}{regular}{false}%
7706     }%
7707   }%
7708 }%
7709 }%
7710 {%
7711   \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtremsuffix}%
7712   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
7713   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
7714   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
7715   \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

7716   \renewcommand*{\glsxtrfullformat}[2]{%
7717     \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
7718     \ifglsxtrinsertinside\else##2\fi
7719     \glsxtrfullsep{##1}%
7720     \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
7721   }%
7722   \renewcommand*{\glsxtrfullplformat}[2]{%

```

```

7723 \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7724 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7725 \glxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
7726 }%
7727 \renewcommand*{\Glsxtrfullformat}[2]{%
7728 \glsfirstlongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
7729 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7730 \glxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
7731 }%
7732 \renewcommand*{\Glsxtrfullplformat}[2]{%
7733 \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
7734 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7735 \glxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
7736 }%
7737 }

```

m-short-em-desc

```

7738 \newabbreviationstyle{long-em-short-em-desc}%
7739 {%
7740 \renewcommand*{\CustomAbbreviationFields}{%
7741 name={\glxtrlongshortdescname},
7742 sort={\glxtrlongshortdescsort},%
7743 first={\protect\glsfirstlongemfont{\the\glslongtok}%
7744 \protect\glxtrfullsep{\the\glslabeltok}%
7745 \glxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
7746 firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}%
7747 \protect\glxtrfullsep{\the\glslabeltok}%
7748 \glxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
7749 text={\protect\glsabbrvemfont{\the\glsshorttok}},%
7750 plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
7751 }%

```

Unset the regular attribute if it has been set.

```

7752 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7753 \glshasattribute{\the\glslabeltok}{regular}%
7754 {%
7755 \glssetattribute{\the\glslabeltok}{regular}{false}%
7756 }%
7757 {}%
7758 }%
7759 }%
7760 {%
7761 \GlsXtrUseAbbrStyleFmts{long-em-short-em}%
7762 }

```

short-em-long Now the short (long) version

```

7763 \newabbreviationstyle{short-em-long}%
7764 {%
7765 \renewcommand*{\CustomAbbreviationFields}{%
7766 name={\protect\glsabbrvemfont{\the\glsshorttok}},

```

```

7767 sort={\the\glsshorttok},
7768 description={\the\glslongtok},%
7769 first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
7770 \protect\glsxtrfullsep{\the\glslabeltok}%
7771 \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
7772 firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
7773 \protect\glsxtrfullsep{\the\glslabeltok}%
7774 \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
7775 plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%

```

Unset the regular attribute if it has been set.

```

7776 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7777 \glsattribute{\the\glslabeltok}{regular}%
7778 {%
7779 \glssetattribute{\the\glslabeltok}{regular}{false}%
7780 }%
7781 {}%
7782 }%
7783 }%
7784 {%

```

Mostly as short-long style:

```

7785 \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtrsuffix}%
7786 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
7787 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
7788 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
7789 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

7790 \renewcommand*{\glsxtrfullformat}[2]{%
7791 \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7792 \ifglxtrinsertinside\else##2\fi
7793 \glsxtrfullsep{##1}%
7794 \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7795 }%
7796 \renewcommand*{\glsxtrfullplformat}[2]{%
7797 \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7798 \ifglxtrinsertinside\else##2\fi
7799 \glsxtrfullsep{##1}%
7800 \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
7801 }%
7802 \renewcommand*{\Glsxtrfullformat}[2]{%
7803 \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7804 \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
7805 \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7806 }%
7807 \renewcommand*{\Glsxtrfullplformat}[2]{%
7808 \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7809 \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
7810 \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
7811 }%

```

7812 }

rt-em-long-desc As before but user provides description

```
7813 \newabbreviationstyle{short-em-long-desc}%
7814 {%
7815   \renewcommand*{\CustomAbbreviationFields}{%
7816     name={\glstrshortlongdescname},
7817     sort={\glstrshortlongdescsort},
7818     first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
7819       \protect\glstrfullsep{\the\glslabeltok}%
7820       \glstrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
7821     firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
7822       \protect\glstrfullsep{\the\glslabeltok}%
7823       \glstrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
7824     text={\protect\glsabbrvemfont{\the\glsshorttok}},%
7825     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
7826   }%
```

Unset the regular attribute if it has been set.

```
7827 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7828   \glshasattribute{\the\glslabeltok}{regular}%
7829   {%
7830     \glissetattribute{\the\glslabeltok}{regular}{false}%
7831   }%
7832   {}%
7833 }%
7834 }%
7835 {%
7836   \GlsXtrUseAbbrStyleFmts{short-em-long}%
7837 }
```

hort-em-long-em

```
7838 \newabbreviationstyle{short-em-long-em}%
7839 {%
7840   \glslongemfont is used in the description since \glsdesc doesn't set the style.
7841   \renewcommand*{\CustomAbbreviationFields}{%
7842     name={\protect\glsabbrvemfont{\the\glsshorttok}},
7843     sort={\the\glsshorttok},
7844     description={\protect\glslongemfont{\the\glslongtok}},%
7845     first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
7846       \protect\glstrfullsep{\the\glslabeltok}%
7847       \glstrparen{\protect\glsfirstlongemfont{\the\glslongtok}}},%
7848     firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
7849       \protect\glstrfullsep{\the\glslabeltok}%
7850       \glstrparen{\protect\glsfirstlongemfont{\the\glslongpltok}}},%
7851     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
7852   }%
```

Unset the regular attribute if it has been set.

```

7851 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7852   \glshasattribute{\the\glslabeltok}{regular}}%
7853   {%
7854     \glsssetAttribute{\the\glslabeltok}{regular}{false}%
7855   }%
7856   {}%
7857 }%
7858 }%
7859 {%
7860 \renewcommand*{\abbrvpluralsuffix}{\protect\glxxtremsuffix}%
7861 \renewcommand*{\glssabbrvfont}[1]{\glssabbrvemfont{##1}}%
7862 \renewcommand*{\glssfirstabbrvfont}[1]{\glssfirstabbrvemfont{##1}}%
7863 \renewcommand*{\glssfirstlongfont}[1]{\glssfirstlongemfont{##1}}%
7864 \renewcommand*{\glsslongfont}[1]{\glsslongemfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

7865 \renewcommand*{\glsxtrfullformat}[2]{%
7866   \glssfirstabbrvemfont{\glssaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7867   \ifglsxtrininsertinside\else##2\fi
7868   \glsxtrfullsep{##1}%
7869   \glsxtrparen{\glssfirstlongemfont{\glssaccesslong{##1}}}%
7870 }%
7871 \renewcommand*{\glsxtrfullplformat}[2]{%
7872   \glssfirstabbrvemfont{\glssaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7873   \ifglsxtrininsertinside\else##2\fi
7874   \glsxtrfullsep{##1}%
7875   \glsxtrparen{\glssfirstlongemfont{\glssaccesslongpl{##1}}}%
7876 }%
7877 \renewcommand*{\Glsxtrfullformat}[2]{%
7878   \glssfirstabbrvemfont{\glssaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
7879   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7880   \glsxtrparen{\glssfirstlongemfont{\glssaccesslong{##1}}}%
7881 }%
7882 \renewcommand*{\Glsxtrfullplformat}[2]{%
7883   \glssfirstabbrvemfont{\glssaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
7884   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7885   \glsxtrparen{\glssfirstlongemfont{\glssaccesslongpl{##1}}}%
7886 }%
7887 }

```

em-long-em-desc

```

7888 \newabbreviationstyle{short-em-long-em-desc}%
7889 {%
7890 \renewcommand*{\CustomAbbreviationFields}{%
7891   name={\glsxtrshortlongdescname},%
7892   sort={\glsxtrshortlongdescsort},%
7893   first={\protect\glssfirstabbrvemfont{\the\glssshorttok}}%
7894   \protect\glsxtrfullsep{\the\glslabeltok}}%
7895   \glsxtrparen{\protect\glssfirstlongemfont{\the\glsslongtok}}},%
7896   firstplural={\protect\glssfirstabbrvemfont{\the\glssshortpltok}}%

```

```

7897 \protect\glxtrfullsep{\the\glslabeltok}%
7898 \glxtrparen{\protect\glsfirstlongemfont{\the\glslongpltok}}},%
7899 text={\protect\glsabbrvemfont{\the\glsshorttok}},%
7900 plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
7901 }%

```

Unset the regular attribute if it has been set.

```

7902 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7903 \glshasattribute{\the\glslabeltok}{regular}%
7904 {%
7905 \glissetattribute{\the\glslabeltok}{regular}{false}%
7906 }%
7907 {}%
7908 }%
7909 }%
7910 {%
7911 \GlsXtrUseAbbrStyleFmts{short-em-long-em}%
7912 }

```

short-em

```

7913 \newabbreviationstyle{short-em}%
7914 {%
7915 \renewcommand*{\CustomAbbreviationFields}{%
7916 name={\protect\glsabbrvemfont{\the\glsshorttok}},
7917 sort={\the\glsshorttok},
7918 first={\protect\glsfirstabbrvemfont{\the\glsshorttok}},
7919 firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}},
7920 text={\protect\glsabbrvemfont{\the\glsshorttok}},
7921 plural={\protect\glsabbrvemfont{\the\glsshortpltok}},
7922 description={\the\glslongtok}}%
7923 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7924 \glissetattribute{\the\glslabeltok}{regular}{true}}%
7925 }%
7926 {%
7927 \renewcommand*{\abbrvpluralsuffix}{\protect\glxtremsuffix}%
7928 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
7929 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
7930 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
7931 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The inline full form displays the short form followed by the long form in parentheses.

```

7932 \renewcommand*{\glxtrinlinefullformat}[2]{%
7933 \protect\glsfirstabbrvemfont{\glsaccessshort{##1}}%
7934 \ifglxtrininsertinside##2\fi}%
7935 \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
7936 \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7937 }%
7938 \renewcommand*{\glxtrinlinefullplformat}[2]{%
7939 \protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}%
7940 \ifglxtrininsertinside##2\fi}%

```



```

7941 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7942 \glxtrparen{\glsfirstlongdefaultfont{\glaccesslongpl{##1}}}%
7943 }%
7944 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
7945 \protect\glsfirstabbrvemfont{\glaccessshort{##1}%
7946 \ifglxtrinsertinside##2\fi}%
7947 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7948 \glxtrparen{\glsfirstlongdefaultfont{\Glsaccesslong{##1}}}%
7949 }%
7950 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
7951 \protect\glsfirstabbrvemfont{\glaccessshortpl{##1}%
7952 \ifglxtrinsertinside##2\fi}%
7953 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
7954 \glxtrparen{\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}}}%
7955 }%

```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

7956 \renewcommand*{\glxtrfullformat}[2]{%
7957 \glsfirstabbrvemfont{\glaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7958 \ifglxtrinsertinside\else##2\fi
7959 }%
7960 \renewcommand*{\glxtrfullplformat}[2]{%
7961 \glsfirstabbrvemfont{\glaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7962 \ifglxtrinsertinside\else##2\fi
7963 }%
7964 \renewcommand*{\Glsxtrfullformat}[2]{%
7965 \glsfirstabbrvemfont{\glaccessshort{##1}\ifglxtrinsertinside##2\fi}%
7966 \ifglxtrinsertinside\else##2\fi
7967 }%
7968 \renewcommand*{\Glsxtrfullplformat}[2]{%
7969 \glsfirstabbrvemfont{\glaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
7970 \ifglxtrinsertinside\else##2\fi
7971 }%
7972 }

```

short-em-nolong

```

7973 \letabbreviationstyle{short-em-nolong}{short-em}

```

short-em-desc

```

7974 \newabbreviationstyle{short-em-desc}%
7975 {%
7976 \renewcommand*{\CustomAbbreviationFields}{%
7977 name={\glxtrshortdescname},
7978 sort={\the\glsshorttok},
7979 first={\protect\glsfirstabbrvemfont{\the\glsshorttok}},
7980 firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}},
7981 text={\protect\glsabbrvemfont{\the\glsshorttok}},
7982 plural={\protect\glsabbrvemfont{\the\glsshortpltok}},

```

```

7983     description={\the\glslongtok}}%
7984 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
7985     \glssetattribute{\the\glslabeltok}{regular}{true}}%
7986 }%
7987 {%
7988 \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtremsuffix}%
7989 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
7990 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
7991 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
7992 \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The inline full form displays the short format followed by the long form in parentheses.

```

7993 \renewcommand*{\glsxtrinlinefullformat}[2]{%
7994     \glsfirstabbrvemfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
7995     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
7996 \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
7997 }%
7998 \renewcommand*{\glsxtrinlinefullplformat}[2]{%
7999     \glsfirstabbrvemfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
8000     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8001 \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
8002 }%
8003 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
8004     \glsfirstabbrvemfont{\Glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
8005     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8006 \glsxtrparen{\glsfirstlongdefaultfont{\Glsaccesslong{##1}}}%
8007 }%
8008 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
8009     \glsfirstabbrvemfont{\Glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
8010     \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8011 \glsxtrparen{\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}}}%
8012 }%

```

The first use full form only displays the short form, but it typically won't be used as the regular attribute is set by this style.

```

8013 \renewcommand*{\glsxtrfullformat}[2]{%
8014     \glsfirstabbrvemfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
8015     \ifglsxtrininsertinside\else##2\fi
8016 }%
8017 \renewcommand*{\glsxtrfullplformat}[2]{%
8018     \glsfirstabbrvemfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
8019     \ifglsxtrininsertinside\else##2\fi
8020 }%
8021 \renewcommand*{\Glsxtrfullformat}[2]{%
8022     \glsfirstabbrvemfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
8023     \ifglsxtrininsertinside\else##2\fi
8024 }%
8025 \renewcommand*{\Glsxtrfullplformat}[2]{%
8026     \glsfirstabbrvemfont{\Glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
8027     \ifglsxtrininsertinside\else##2\fi

```

```

8028 }%
8029 }

```

-em-nolong-desc

```

8030 \letabbreviationstyle{short-em-nolong-desc}{short-em-desc}

```

long-noshort-em The short form is explicitly invoked through commands like \glssshort.

```

8031 \newabbreviationstyle{long-noshort-em}%
8032 {%
8033   \renewcommand*{\CustomAbbreviationFields}{%
8034     name={\protect\glsabbrvemfont{\the\glssshorttok}},
8035     sort={\the\glssshorttok},
8036     first={\protect\glsfirstlongdefaultfont{\the\glslongtok}},
8037     firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},
8038     text={\protect\glslongdefaultfont{\the\glslongtok}},
8039     plural={\protect\glslongdefaultfont{\the\glslongpltok}},%
8040     description={\the\glslongtok}%
8041   }%
8042   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8043     \glssetattribute{\the\glslabeltok}{regular}{true}}%
8044 }%
8045 {%
8046   \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtremsuffix}%
8047   \renewcommand*{\glsabbrvfnt}[1]{\glsabbrvemfont{##1}}%
8048   \renewcommand*{\glsfirstabbrvfnt}[1]{\glsfirstabbrvemfont{##1}}%
8049   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
8050   \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

8051 \renewcommand*{\glsxtrsubsequentfmt}[2]{%
8052   \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
8053   \ifglsxtrininsertinside \else##2\fi
8054 }%
8055 \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
8056   \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
8057   \ifglsxtrininsertinside \else##2\fi
8058 }%
8059 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
8060   \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
8061   \ifglsxtrininsertinside \else##2\fi
8062 }%
8063 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
8064   \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
8065   \ifglsxtrininsertinside \else##2\fi
8066 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

8067 \renewcommand*{\glsxtrinlinefullformat}[2]{%
8068   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8069   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%

```

```

8070 \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
8071 }%
8072 \renewcommand*{\glsxtrinlinefullplformat}[2]{%
8073 \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
8074 \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
8075 \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
8076 }%
8077 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
8078 \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
8079 \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
8080 \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
8081 }%
8082 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
8083 \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
8084 \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
8085 \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
8086 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

8087 \renewcommand*{\glsxtrfullformat}[2]{%
8088 \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
8089 \ifglsxtrinsertinside\else##2\fi
8090 }%
8091 \renewcommand*{\glsxtrfullplformat}[2]{%
8092 \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
8093 \ifglsxtrinsertinside\else##2\fi
8094 }%
8095 \renewcommand*{\Glsxtrfullformat}[2]{%
8096 \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
8097 \ifglsxtrinsertinside\else##2\fi
8098 }%
8099 \renewcommand*{\Glsxtrfullplformat}[2]{%
8100 \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
8101 \ifglsxtrinsertinside\else##2\fi
8102 }%
8103 }

```

long-em Backward compatibility:

```

8104 \@glsxtr@deprecated@abbrstyle{long-em}{long-noshort-em}

```

g-em-noshort-em The short form is explicitly invoked through commands like \glsshort.

```

8105 \newabbreviationstyle{long-em-noshort-em}%
8106 {%
8107 \renewcommand*{\CustomAbbreviationFields}{%
8108 name={\protect\glsabbrvemfont{\the\glsshorttok}},
8109 sort={\the\glsshorttok},
8110 first={\protect\glsfirstlongemfont{\the\glslongtok}},
8111 firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}},

```

```

8112   text={\protect\glslongemfont{\the\glslongtok}},
8113   plural={\protect\glslongemfont{\the\glslongpltok}},%
8114   description={\protect\glslongemfont{\the\glslongtok}}%
8115 }%
8116 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8117   \glssetattribute{\the\glslabeltok}{regular}{true}}%
8118 }%
8119 {%
8120 \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtremsuffix}%
8121 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
8122 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
8123 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
8124 \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

8125 \renewcommand*{\glsxtrsubsequentfmt}[2]{%
8126   \glslongemfont{\glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
8127   \ifglsxtrininsertinside \else##2\fi
8128 }%
8129 \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
8130   \glslongemfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
8131   \ifglsxtrininsertinside \else##2\fi
8132 }%
8133 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
8134   \glslongemfont{\Glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
8135   \ifglsxtrininsertinside \else##2\fi
8136 }%
8137 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
8138   \glslongemfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
8139   \ifglsxtrininsertinside \else##2\fi
8140 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

8141 \renewcommand*{\glsxtrininlinefullformat}[2]{%
8142   \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8143   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8144   \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
8145 }%
8146 \renewcommand*{\glsxtrininlinefullplformat}[2]{%
8147   \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
8148   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8149   \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
8150 }%
8151 \renewcommand*{\Glsxtrininlinefullformat}[2]{%
8152   \glsfirstlongemfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8153   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8154   \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
8155 }%
8156 \renewcommand*{\Glsxtrininlinefullplformat}[2]{%
8157   \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%

```

```

8158     \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8159     \glxtrparen{\protect\glsfirstabbrvemfont{\glssaccessshortpl{##1}}}%
8160 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

8161 \renewcommand*{\glxtrfullformat}[2]{%
8162   \glsfirstlongemfont{\glssaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8163   \ifglxtrinsertinside\else##2\fi
8164 }%
8165 \renewcommand*{\glxtrfullplformat}[2]{%
8166   \glsfirstlongemfont{\glssaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8167   \ifglxtrinsertinside\else##2\fi
8168 }%
8169 \renewcommand*{\Glsxtrfullformat}[2]{%
8170   \glsfirstlongemfont{\glssaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8171   \ifglxtrinsertinside\else##2\fi
8172 }%
8173 \renewcommand*{\Glsxtrfullplformat}[2]{%
8174   \glsfirstlongemfont{\glssaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8175   \ifglxtrinsertinside\else##2\fi
8176 }%
8177 }

```

**noshort-em-noreg** Like long-em-noshort-em but doesn't set the regular attribute.

```

8178 \newabbreviationstyle{long-em-noshort-em-noreg}%
8179 {%
8180   \GlsXtrUseAbbrStyleSetup{long-em-noshort-em}%

```

Unset the regular attribute if it has been set.

```

8181 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8182   \glshasattribute{\the\glslabeltok}{regular}%
8183   {%
8184     \glsssetattribute{\the\glslabeltok}{regular}{false}%
8185   }%
8186   {}%
8187 }%
8188 }%
8189 {%
8190   \GlsXtrUseAbbrStyleFmts{long-em-noshort-em}%
8191 }

```

**noshort-em-desc** The emphasized font will only be used if the short form is explicitly invoked through commands like `\glssshort`.

```

8192 \newabbreviationstyle{long-noshort-em-desc}%
8193 {%
8194   \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
8195 }%
8196 {%
8197   \renewcommand*{\abbrvpluralsuffix}{\protect\glstxtremsuffix}%

```

```

8198 \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
8199 \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvemfont{##1}}%
8200 \renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
8201 \renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

8202 \renewcommand*\glsxtrsubsequentfmt[2]{%
8203   \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
8204   \ifglsxtrininsertinside \else##2\fi
8205 }%
8206 \renewcommand*\glsxtrsubsequentplfmt[2]{%
8207   \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
8208   \ifglsxtrininsertinside \else##2\fi
8209 }%
8210 \renewcommand*\Glsxtrsubsequentfmt[2]{%
8211   \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
8212   \ifglsxtrininsertinside \else##2\fi
8213 }%
8214 \renewcommand*\Glsxtrsubsequentplfmt[2]{%
8215   \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
8216   \ifglsxtrininsertinside \else##2\fi
8217 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

8218 \renewcommand*\glsxtrinlinefullformat[2]{%
8219   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8220   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8221   \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
8222 }%
8223 \renewcommand*\glsxtrinlinefullplformat[2]{%
8224   \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
8225   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8226   \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
8227 }%
8228 \renewcommand*\Glsxtrinlinefullformat[2]{%
8229   \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8230   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8231   \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
8232 }%
8233 \renewcommand*\Glsxtrinlinefullplformat[2]{%
8234   \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
8235   \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
8236   \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
8237 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

8238 \renewcommand*\glsxtrfullformat[2]{%
8239   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8240   \ifglsxtrininsertinside\else##2\fi
8241 }%

```

```

8242 \renewcommand*{\glxtrfullplformat}[2]{%
8243   \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8244   \ifglxtrinsertinside\else##2\fi
8245 }%
8246 \renewcommand*{\Glsxtrfullformat}[2]{%
8247   \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8248   \ifglxtrinsertinside\else##2\fi
8249 }%
8250 \renewcommand*{\Glsxtrfullplformat}[2]{%
8251   \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8252   \ifglxtrinsertinside\else##2\fi
8253 }%
8254 }

```

long-desc-em    Backward compatibility:

```
8255 \@glxtr@deprecated@abbrstyle{long-desc-em}{long-noshort-em-desc}
```

noshort-em-desc    The short form is explicitly invoked through commands like `\glsshort`. The long form is emphasized.

```

8256 \newabbreviationstyle{long-em-noshort-em-desc}%
8257 {%
8258   \renewcommand*{\CustomAbbreviationFields}{%
8259     name={\protect\protect\glslongemfont{\the\glslongtok}},
8260     sort={\the\glslongtok},
8261     first={\protect\glsfirstlongemfont{\the\glslongtok}},
8262     firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}},
8263     text={\glslongemfont{\the\glslongtok}},
8264     plural={\glslongemfont{\the\glslongpltok}}}%
8265   }%
8266   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8267     \glssetattribute{\the\glslabeltok}{regular}{true}}%
8268 }%
8269 {%
8270   \renewcommand*{\abbrvpluralsuffix}{\protect\glxtremsuffix}%
8271   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
8272   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
8273   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
8274   \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

8275 \renewcommand*{\glxtrsubsequentfmt}[2]{%
8276   \glslongemfont{\glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
8277   \ifglxtrinsertinside \else##2\fi
8278 }%
8279 \renewcommand*{\glxtrsubsequentplfmt}[2]{%
8280   \glslongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
8281   \ifglxtrinsertinside \else##2\fi
8282 }%
8283 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%

```



```

8284 \glslongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
8285 \ifglxtrinsertinside \else##2\fi
8286 }%
8287 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
8288 \glslongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
8289 \ifglxtrinsertinside \else##2\fi
8290 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

8291 \renewcommand*{\glxtrinlinefullformat}[2]{%
8292 \glsfirstlongemfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8293 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8294 \glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
8295 }%
8296 \renewcommand*{\glxtrinlinefullplformat}[2]{%
8297 \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8298 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8299 \glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
8300 }%
8301 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
8302 \glsfirstlongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8303 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8304 \glxtrparen{\protect\glsfirstabbrvemfont{\Glsaccessshort{##1}}}%
8305 }%
8306 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
8307 \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8308 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8309 \glxtrparen{\protect\glsfirstabbrvemfont{\Glsaccessshortpl{##1}}}%
8310 }%

```

The first use full form only displays the long form, but it typically won't be used as the regular attribute is set by this style.

```

8311 \renewcommand*{\glxtrfullformat}[2]{%
8312 \glsfirstlongemfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8313 \ifglxtrinsertinside\else##2\fi
8314 }%
8315 \renewcommand*{\glxtrfullplformat}[2]{%
8316 \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8317 \ifglxtrinsertinside\else##2\fi
8318 }%
8319 \renewcommand*{\Glsxtrfullformat}[2]{%
8320 \glsfirstlongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8321 \ifglxtrinsertinside\else##2\fi
8322 }%
8323 \renewcommand*{\Glsxtrfullplformat}[2]{%
8324 \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8325 \ifglxtrinsertinside\else##2\fi
8326 }%
8327 }

```

t-em-desc-noreg Like long-em-noshort-em-desc but doesn't set the regular attribute.

```
8328 \newabbreviationstyle{long-em-noshort-em-desc-noreg}%
8329 {%
8330   \GlsXtrUseAbbrStyleSetup{long-em-noshort-em-desc}%
      Unset the regular attribute if it has been set.
8331   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8332     \glshasattribute{\the\glslabeltok}{regular}%
8333     {%
8334       \glissetattribute{\the\glslabeltok}{regular}{false}%
8335     }%
8336   }%
8337 }%
8338 }%
8339 {%
8340   \GlsXtrUseAbbrStyleFmts{long-em-noshort-em-desc}%
8341 }
```

ort-em-footnote

```
8342 \newabbreviationstyle{short-em-footnote}%
8343 {%
8344   \renewcommand*{\CustomAbbreviationFields}{%
8345     name={\protect\glsabbrvemfont{\the\glsshorttok}},
8346     sort={\the\glsshorttok},
8347     description={\the\glslongtok},%
8348     first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
8349       \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
8350       {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
8351     firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
8352       \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
8353       {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
8354     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
      Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the regular attribute
      if it has been set.
8355   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8356     \glissetattribute{\the\glslabeltok}{nohyperfirst}{true}%
8357     \glshasattribute{\the\glslabeltok}{regular}%
8358     {%
8359       \glissetattribute{\the\glslabeltok}{regular}{false}%
8360     }%
8361   }%
8362 }%
8363 }%
8364 {%
8365   \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtremsuffix}%
8366   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
8367   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
8368   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
8369   \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form followed by the long form as a footnote.

```

8370 \renewcommand*{\glxtrfullformat}[2]{%
8371   \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
8372   \ifglxtrininsertinside\else##2\fi
8373   \protect\glxtrabbrvfootnote{##1}%
8374   {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
8375 }%
8376 \renewcommand*{\glxtrfullplformat}[2]{%
8377   \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
8378   \ifglxtrininsertinside\else##2\fi
8379   \protect\glxtrabbrvfootnote{##1}%
8380   {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
8381 }%
8382 \renewcommand*{\Glsxtrfullformat}[2]{%
8383   \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
8384   \ifglxtrininsertinside\else##2\fi
8385   \protect\glxtrabbrvfootnote{##1}%
8386   {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
8387 }%
8388 \renewcommand*{\Glsxtrfullplformat}[2]{%
8389   \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
8390   \ifglxtrininsertinside\else##2\fi
8391   \protect\glxtrabbrvfootnote{##1}%
8392   {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
8393 }%

```

The first use full form and the inline full form use the short (long) style.

```

8394 \renewcommand*{\glxtrinlinefullformat}[2]{%
8395   \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
8396   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
8397   \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
8398 }%
8399 \renewcommand*{\glxtrinlinefullplformat}[2]{%
8400   \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
8401   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
8402   \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
8403 }%
8404 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
8405   \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
8406   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
8407   \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
8408 }%
8409 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
8410   \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
8411   \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
8412   \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
8413 }%
8414 }

```

footnote-em Backward compatibility:

```
8415 \@glxtr@deprecated@abbrstyle{footnote-em}{short-em-footnote}
```

em-postfootnote

```
8416 \newabbreviationstyle{short-em-postfootnote}%
8417 {%
8418   \renewcommand*{\CustomAbbreviationFields}{%
8419     name={\protect\glsabbrvemfont{\the\glsshorttok}},
8420     sort={\the\glsshorttok},
8421     description={\the\glslongtok},%
8422     first={\protect\glsfirstabbrvemfont{\the\glsshorttok}},%
8423     firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}},%
8424     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
8425   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8426     \csdef{glxtrpostlink\glscategorylabel}{%
8427       \glxtrifwasfirstuse
8428       {%
8429         \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}}%
8430         {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}%
8431       }%
8432     }%
8433   }%
8434   \glshasattribute{\the\glslabeltok}{regular}%
8435   {%
8436     \glissetattribute{\the\glslabeltok}{regular}{false}%
8437   }%
8438   {}%
8439 }%
```

Make this category insert a footnote after the link if this was the first use, and unset the regular attribute if it has been set.

```
8425 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8426   \csdef{glxtrpostlink\glscategorylabel}{%
8427     \glxtrifwasfirstuse
8428     {%
8429       \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}}%
8430       {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}%
8431     }%
8432   }%
8433 }%
8434 \glshasattribute{\the\glslabeltok}{regular}%
8435 {%
8436   \glissetattribute{\the\glslabeltok}{regular}{false}%
8437 }%
8438 {}%
8439 }%
```

Needs the specific font command here as the style may have been lost by the time the footnote occurs.

```
8429   \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}}%
8430   {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}%
8431 }%
8432 }%
8433 }%
8434 \glshasattribute{\the\glslabeltok}{regular}%
8435 {%
8436   \glissetattribute{\the\glslabeltok}{regular}{false}%
8437 }%
8438 {}%
8439 }%
```

The footnote needs to be suppressed in the inline form, so \glxtrfull must set the first use switch off.

```
8440 \renewcommand*{\glxtrsetupfulldefs}{%
8441   \let\glxtrifwasfirstuse\@secondoftwo
8442 }%
8443 }%
8444 {%
8445   \renewcommand*{\abbrvpluralsuffix}{\protect\glxtremsuffix}%
8446   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
8447   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
8448   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
8449   \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form. The long form is deferred.

```
8450 \renewcommand*{\glxtrfullformat}[2]{%
```

```

8451 \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8452 \ifglxtrinsertinside\else##2\fi
8453 }%
8454 \renewcommand*{\glxtrfullplformat}[2]{%
8455 \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8456 \ifglxtrinsertinside\else##2\fi
8457 }%
8458 \renewcommand*{\Glsxtrfullformat}[2]{%
8459 \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8460 \ifglxtrinsertinside\else##2\fi
8461 }%
8462 \renewcommand*{\Glsxtrfullplformat}[2]{%
8463 \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8464 \ifglxtrinsertinside\else##2\fi
8465 }%

```

The first use full form and the inline full form use the short (long) style.

```

8466 \renewcommand*{\glxtrinlinfullformat}[2]{%
8467 \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8468 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8469 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
8470 }%
8471 \renewcommand*{\glxtrinlinfullplformat}[2]{%
8472 \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8473 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8474 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
8475 }%
8476 \renewcommand*{\Glsxtrinlinfullformat}[2]{%
8477 \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8478 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8479 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
8480 }%
8481 \renewcommand*{\Glsxtrinlinfullplformat}[2]{%
8482 \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8483 \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
8484 \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
8485 }%
8486 }

```

postfootnote-em Backward compatibility:

```

8487 \@glxtr@deprecated@abbrstyle{postfootnote-em}{short-em-postfootnote}

```

## 1.6.6 Predefined Styles (User Parentheses Hook)

These styles allow the user to adjust the parenthetical forms. These styles all test for the existence of the field given by:

glxtruserfield Default is the useri field.

```

8488 \newcommand*{\glxtruserfield}{useri}

```

`glsxtruserparen` The format of the parenthetical information. The first argument is the long/short form. The second argument is the entry's label. If `\glscurrentfieldvalue` has been defined, then we have at least glossaries v4.23, which makes it easier for the user to adjust this.

```

8489 \ifdef\glscurrentfieldvalue
8490 {
8491   \newcommand*{\glsxtruserparen}[2]{%
8492     \glsxtrfullsep{#2}%
8493     \glsxtrparen
8494     {#1\ifglshasfield{\glsxtruserfield}{#2}{, \glscurrentfieldvalue}{}}%
8495   }
8496 }
8497 {
8498   \newcommand*{\glsxtruserparen}[2]{%
8499     \glsxtrfullsep{#2}%
8500     \glsxtrparen
8501     {#1\ifglshasfield{\glsxtruserfield}{#2}{, \@glo@thisvalue}{}}%
8502   }
8503 }

```

Font used for short form:

`lsabbrvuserfont`

```

8504 \newcommand*{\glsabbrvuserfont}[1]{\glsabbrvdefaultfont{#1}}

```

Font used for short form on first use:

`stabbrvuserfont`

```

8505 \newcommand*{\glsfirstabbrvuserfont}[1]{\glsabbrvuserfont{#1}}

```

Font used for long form:

`glslonguserfont`

```

8506 \newcommand*{\glslonguserfont}[1]{\glslongdefaultfont{#1}}

```

Font used for long form on first use:

`rstlonguserfont`

```

8507 \newcommand*{\glsfirstlonguserfont}[1]{\glslonguserfont{#1}}

```

The default short form suffix:

`lsxtrusersuffix`

```

8508 \newcommand*{\glsxtrusersuffix}{\glsxtrabbrvpluralsuffix}

```

`long-short-user`

```

8509 \newabbreviationstyle{long-short-user}%
8510 {%
8511   \renewcommand*{\CustomAbbreviationFields}{%
8512     name={\protect\glsabbrvuserfont{\the\glsshorttok}},
8513     sort={\the\glsshorttok},

```

```

8514 first={\protect\glsfirstlonguserfont{\the\glslongtok}%
8515 \protect\glsextruserparen{\protect\glsfirstabbrvuserfont{\the\glsshorttok}}}%
8516 {\the\glslabeltok}},%
8517 firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}%
8518 \protect\glsextruserparen
8519 {\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%

8520 plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
8521 description={\protect\glslonguserfont{\the\glslongtok}}}%

```

Unset the regular attribute if it has been set.

```

8522 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8523 \glshasattribute{\the\glslabeltok}{regular}%
8524 {%
8525 \glssetattribute{\the\glslabeltok}{regular}{false}%
8526 }%
8527 {}%
8528 }%
8529 }%
8530 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

8531 \renewcommand*{\abbrvpluralsuffix}{\glsextrusersuffix}%
8532 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
8533 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
8534 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
8535 \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

8536 \renewcommand*{\glsextrfullformat}[2]{%
8537 \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsextrinsertinside##2\fi}%
8538 \ifglsextrinsertinside\else##2\fi
8539 \glsextruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
8540 }%
8541 \renewcommand*{\glsextrfullplformat}[2]{%
8542 \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsextrinsertinside##2\fi}%
8543 \ifglsextrinsertinside\else##2\fi
8544 \glsextruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
8545 }%
8546 \renewcommand*{\Glsxtrfullformat}[2]{%
8547 \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsextrinsertinside##2\fi}%
8548 \ifglsextrinsertinside\else##2\fi
8549 \glsextruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
8550 }%
8551 \renewcommand*{\Glsxtrfullplformat}[2]{%
8552 \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsextrinsertinside##2\fi}%
8553 \ifglsextrinsertinside\else##2\fi
8554 \glsextruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
8555 }%
8556 }

```

-postshort-user Like long-short-user but defers the parenthetical matter to after the link.

```
8557 \newabbreviationstyle{long-postshort-user}%
8558 {%
8559   \renewcommand*{\CustomAbbreviationFields}{%
8560     name={\protect\glsabbrvuserfont{\the\glsshorttok}},
8561     sort={\the\glsshorttok},
8562     first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
8563     firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
8564     plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
8565     description={\protect\glslonguserfont{\the\glslongtok}}}%
8566 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8567   \csdef{glsxtrpostlink\glscategorylabel}{%
8568     \glsxtrifwasfirstuse
8569     {%
8570       \glsxtruserparen
8571       {\glsfirstabbrvuserfont{\glsentryshort{\glslabel}}}%
8572       {\glslabel}%
8573     }%
8574   }%
8575 }%
8576 \glsattribute{\the\glslabeltok}{regular}%
8577 {%
8578   \glssetattribute{\the\glslabeltok}{regular}{false}%
8579 }%
8580 {}%
8581 }%
8582 }%
8583 {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
8584 \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
8585 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
8586 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
8587 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
8588 \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
```

First use full form:

```
8589 \renewcommand*{\glsxtrfullformat}[2]{%
8590   \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8591   \ifglsxtrininsertinside\else##2\fi
8592 }%
8593 \renewcommand*{\glsxtrfullplformat}[2]{%
8594   \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
8595   \ifglsxtrininsertinside\else##2\fi
8596 }%
8597 \renewcommand*{\Glsxtrfullformat}[2]{%
8598   \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
8599   \ifglsxtrininsertinside\else##2\fi
8600 }%
```



```

8601 \renewcommand*{\Glsxtrfullplformat}[2]{%
8602   \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8603   \ifglxtrinsertinside\else##2\fi
8604 }%

```

In-line format:

```

8605 \renewcommand*{\glxtrinlinefullformat}[2]{%
8606   \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8607   \ifglxtrinsertinside\else##2\fi
8608   \glxtruserparen{\glsfirstabbrvuserfont{\Glsaccessshort{##1}}}{##1}%
8609 }%
8610 \renewcommand*{\glxtrinlinefullplformat}[2]{%
8611   \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8612   \ifglxtrinsertinside\else##2\fi
8613   \glxtruserparen{\glsfirstabbrvuserfont{\Glsaccessshortpl{##1}}}{##1}%
8614 }%
8615 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
8616   \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
8617   \ifglxtrinsertinside\else##2\fi
8618   \glxtruserparen{\glsfirstabbrvuserfont{\Glsaccessshort{##1}}}{##1}%
8619 }%
8620 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
8621   \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
8622   \ifglxtrinsertinside\else##2\fi
8623   \glxtruserparen{\glsfirstabbrvuserfont{\Glsaccessshortpl{##1}}}{##1}%
8624 }%
8625 }

```

short-user-desc Like long-postshort-user but the user supplies the description.

```

8626 \newabbreviationstyle{long-postshort-user-desc}%
8627 {%
8628   \renewcommand*{\CustomAbbreviationFields}{%
8629     name={\protect\glslonguserfont{\the\glslongtok}%
8630       \protect\glxtruserparen
8631         {\protect\glsabbrvuserfont{\the\glsshorttok}}{\the\glslabeltok}},
8632     sort={\the\glslongtok},
8633     first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
8634     firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
8635     text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
8636     plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}}%
8637 }%
8638 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8639   \csdef{glxtrpostlink\glscategorylabel}{%
8640     \glxtrifwasfirstuse
8641     {%
8642       \glxtruserparen
8643         {\glsfirstabbrvuserfont{\glsentryshort{\glslabel}}}%
8644         {\glslabel}%
8645     }%

```

```

8646     {}%
8647   }%
8648   \glshasattribute{\the\glslabeltok}{regular}%
8649   {%
8650     \glissetattribute{\the\glslabeltok}{regular}{false}%
8651   }%
8652   {}%
8653 }%
8654 }%
8655 {%
8656   \GlsXtrUseAbbrStyleFmts{long-postshort-user}%
8657 }

```

t-postlong-user Like short-long-user but defers the parenthetical matter to after the link.

```

8658 \newabbreviationstyle{short-postlong-user}%
8659 {%
8660   \renewcommand*{\CustomAbbreviationFields}{%
8661     name={\protect\glsabbrvuserfont{\the\glsshorttok}},
8662     sort={\the\glsshorttok},
8663     first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
8664     firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
8665     plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
8666     description={\protect\glslonguserfont{\the\glslongtok}}}%
8667   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8668     \csdef{glsxtrpostlink\glscategorylabel}{%
8669       \glsxtrifwasfirstuse
8670       {%
8671         \glsxtruserparen
8672         {\glsfirstlonguserfont{\glsentrylong{\glslabel}}}%
8673         {\glslabel}%
8674       }%
8675     }%
8676   }%
8677   \glshasattribute{\the\glslabeltok}{regular}%
8678   {%
8679     \glissetattribute{\the\glslabeltok}{regular}{false}%
8680   }%
8681   {}%
8682 }%
8683 }%
8684 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

8685 \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
8686 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
8687 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
8688 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
8689 \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%

```

First use full form:

```

8690 \renewcommand*{\glxtrfullformat}[2]{%
8691   \glsfirstabbrvuserfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8692   \ifglxtrinsertinside\else##2\fi
8693 }%
8694 \renewcommand*{\glxtrfullplformat}[2]{%
8695   \glsfirstabbrvuserfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8696   \ifglxtrinsertinside\else##2\fi
8697 }%
8698 \renewcommand*{\Glsxtrfullformat}[2]{%
8699   \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8700   \ifglxtrinsertinside\else##2\fi
8701 }%
8702 \renewcommand*{\Glsxtrfullplformat}[2]{%
8703   \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8704   \ifglxtrinsertinside\else##2\fi
8705 }%

```

In-line format:

```

8706 \renewcommand*{\glxtrinlinefullformat}[2]{%
8707   \glsfirstabbrvuserfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8708   \ifglxtrinsertinside\else##2\fi
8709   \glxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
8710 }%
8711 \renewcommand*{\glxtrinlinefullplformat}[2]{%
8712   \glsfirstabbrvuserfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8713   \ifglxtrinsertinside\else##2\fi
8714   \glxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
8715 }%
8716 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
8717   \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8718   \ifglxtrinsertinside\else##2\fi
8719   \glxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
8720 }%
8721 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
8722   \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8723   \ifglxtrinsertinside\else##2\fi
8724   \glxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
8725 }%
8726 }

```

`long-user-desc` Like `short-postlong-user` but leaves the user to specify the description.

```

8727 \newabbreviationstyle{short-postlong-user-desc}%
8728 {%
8729   \renewcommand*{\CustomAbbreviationFields}{%
8730     name={\protect\glsabbrvuserfont{\the\glsshorttok}%
8731       \protect\glxtruserparen
8732         {\protect\glslonguserfont{\the\glslongpltok}}}%
8733     {\the\glslabeltok}},

```

```

8734     sort={\the\glsshorttok},
8735     first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
8736     firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%

8737     text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
8738     plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}}%
8739 }%
8740 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8741   \csdef{glsxtrpostlink\glscategorylabel}{%
8742     \glsxtrifwasfirstuse
8743     {%
8744       \glsxtruserparen
8745       {\glsfirstlonguserfont{\glsentrylong{\glslabel}}}%
8746       {\glslabel}}%
8747     }%
8748   }%
8749 }%
8750 \glsattribute{\the\glslabeltok}{regular}%
8751 {%
8752   \glssetattribute{\the\glslabeltok}{regular}{false}%
8753 }%
8754 {}%
8755 }%
8756 }%
8757 {%
8758   \GlsXtrUseAbbrStyleFmts{short-postlong-user}%
8759 }

```

short-user-desc

```

8760 \newabbreviationstyle{long-short-user-desc}%
8761 {%
8762   \renewcommand*{\CustomAbbreviationFields}{%
8763     name={\glsxtrlongshortdescname},
8764     sort={\glsxtrlongshortdescsort},%

8765     first={\protect\glsfirstlonguserfont{\the\glslongtok}}%
8766     \protect\glsxtruserparen{\protect\glsfirstabbrvuserfont{\the\glsshorttok}}}%
8767     {\the\glslabeltok}},%
8768     firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}}%
8769     \protect\glsxtruserparen
8770     {\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%
8771     text={\protect\glsabbrvfont{\the\glsshorttok}},%
8772     plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
8773   }%

```

Unset the regular attribute if it has been set.

```

8774   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8775     \glsattribute{\the\glslabeltok}{regular}%
8776     {%
8777       \glssetattribute{\the\glslabeltok}{regular}{false}%

```

```

8778 }%
8779 {}%
8780 }%
8781 }%
8782 {%
8783 \GlsXtrUseAbbrStyleFmts{long-short-user}%
8784 }

```

short-long-user

```

8785 \newabbreviationstyle{short-long-user}%
8786 {%
    \glslonguserfont is used in the description since \glsdesc doesn't set the style.
8787 \renewcommand*{\CustomAbbreviationFields}{%
8788     name={\protect\glsabbrvuserfont{\the\glsshorttok}},
8789     sort={\the\glsshorttok},
8790     description={\protect\glslonguserfont{\the\glslongtok}},%
8791     first={\protect\glsfirstabbrvuserfont{\the\glsshorttok}}%
8792     \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongtok}}}%
8793     {\the\glslabeltok}},%
8794     firstplural={\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}%
8795     \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongpltok}}}%
8796     {\the\glslabeltok}},%

8797     plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}}%

```

Unset the regular attribute if it has been set.

```

8798 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8799     \glshasattribute{\the\glslabeltok}{regular}%
8800     {%
8801         \glissetattribute{\the\glslabeltok}{regular}{false}%
8802     }%
8803     {}%
8804 }%
8805 }%
8806 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

8807 \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
8808 \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
8809 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
8810 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
8811 \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

8812 \renewcommand*{\glxtrfullformat}[2]{%
8813     \glsfirstabbrvuserfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
8814     \ifglxtrininsertinside\else##2\fi
8815     \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
8816 }%
8817 \renewcommand*{\glxtrfullplformat}[2]{%

```

```

8818 \glsfirstabbrvuserfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8819 \ifglxtrinsertinside\else##2\fi
8820 \glxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
8821 }%
8822 \renewcommand*{\Glsxtrfullformat}[2]{%
8823 \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
8824 \ifglxtrinsertinside\else##2\fi
8825 \glxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
8826 }%
8827 \renewcommand*{\Glsxtrfullplformat}[2]{%
8828 \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
8829 \ifglxtrinsertinside\else##2\fi
8830 \glxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
8831 }%
8832 }

```

-long-user-desc

```

8833 \newabbreviationstyle{short-long-user-desc}%
8834 {%
8835 \renewcommand*{\CustomAbbreviationFields}{%
8836 name={\glxtrshortlongdescname},
8837 sort={\glxtrshortlongdescsort},%
8838 first={\protect\glsfirstabbrvuserfont{\the\glsshorttok}}%
8839 \protect\glxtruserparen{\protect\glsfirstlonguserfont{\the\glslongtok}}}%
8840 {\the\glslabeltok}},%
8841 firstplural={\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}%
8842 \protect\glxtruserparen{\protect\glsfirstlonguserfont{\the\glslongpltok}}}%
8843 {\the\glslabeltok}},%
8844 text={\protect\glsabbrvfont{\the\glsshorttok}},%
8845 plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
8846 }%

```

Unset the regular attribute if it has been set.

```

8847 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8848 \glshasattribute{\the\glslabeltok}{regular}%
8849 {%
8850 \glissetattribute{\the\glslabeltok}{regular}{false}%
8851 }%
8852 {}%
8853 }%
8854 }%
8855 {%
8856 \GlsXtrUseAbbrStyleFmts{short-long-user}%
8857 }

```

### 1.6.7 Predefined Styles (Hyphen)

These styles are designed to work with the `markwords` attribute. They check if the inserted material (provided by the final optional argument of commands like `\gls`) starts with a hy-

phen. If it does, the insert is added to the parenthetical material. Note that commands like `\glxtrlong set \glsinsert` to empty with the entire link-text stored in `\glscustomtext`.

`trifhyphenstart` Checks if the argument starts with a hyphen. The argument may be `\glsinsert` so check for that and expand.

```
8858 \newrobustcmd*{\glxtrifhyphenstart}[3]{%
8859   \ifx\glsinsert#1\relax
8860   \expandafter\@glxtrifhyphenstart#1\relax\relax
8861   \@end@glxtrifhyphenstart{#2}{#3}%
8862   \else
8863   \@glxtrifhyphenstart#1\relax\relax\@end@glxtrifhyphenstart{#2}{#3}%
8864   \fi
8865 }
```

`trifhyphenstart`

```
8866 \def\@glxtrifhyphenstart#1#2\@end@glxtrifhyphenstart#3#4{%
8867   \ifx-#1\relax#3\else #4\fi
8868 }
```

`trlonghyphenshort`

```
\glxtrlonghyphenshort{<label>}{<long>}{<short>}{<insert>}
```

The `<long>` and `<short>` arguments may be the plural form. The `<long>` argument may also be the first letter uppercase form.

```
8869 \newcommand*{\glxtrlonghyphenshort}[4]{%
```

Grouping is needed to localise the redefinitions.

```
8870 {%
```

If `<insert>` starts with a hyphen, redefine `\glxtrwordsep` to a hyphen. The inserted material is also inserted into the parenthetical part. (The inserted material is grouped as a precautionary measure.) No change is made to `\glxtrwordsep` if `<insert>` doesn't start with a hyphen.

```
8871   \glxtrifhyphenstart{#4}{\def\glxtrwordsep{-}}{}%
8872   \glsfirstlonghyphenfont{#2\ifglxtrininsertinside{#4}\fi}%
8873   \ifglxtrininsertinside\else{#4}\fi
8874   \glxtrfullsep{#1}%
8875   \glxtrparen{\glsfirstabbrvhyphenfont{#3\ifglxtrininsertinside{#4}\fi}%
8876   \ifglxtrininsertinside\else{#4}\fi}%
8877 }%
8878 }
```

`abbrvhyphenfont`

```
8879 \newcommand*{\glsabbrvhyphenfont}{\glsabbrvdefaultfont}%
```

`abbrvhyphenfont`

```
8880 \newcommand*{\glsfirstabbrvhyphenfont}{\glsabbrvhyphenfont}%
```

slonghyphenfont

```
8881 \newcommand*{\glslonghyphenfont}{\glslongdefaultfont}%
```

tlonghyphenfont

```
8882 \newcommand*{\glsfirstlonghyphenfont}{\glslonghyphenfont}%
```

The default short form suffix:

xtrhyphensuffix

```
8883 \newcommand*{\glsxtrhyphensuffix}{\glsxtrabbrvpluralsuffix}
```

en-short-hyphen Designed for use with the markwords attribute.

```
8884 \newabbreviationstyle{long-hyphen-short-hyphen}%
8885 {%
8886   \renewcommand*{\CustomAbbreviationFields}{%
8887     name={\protect\glsabbrvhyphenfont{\the\glsshorttok}},
8888     sort={\the\glsshorttok},
8889     first={\protect\glsfirstlonghyphenfont{\the\glslongtok}}%
8890     \protect\glsxtrfullsep{\the\glslabeltok}%
8891     \glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}}},%
8892     firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}}%
8893     \protect\glsxtrfullsep{\the\glslabeltok}%
8894     \glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}}},%
8895     plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}},%
8896     description={\protect\glslonghyphenfont{\the\glslongtok}}}%
```

Unset the regular attribute if it has been set.

```
8897 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8898   \glsasattribute{\the\glslabeltok}{regular}%
8899   {%
8900     \glssetattribute{\the\glslabeltok}{regular}{false}%
8901     }%
8902   }%
8903 }%
8904 }%
8905 {%
8906   \renewcommand*{\abbrvpluralsuffix}{\glsxtrhyphensuffix}%
8907   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
8908   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
8909   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
8910   \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
8911 \renewcommand*{\glsxtrfullformat}[2]{%
8912   \glsxtrlonghyphenshort{##1}{\glsaccesslong{##1}}{\glsaccessshort{##1}}{##2}%
8913 }%
8914 \renewcommand*{\glsxtrfullplformat}[2]{%
8915   \glsxtrlonghyphenshort{##1}{\glsaccesslongpl{##1}}%
8916   {\glsaccessshortpl{##1}}{##2}%
8917 }%
```



```

8918 \renewcommand*{\Glsxtrfullformat}[2]{%
8919   \glxtrlonghyphenshort{##1}{\Glsaccesslong{##1}}{\Glsaccessshort{##1}}{##2}%
8920 }%
8921 \renewcommand*{\Glsxtrfullplformat}[2]{%
8922   \glxtrlonghyphenshort{##1}{\Glsaccesslongpl{##1}}%
8923   {\Glsaccessshortpl{##1}}{##2}%
8924 }%
8925 }

```

ort-hyphen-desc Like long-hyphen-short-hyphen but the description must be supplied by the user.

```

8926 \newabbreviationstyle{long-hyphen-short-hyphen-desc}%
8927 {%
8928   \renewcommand*{\CustomAbbreviationFields}{%
8929     name={\glxtrlongshortdescname},
8930     sort={\glxtrlongshortdescsort},
8931     first={\protect\glsfirstlonghyphenfont{\the\glslongtok}%
8932       \protect\glxtrfullsep{\the\glslabeltok}%
8933       \glxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}}},%
8934     firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}%
8935       \protect\glxtrfullsep{\the\glslabeltok}%
8936       \glxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}}},%
8937     text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
8938     plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
8939   }%

```

Unset the regular attribute if it has been set.

```

8940 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8941   \glshasattribute{\the\glslabeltok}{regular}%
8942   {%
8943     \glissetattribute{\the\glslabeltok}{regular}{false}%
8944   }%
8945   {}%
8946 }%
8947 }%
8948 {%
8949   \GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%
8950 }

```

onghyphennoshort `\glxtrlonghyphennoshort{<label>}{<long>}{<insert>}`

```

8951 \newcommand*{\glxtrlonghyphennoshort}[3]{%

```

Grouping is needed to localise the redefinitions.

```

8952 {%

```

If *<insert>* starts with a hyphen, redefine `\glxtrwordsep` to a hyphen. The inserted material is also inserted into the parenthetical part. (The inserted material is grouped as a precautionary measure.) No change is made to `\glxtrwordsep` if *<insert>* doesn't start with a hyphen.

```

8953 \glxtrifhyphenstart{#3}{\def\glxtrwordsep{-}}{}%
8954 \glsfirstlonghyphenfont{#2\ifglxtrininsertinside{#3}\fi}%
8955 \ifglxtrininsertinside\else{#3}\fi
8956 }%
8957 }

```

short-desc-noreg This version doesn't show the short form (except explicitly with \glxtrshort). Since \glxtrshort doesn't support the hyphen switch, the short form just uses the default short-form font command. This style won't work with the regular as the regular form isn't flexible enough.

```

8958 \newabbreviationstyle{long-hyphen-noshort-desc-noreg}%
8959 {%
8960 \renewcommand*{\CustomAbbreviationFields}{%
8961 name={\protect\protect\glslonghyphenfont{\the\glslongtok}},
8962 sort={\expandonce\glxtrorglong},
8963 first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
8964 firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
8965 plural={\protect\glslonghyphenfont{\the\glslongpltok}}%
8966 }%

```

Unset the regular attribute if it has been set.

```

8967 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
8968 \glshasattribute{\the\glslabeltok}{regular}%
8969 {%
8970 \glissetattribute{\the\glslabeltok}{regular}{false}%
8971 }%
8972 }%
8973 }%
8974 }%
8975 {%
8976 \GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%

```

In case the user wants to mix and match font styles, these are redefined here.

```

8977 \renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
8978 \renewcommand*{\glsabrvfont}[1]{\glsabrvdefaultfont{##1}}%
8979 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
8980 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
8981 \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

8982 \renewcommand*{\glxtrsubsequentfmt}[2]{%
8983 \glxtrlonghyphennoshort{##1}{\glssaccesslong{##1}}{##2}%
8984 }%
8985 \renewcommand*{\glxtrsubsequentplfmt}[2]{%
8986 \glxtrlonghyphennoshort{##1}{\glssaccesslongpl{##1}}{##2}%
8987 }%
8988 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
8989 \glxtrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
8990 }%
8991 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%

```

```

8992 \glstrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
8993 }%

```

The inline full form displays the long format followed by the short form in parentheses.

```

8994 \renewcommand*{\glxtrinlinefullformat}[2]{%
8995 \glstrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
8996 \glxtrfullsep{##1}%
8997 \glxtrparen{\protect\glsfirstabbrvfont{\Glsaccessshort{##1}}}%
8998 }%
8999 \renewcommand*{\glxtrinlinefullplformat}[2]{%
9000 \glstrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
9001 \glxtrfullsep{##1}%
9002 \glxtrparen{\protect\glsfirstabbrvfont{\Glsaccessshortpl{##1}}}%
9003 }%
9004 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
9005 \glstrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
9006 \glxtrfullsep{##1}%
9007 \glxtrparen{\protect\glsfirstabbrvfont{\Glsaccessshort{##1}}}%
9008 }%
9009 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
9010 \glstrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
9011 \glxtrfullsep{##1}%
9012 \glxtrparen{\protect\glsfirstabbrvfont{\Glsaccessshortpl{##1}}}%
9013 }%

```

The first use full form only displays the long form.

```

9014 \renewcommand*{\glxtrfullformat}[2]{%
9015 \glstrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
9016 }%
9017 \renewcommand*{\glxtrfullplformat}[2]{%
9018 \glstrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
9019 }%
9020 \renewcommand*{\Glsxtrfullformat}[2]{%
9021 \glstrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
9022 }%
9023 \renewcommand*{\Glsxtrfullplformat}[2]{%
9024 \glstrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
9025 }%
9026 }

```

n-noshort-noreg It doesn't really make a great deal of sense to have a long-only style that doesn't have a description (unless no glossary is required), but the best course of action here is to use the short form as the name and the long form as the description.

```

9027 \newabbreviationstyle{long-hyphen-noshort-noreg}%
9028 {%
9029 \renewcommand*{\CustomAbbreviationFields}{%
9030 name={\protect\glsabbrvfont{\the\glsshorttok}},
9031 sort={\the\glsshorttok},
9032 first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
9033 firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%

```

```

9034     text={\protect\glslonghyphenfont{\the\glslongtok}},%
9035     plural={\protect\glslonghyphenfont{\the\glslongpltok}},%
9036     description={\the\glslongtok}%
9037 }%

```

Unset the regular attribute if it has been set.

```

9038 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9039   \glshasattribute{\the\glslabeltok}{regular}%
9040   {%
9041     \glssetattribute{\the\glslabeltok}{regular}{false}%
9042   }%
9043   {}}%
9044 }%
9045 }%
9046 {%
9047   \GlsXtrUseAbbrStyleFmts{long-desc}%
9048 }

```

`\glsxtrlonghyphen` `\glsxtrlonghyphen{<long>}{<label>}{<insert>}`

Used by long-hyphen-postshort-hyphen. The `<insert>` is check to determine if it starts with a hyphen but isn't used here as it's moved to the post-link hook.

```

9049 \newcommand*{\glsxtrlonghyphen}[3]{%

```

Grouping is needed to localise the redefinitions.

```

9050   {%
9051     \glsxtrifhyphenstart{#3}{\def\glsxtrwordsep{-}}{}%
9052     \glsfirstlonghyphenfont{#1}%
9053   }%
9054 }

```

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

Used in the post-link hook for the long-hyphen-postshort-hyphen style. Much like `\glsxtrlonghyphenshort` but omits the `<long>` part. This always uses the singular short form.

```

9055 \newcommand*{\glsxtrposthyphenshort}[2]{%
9056   {%
9057     \glsxtrifhyphenstart{#2}{\def\glsxtrwordsep{-}}{}%
9058     \ifglsxtrininsertinside{\glsfirstlonghyphenfont{#2}}\else{#2}\fi
9059     \glsxtrfullsep{#1}%
9060     \glsxtrparen
9061     {\glsfirstabbrvhyphenfont{\glsentryshort{#1}}\ifglsxtrininsertinside{#2}\fi}%
9062     \ifglsxtrininsertinside\else{#2}\fi
9063   }%

```

```
9064 }%
9065 }
```

hyphensubsequent `\glxtrposthyphensubsequent{<label>}{<insert>}`

Format in the post-link hook for subsequent use. The label is ignored by default.

```
9066 \newcommand*{\glxtrposthyphensubsequent}[2]{%
9067   \glssabrvfont{\ifglxtrininsertinside {#2}\fi}%
9068   \ifglxtrininsertinside \else{#2}\fi
9069 }
```

ostshort-hyphen Like long-hyphen-short-hyphen but shifts the insert and parenthetical material to the post-link hook.

```
9070 \newabbreviationstyle{long-hyphen-postshort-hyphen}%
9071 {%
9072   \renewcommand*{\CustomAbbreviationFields}{%
9073     name={\protect\glssabrvhyphenfont{\the\glsshorttok}},
9074     sort={\the\glsshorttok},
9075     first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
9076     firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
9077     plural={\protect\glssabrvhyphenfont{\the\glsshortpltok}},%
9078     description={\protect\glslonghyphenfont{\the\glslongtok}}}%
9079   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9080     \csdef{glxtrpostlink\glscategorylabel}{%
9081       \glxtrifwasfirstuse
9082       {%
9083         \glxtrposthyphenshort{\glslabel}{\glsinsert}%
9084       }%
9085     }%
```

Put the insertion into the post-link:

```
9086       \glxtrposthyphensubsequent{\glslabel}{\glsinsert}%
9087     }%
9088   }%
9089   \glshasattribute{\the\glslabeltok}{regular}%
9090   {%
9091     \glissetattribute{\the\glslabeltok}{regular}{false}%
9092   }%
9093   {}%
9094 }%
9095 }%
9096 {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
9097 \renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
9098 \renewcommand*{\glssabrvfont}[1]{\glssabrvhyphenfont{##1}}%
9099 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
```

```

9100 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
9101 \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%

```

Subsequent use needs to omit the insertion:

```

9102 \renewcommand*{\glsxtrsubsequentfmt}[2]{%
9103   \glsabbrvfont{\glsaccessshort{##1}}}%
9104 }%
9105 \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
9106   \glsabbrvfont{\glsaccessshortpl{##1}}}%
9107 }%
9108 \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
9109   \glsabbrvfont{\Glsaccessshort{##1}}}%
9110 }%
9111 \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
9112   \glsabbrvfont{\Glsaccessshortpl{##1}}}%
9113 }%

```

First use full form:

```

9114 \renewcommand*{\glsxtrfullformat}[2]{%
9115   \glsxtrlonghyphen{\glsaccesslong{##1}}{##1}{##2}%
9116 }%
9117 \renewcommand*{\glsxtrfullplformat}[2]{%
9118   \glsxtrlonghyphen{\glsaccesslongpl{##1}}{##1}{##2}%
9119 }%
9120 \renewcommand*{\Glsxtrfullformat}[2]{%
9121   \glsxtrlonghyphen{\Glsaccesslong{##1}}{##1}{##2}%
9122 }%
9123 \renewcommand*{\Glsxtrfullplformat}[2]{%
9124   \glsxtrlonghyphen{\Glsaccesslongpl{##1}}{##1}{##2}%
9125 }%

```

In-line format.

```

9126 \renewcommand*{\glsxtrinlinefullformat}[2]{%
9127   \glsfirstlonghyphenfont{\glsaccesslong{##1}}%
9128   \ifglsxtrininsertinside{##2}\fi}%
9129 \ifglsxtrininsertinside \else{##2}\fi
9130 }%
9131 \renewcommand*{\glsxtrinlinefullplformat}[2]{%
9132   \glsfirstlonghyphenfont{\glsaccesslongpl{##1}}%
9133   \ifglsxtrininsertinside{##2}\fi}%
9134 \ifglsxtrininsertinside \else{##2}\fi
9135 }%
9136 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
9137   \glsfirstlonghyphenfont{\Glsaccesslong{##1}}%
9138   \ifglsxtrininsertinside{##2}\fi}%
9139 \ifglsxtrininsertinside \else{##2}\fi
9140 }%
9141 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
9142   \glsfirstlonghyphenfont{\Glsaccesslongpl{##1}}%
9143   \ifglsxtrininsertinside{##2}\fi}%
9144 \ifglsxtrininsertinside \else{##2}\fi

```

```

9145 }%
9146 }

```

`ort-hyphen-desc` Like `long-hyphen-postshort-hyphen` but the description must be supplied by the user.

```

9147 \newabbreviationstyle{long-hyphen-postshort-hyphen-desc}%
9148 {%
9149   \renewcommand*{\CustomAbbreviationFields}{%
9150     name={\glxtrlongshortdescname},
9151     sort={\glxtrlongshortdescsort},%
9152     first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
9153     firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
9154     text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
9155     plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}}%
9156 }%
9157 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9158   \csdef{glxtrpostlink\glscategorylabel}{%
9159     \glxtrifwasfirstuse
9160     {%
9161       \glxtrposthyphenshort{\glslabel}{\glsinsert}%
9162     }%
9163     {%

```

Put the insertion into the post-link:

```

9164       \glxtrposthyphensubsequent{\glslabel}{\glsinsert}%
9165     }%
9166   }%
9167   \glshasattribute{\the\glslabeltok}{regular}%
9168   {%
9169     \glissetattribute{\the\glslabeltok}{regular}{false}%
9170   }%
9171   {}%
9172 }%
9173 }%
9174 {%
9175   \GlsXtrUseAbbrStyleFmts{long-hyphen-postshort-hyphen}%
9176 }

```

`rshorthyphenlong`

```
\glxtrshorthyphenlong{<label>}{<short>}{<long>}{<insert>}
```

The `<long>` and `<short>` arguments may be the plural form. The `<long>` argument may also be the first letter uppercase form.

```
9177 \newcommand*{\glxtrshorthyphenlong}[4]{%
```

Grouping is needed to localise the redefinitions.

```
9178 {%
```

If *<insert>* starts with a hyphen, redefine `\glxtrwordsep` to a hyphen. The inserted material is also inserted into the parenthetical part. (The inserted material is grouped as a precautionary measure.)

```

9179 \glxtrifhyphenstart{#4}{\def\glxtrwordsep{-}}{}%
9180 \glsfirstabbrvhyphenfont{#2\ifglxtrininsertinside{#4}\fi}%
9181 \ifglxtrininsertinside\else{#4}\fi
9182 \glxtrfullsep{#1}%
9183 \glxtrparen{\glsfirstlonghyphenfont{#3\ifglxtrininsertinside{#4}\fi}%
9184 \ifglxtrininsertinside\else{#4}\fi}%
9185 }%
9186 }

```

hen-long-hyphen Designed for use with the `keywords` attribute.

```

9187 \newabbreviationstyle{short-hyphen-long-hyphen}%
9188 {%
9189 \renewcommand*{\CustomAbbreviationFields}{%
9190 name={\protect\glssabbrvhyphenfont{\the\glssshorttok}},
9191 sort={\the\glssshorttok},
9192 first={\protect\glsfirstabbrvhyphenfont{\the\glssshorttok}}%,
9193 \protect\glxtrfullsep{\the\glslabeltok}}%,
9194 \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%,
9195 firstplural={\protect\glsfirstabbrvhyphenfont{\the\glssshortpltok}}%,
9196 \protect\glxtrfullsep{\the\glslabeltok}}%,
9197 \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%,
9198 plural={\protect\glssabbrvhyphenfont{\the\glssshortpltok}}},%,
9199 description={\protect\glslonghyphenfont{\the\glslongtok}}}%

```

Unset the regular attribute if it has been set.

```

9200 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9201 \glshasattribute{\the\glslabeltok}{regular}}%
9202 {%
9203 \glissetattribute{\the\glslabeltok}{regular}{false}}%
9204 }%
9205 {}%
9206 }%
9207 }%
9208 {%
9209 \renewcommand*{\abbrvpluralsuffix}{\glxtrhyphensuffix}%
9210 \renewcommand*{\glssabbrvfont}[1]{\glssabbrvhyphenfont{##1}}%
9211 \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
9212 \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
9213 \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%

```

The first use full form and the inline full form are the same for this style.

```

9214 \renewcommand*{\glxtrfullformat}[2]{%
9215 \glxtrshorthyphenlong{##1}{\glssaccessshort{##1}}{\glssaccesslong{##1}}{##2}}%
9216 }%
9217 \renewcommand*{\glxtrfullplformat}[2]{%
9218 \glxtrshorthyphenlong{##1}%
9219 {\glssaccessshortpl{##1}}{\glssaccesslongpl{##1}}{##2}}%

```



```

9220 }%
9221 \renewcommand*{\Glsxtrfullformat}[2]{%
9222   \glxtrshorthyphenlong{##1}{\glsaccessshort{##1}}{\Glsaccesslong{##1}}{##2}%
9223 }%
9224 \renewcommand*{\Glsxtrfullplformat}[2]{%
9225   \glxtrshorthyphenlong{##1}%
9226   {\glsaccessshortpl{##1}}{\Glsaccesslongpl{##1}}{##2}%
9227 }%
9228 }

```

ong-hyphen-desc Like short-hyphen-long-hyphen but the description must be supplied by the user.

```

9229 \newabbreviationstyle{short-hyphen-long-hyphen-desc}%
9230 {%
9231   \renewcommand*{\CustomAbbreviationFields}{%
9232     name={\glxtrshortlongdescname},
9233     sort={\glxtrshortlongdescsort},
9234     first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
9235       \protect\glxtrfullsep{\the\glslabeltok}%
9236       \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
9237     firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
9238       \protect\glxtrfullsep{\the\glslabeltok}%
9239       \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%
9240     text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
9241     plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
9242   }%

```

Unset the regular attribute if it has been set.

```

9243 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9244   \glshasattribute{\the\glslabeltok}{regular}%
9245   {%
9246     \glissetattribute{\the\glslabeltok}{regular}{false}%
9247   }%
9248   {}%
9249 }%
9250 }%
9251 {%
9252   \GlsXtrUseAbbrStyleFmts{short-hyphen-long-hyphen}%
9253 }

```

lsxtrshorthyphen `\glxtrshorthyphen{<short>}{<label>}{<insert>}`

Used by short-hyphen-postlong-hyphen. The *<insert>* is check to determine if it starts with a hyphen but isn't used here as it's moved to the post-link hook.

```

9254 \newcommand*{\glxtrshorthyphen}[3]{%

```

Grouping is needed to localise the redefinitions.

```

9255 {%

```

```

9256 \glxtrifhyphenstart{#3}{\def\glxtrwordsep{-}}{}%
9257 \glsfirstabbrvhyphenfont{#1}%
9258 }%
9259 }

```

trposthyphenlong `\glxtrposthyphenlong{<label>}{<insert>}`

Used in the post-link hook for the short-hyphen-postlong-hyphen style. Much like `\glxtrshorthyphenlong` but omits the *<short>* part. This always uses the singular long form.

```

9260 \newcommand*{\glxtrposthyphenlong}[2]{%
9261   {%
9262     \glxtrifhyphenstart{#2}{\def\glxtrwordsep{-}}{}%
9263     \ifglxtrininsertinside{\glsfirstabbrvhyphenfont{#2}}\else{#2}\fi
9264     \glxtrfullsep{#1}%
9265     \glxtrparen
9266     {\glsfirstlonghyphenfont{\glsentrylong{#1}}\ifglxtrininsertinside{#2}\fi}%
9267     \ifglxtrininsertinside\else{#2}\fi
9268   }%
9269 }%
9270 }

```

postlong-hyphen Like short-hyphen-long-hyphen but shifts the insert and parenthetical material to the post-link hook.

```

9271 \newabbreviationstyle{short-hyphen-postlong-hyphen}%
9272 {%
9273   \renewcommand*{\CustomAbbreviationFields}{%
9274     name={\protect\glssabbrvhyphenfont{\the\glssshorttok}},
9275     sort={\the\glssshorttok},
9276     first={\protect\glsfirstabbrvhyphenfont{\the\glssshorttok}},%
9277     firstplural={\protect\glsfirstabbrvhyphenfont{\the\glssshortpltok}},%
9278     plural={\protect\glssabbrvhyphenfont{\the\glssshortpltok}},%
9279     description={\protect\glslonghyphenfont{\the\glslongtok}}}%
9280 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9281   \csdef{glxtrpostlink\glscategorylabel}{%
9282     \glxtrifwasfirstuse
9283     {%
9284       \glxtrposthyphenlong{\glslabel}{\glinsert}%
9285     }%
9286     {%

```

Put the insertion into the post-link:

```

9287       \glxtrposthyphensubsequent{\glslabel}{\glinsert}%
9288     }%
9289   }%
9290   \glshasattribute{\the\glslabeltok}{regular}%
9291   {%
9292     \glissetattribute{\the\glslabeltok}{regular}{false}%

```

```

9293     }%
9294     {}%
9295 }%
9296 }%
9297 {%

```

In case the user wants to mix and match font styles, these are redefined here.

```

9298 \renewcommand*\abbrvpluralsuffix{\glxtrabbrvpluralsuffix}%
9299 \renewcommand*\glsabbrvfont[1]{\glsabbrvhyphenfont{##1}}%
9300 \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvhyphenfont{##1}}%
9301 \renewcommand*\glsfirstlongfont[1]{\glsfirstlonghyphenfont{##1}}%
9302 \renewcommand*\glslongfont[1]{\glslonghyphenfont{##1}}%

```

Subsequent use needs to omit the insertion:

```

9303 \renewcommand*\glxtrsubsequentfmt[2]{%
9304   \glsabbrvfont{\glsaccessshort{##1}}%
9305 }%
9306 \renewcommand*\glxtrsubsequentplfmt[2]{%
9307   \glsabbrvfont{\glsaccessshortpl{##1}}%
9308 }%
9309 \renewcommand*\Glsxtrsubsequentfmt[2]{%
9310   \glsabbrvfont{\Glsaccessshort{##1}}%
9311 }%
9312 \renewcommand*\Glsxtrsubsequentplfmt[2]{%
9313   \glsabbrvfont{\Glsaccessshortpl{##1}}%
9314 }%

```

First use full form:

```

9315 \renewcommand*\glxtrfullformat[2]{%
9316   \glxtrshorthyphen{\glsaccessshort{##1}}{##1}{##2}%
9317 }%
9318 \renewcommand*\glxtrfullplformat[2]{%
9319   \glxtrshorthyphen{\glsaccessshortpl{##1}}{##1}{##2}%
9320 }%
9321 \renewcommand*\Glsxtrfullformat[2]{%
9322   \glxtrshorthyphen{\Glsaccessshort{##1}}{##1}{##2}%
9323 }%
9324 \renewcommand*\Glsxtrfullplformat[2]{%
9325   \glxtrshorthyphen{\Glsaccessshortpl{##1}}{##1}{##2}%
9326 }%

```

In-line format. Commands like `\glxtrfull` set `\glsinsert` to empty. The entire link-text (provided by the following commands) is stored in `\glscustomtext`.

```

9327 \renewcommand*\glxtrinlinefullformat[2]{%
9328   \glsfirstabbrvhyphenfont{\glsaccessshort{##1}}%
9329   \ifglxtrininsertinside{##2}\fi}%
9330   \ifglxtrininsertinside \else{##2}\fi
9331 }%
9332 \renewcommand*\glxtrinlinefullplformat[2]{%
9333   \glsfirstabbrvhyphenfont{\glsaccessshortpl{##1}}%
9334   \ifglxtrininsertinside{##2}\fi}%

```

```

9335 \ifglxtrinsertinside \else{##2}\fi
9336 }%
9337 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
9338 \glsfirstabbrvhyphenfont{\Glsaccessshort{##1}%
9339 \ifglxtrinsertinside{##2}\fi}%
9340 \ifglxtrinsertinside \else{##2}\fi
9341 }%
9342 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
9343 \glsfirstabbrvhyphenfont{\Glsaccessshortpl{##1}%
9344 \ifglxtrinsertinside{##2}\fi}%
9345 \ifglxtrinsertinside \else{##2}\fi
9346 }%
9347 }

```

ong-hyphen-desc Like short-hyphen-postlong-hyphen but the description must be supplied by the user.

```

9348 \newabbreviationstyle{short-hyphen-postlong-hyphen-desc}%
9349 {%
9350 \renewcommand*{\CustomAbbreviationFields}{%
9351 name={\glxtrshortlongdescname},
9352 sort={\glxtrshortlongdescsort},%
9353 first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}},%
9354 firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}},%
9355 text={\protect\glssabbrvhyphenfont{\the\glsshorttok}},%
9356 plural={\protect\glssabbrvhyphenfont{\the\glsshortpltok}}}%
9357 }%
9358 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9359 \csdef{glxtrpostlink\glscategorylabel}{%
9360 \glxtrifwasfirstuse
9361 {%
9362 \glxtrposthyphenlong{\glslabel}{\glssinsert}%
9363 }%
9364 {%

```

Put the insertion into the post-link:

```

9365 \glxtrposthyphensubsequent{\glslabel}{\glssinsert}%
9366 }%
9367 }%
9368 \glshasattribute{\the\glslabeltok}{regular}%
9369 {%
9370 \glsssetAttribute{\the\glslabeltok}{regular}{false}%
9371 }%
9372 {}%
9373 }%
9374 }%
9375 {%
9376 \GlsXtrUseAbbrStyleFmts{short-hyphen-postlong-hyphen}%
9377 }

```

## 1.6.8 Predefined Styles (No Short on First Use)

These styles show only the long form on first use and only the short form on subsequent use.

lsabbrvonlyfont

```
9378 \newcommand*{\glsabbrvonlyfont}{\glsabbrvdefaultfont}%
```

stabbrvonlyfont

```
9379 \newcommand*{\glsfirstabbrvonlyfont}{\glsabbrvonlyfont}%
```

glslongonlyfont

```
9380 \newcommand*{\glslongonlyfont}{\glslongdefaultfont}%
```

rstlongonlyfont

```
9381 \newcommand*{\glsfirstlongonlyfont}{\glslongonlyfont}%
```

The default short form suffix:

lsxtronlysuffix

```
9382 \newcommand*{\glsxtronlysuffix}{\glsxtrabbrvpluralsuffix}
```

only-short-only

```
9383 \newabbreviationstyle{long-only-short-only}%
9384 {%
9385   \renewcommand*{\CustomAbbreviationFields}{%
9386     name={\protect\glsabbrvonlyfont{\the\glsshorttok}},
9387     sort={\the\glsshorttok},
9388     first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
9389     firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
9390     plural={\protect\glsabbrvonlyfont{\the\glsshortpltok}},%
9391     description={\protect\glslongonlyfont{\the\glslongtok}}}%
9392 }
```

Unset the regular attribute if it has been set.

```
9392 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9393   \glshasattribute{\the\glslabeltok}{regular}%
9394   {%
9395     \glssetattribute{\the\glslabeltok}{regular}{false}%
9396   }%
9397   {}%
9398 }%
9399 }%
9400 {%
9401   \renewcommand*{\abbrvpluralsuffix}{\protect\glsxtronlysuffix}%
9402   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvonlyfont{##1}}%
9403   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvonlyfont{##1}}%
9404   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
9405   \renewcommand*{\glslongfont}[1]{\glslongonlyfont{##1}}%
```

The first use full form doesn't show the short form.

```

9406 \renewcommand*{\glxtrfullformat}[2]{%
9407   \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
9408   \ifglxtrinsertinside\else##2\fi
9409 }%
9410 \renewcommand*{\glxtrfullplformat}[2]{%
9411   \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
9412   \ifglxtrinsertinside\else##2\fi
9413 }%
9414 \renewcommand*{\Glsxtrfullformat}[2]{%
9415   \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
9416   \ifglxtrinsertinside\else##2\fi
9417 }%
9418 \renewcommand*{\Glsxtrfullplformat}[2]{%
9419   \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
9420   \ifglxtrinsertinside\else##2\fi
9421 }%

```

The inline full form does show the short form.

```

9422 \renewcommand*{\glxtrinlinefullformat}[2]{%
9423   \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
9424   \ifglxtrinsertinside\else##2\fi
9425   \glxtrfullsep{##1}%
9426   \glxtrparen{\protect\glsfirstabbrvonlyfont{\glsaccessshort{##1}}}%
9427 }%
9428 \renewcommand*{\glxtrinlinefullplformat}[2]{%
9429   \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
9430   \ifglxtrinsertinside\else##2\fi
9431   \glxtrfullsep{##1}%
9432   \glxtrparen{\protect\glsfirstabbrvonlyfont{\glsaccessshortpl{##1}}}%
9433 }%
9434 \renewcommand*{\Glsxtrinlinefullformat}[2]{%
9435   \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
9436   \ifglxtrinsertinside\else##2\fi
9437   \glxtrfullsep{##1}%
9438   \glxtrparen{\protect\glsfirstabbrvonlyfont{\glsaccessshortpl{##1}}}%
9439 }%
9440 \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
9441   \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
9442   \ifglxtrinsertinside\else##2\fi
9443   \glxtrfullsep{##1}%
9444   \glxtrparen{\protect\glsfirstabbrvonlyfont{\Glsaccessshortpl{##1}}}%
9445 }%
9446 }

```

xtronlydescsort

```

9447 \newcommand*{\glxtronlydescsort}{\the\glslongtok}

```

xtronlydescname

```

9448 \newcommand*{\glxtronlydescname}{%
9449   \protect\glslongfont{\the\glslongtok}}%
9450 }

```

short-only-desc

```

9451 \newabbreviationstyle{long-only-short-only-desc}%
9452 {%
9453   \renewcommand*{\CustomAbbreviationFields}{%
9454     name={\glxtronlydescname},
9455     sort={\glxtronlydescsort},%
9456     first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
9457     firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
9458     text={\protect\glsabbrvonlyfont{\the\glsshorttok}},%
9459     plural={\protect\glsabbrvonlyfont{\the\glsshortpltok}}}%
9460 }%

```

Unset the regular attribute if it has been set.

```

9461 \renewcommand*{\GlsXtrPostNewAbbreviation}{%
9462   \glshasattribute{\the\glslabeltok}{regular}%
9463   {%
9464     \glissetattribute{\the\glslabeltok}{regular}{false}%
9465   }%
9466   {}%
9467 }%
9468 }%
9469 {%
9470   \GlsXtrUseAbbrStyleFmts{long-only-short-only}%
9471 }

```

## 1.7 Using Entries in Headings

There are four main problems with using entries in sectioning commands: they can mess with the first use flag if they end up in the table of contents, they can add unwanted numbers to the entry's location list, the label is corrupted if used inside `\MakeUppercase` (which is used by the default headings style) and they need to be expandable for PDF bookmarks. The glossaries package therefore recommends the use of the expandable commands, such as `\glsentryshort`, instead but this doesn't reflect the formatting since it doesn't include `\glsabbrvfont`. The commands below are an attempt to get around these problems.

The PDF bookmark issue can easily be fixed with `hyperref`'s `\texorpdfstring` which can simply use the expandable command in the PDF string case. The  $\TeX$  string case can now use `\glxtrshort` with the `noindex` key set, which prevents the unwanted additions to the location list, and the `hyper` key set to false, which prevents the problem of nested links. This just leaves one thing left that needs to be dealt with, and that's what to do if the heading style uses `\MakeUppercase`.

Note that glossaries automatically loads `textcase`, so the label can be protected from case change with `textcase`'s `\NoCaseChange`. This means that we don't have a problem pro-

vided the page style uses `\MakeTextUppercase`, but the default heading page style uses `\MakeUppercase`.

To get around this, save the original definition of `\markboth` and `\markright` and adjust it so that `\MakeUppercase` is temporarily redefined to `\MakeTextUppercase`. Some packages or classes redefine these commands, so we can't just assume they still have the original kernel definition.

`\markright` Save original definition:

```
9472 \let\@glxstr@org@markright\markright
```

Redefine (grouping not added in case it interferes with the original code):

```
9473 \renewcommand*{\markright}[1]{%
9474   \glxstrmarkhook
9475   \@glxstr@org@markright{\@glxstrinmark#1\@glxstrnotinmark}%
9476   \glxstrrestoremarkhook
9477 }
```

`\markboth` Save original definition:

```
9478 \let\@glxstr@org@markboth\markboth
```

Redefine (grouping not added in case it interferes with the original code):

```
9479 \renewcommand*{\markboth}[2]{%
9480   \glxstrmarkhook
9481   \@glxstr@org@markboth
9482   {\@glxstrinmark#1\@glxstrnotinmark}%
9483   {\@glxstrinmark#2\@glxstrnotinmark}%
9484   \glxstrrestoremarkhook
9485 }
```

If this causes a problem provide a simple way of switching back to the original definitions:

`sxtrRevertMarks`

```
9486 \newcommand*{\glxstrRevertMarks}{%
9487   \let\markright\@glxstr@org@markright
9488   \let\markboth\@glxstr@org@markboth
9489 }
```

`\glxstrifinmark`

```
9490 \newcommand*{\glxstrifinmark}[2]{#2}
```

`\@glxstrinmark`

```
9491 \newrobustcmd*{\@glxstrinmark}{%
9492   \let\glxstrifinmark\@firstoftwo
9493 }
```

`glxstrnotinmark`

```
9494 \newrobustcmd*{\@glxstrnotinmark}{%
9495   \let\glxstrifinmark\@secondoftwo
9496 }
```



\glxstrmarkhook Hook used in new definition of \markboth and \markright to make some changes to apply to the marks:

```
9497 \newcommand*{\glxstrmarkhook}{%
```

Save current definitions:

```
9498 \let\@glxstr@org@MakeUppercase\MakeUppercase
9499 \let\@glxstr@org@glxstrtitleshort\glxstrtitleshort
9500 \let\@glxstr@org@glxstrtitleshortpl\glxstrtitleshortpl
9501 \let\@glxstr@org@Glsxtrtitleshort\Glsxtrtitleshort
9502 \let\@glxstr@org@Glsxtrtitleshortpl\Glsxtrtitleshortpl
9503 \let\@glxstr@org@glxstrtitletext\glxstrtitletext
9504 \let\@glxstr@org@Glsxtrtitletext\Glsxtrtitletext
9505 \let\@glxstr@org@glxstrtitleplural\glxstrtitleplural
9506 \let\@glxstr@org@Glsxtrtitleplural\Glsxtrtitleplural
9507 \let\@glxstr@org@glxstrtitlefirst\glxstrtitlefirst
9508 \let\@glxstr@org@Glsxtrtitlefirst\Glsxtrtitlefirst
9509 \let\@glxstr@org@glxstrtitlefirstplural\glxstrtitlefirstplural
9510 \let\@glxstr@org@Glsxtrtitlefirstplural\Glsxtrtitlefirstplural
9511 \let\@glxstr@org@glxstrtitlelong\glxstrtitlelong
9512 \let\@glxstr@org@glxstrtitlelongpl\glxstrtitlelongpl
9513 \let\@glxstr@org@Glsxtrtitlelong\Glsxtrtitlelong
9514 \let\@glxstr@org@Glsxtrtitlelongpl\Glsxtrtitlelongpl
9515 \let\@glxstr@org@glxstrtitlefull\glxstrtitlefull
9516 \let\@glxstr@org@glxstrtitlefullpl\glxstrtitlefullpl
9517 \let\@glxstr@org@Glsxtrtitlefull\Glsxtrtitlefull
9518 \let\@glxstr@org@Glsxtrtitlefullpl\Glsxtrtitlefullpl
```

New definitions

```
9519 \let\glxstrifinmark\@firstoftwo
9520 \let\MakeUppercase\MakeTextUppercase
9521 \let\glxstrtitleshort\glxstrheadshort
9522 \let\glxstrtitleshortpl\glxstrheadshortpl
9523 \let\Glsxtrtitleshort\Glsxtrheadshort
9524 \let\Glsxtrtitleshortpl\Glsxtrheadshortpl
9525 \let\glxstrtitletext\glxstrheadtext
9526 \let\Glsxtrtitletext\Glsxtrheadtext
9527 \let\glxstrtitleplural\glxstrheadplural
9528 \let\Glsxtrtitleplural\Glsxtrheadplural
9529 \let\glxstrtitlefirst\glxstrheadfirst
9530 \let\Glsxtrtitlefirst\Glsxtrheadfirst
9531 \let\glxstrtitlefirstplural\glxstrheadfirstplural
9532 \let\Glsxtrtitlefirstplural\Glsxtrheadfirstplural
9533 \let\glxstrtitlelong\glxstrheadlong
9534 \let\glxstrtitlelongpl\glxstrheadlongpl
9535 \let\Glsxtrtitlelong\Glsxtrheadlong
9536 \let\Glsxtrtitlelongpl\Glsxtrheadlongpl
9537 \let\glxstrtitlefull\glxstrheadfull
9538 \let\glxstrtitlefullpl\glxstrheadfullpl
9539 \let\Glsxtrtitlefull\Glsxtrheadfull
9540 \let\Glsxtrtitlefullpl\Glsxtrheadfullpl
```

9541 }

`\restoremarkhook` Hook used in new definition of `\markboth` and `\markright` to restore the modified definitions. (This is in case the original `\markboth` and `\markright` shouldn't be grouped for some reason. There already is some grouping within those original definitions, but some of the code lies outside that grouping, and possibly there's a reason for it.)

```
9542 \newcommand*{\glxtrrestoremarkhook}{%
9543   \let\glxtrifinmark\@secondoftwo
9544   \let\MakeUppercase\@glxtr@org@MakeUppercase
9545   \let\glxtrtitleshort\@glxtr@org@glxtrtitleshort
9546   \let\glxtrtitleshortpl\@glxtr@org@glxtrtitleshortpl
9547   \let\Glsxtrtitleshort\@glxtr@org@Glsxtrtitleshort
9548   \let\Glsxtrtitleshortpl\@glxtr@org@Glsxtrtitleshortpl
9549   \let\glxtrtitletext\@glxtr@org@glxtrtitletext
9550   \let\Glsxtrtitletext\@glxtr@org@Glsxtrtitletext
9551   \let\glxtrtitleplural\@glxtr@org@glxtrtitleplural
9552   \let\Glsxtrtitleplural\@glxtr@org@Glsxtrtitleplural
9553   \let\glxtrtitlefirst\@glxtr@org@glxtrtitlefirst
9554   \let\Glsxtrtitlefirst\@glxtr@org@Glsxtrtitlefirst
9555   \let\glxtrtitlefirstplural\@glxtr@org@glxtrtitlefirstplural
9556   \let\Glsxtrtitlefirstplural\@glxtr@org@Glsxtrtitlefirstplural
9557   \let\glxtrtitlelong\@glxtr@org@glxtrtitlelong
9558   \let\glxtrtitlelongpl\@glxtr@org@glxtrtitlelongpl
9559   \let\Glsxtrtitlelong\@glxtr@org@Glsxtrtitlelong
9560   \let\Glsxtrtitlelongpl\@glxtr@org@Glsxtrtitlelongpl
9561   \let\glxtrtitlefull\@glxtr@org@glxtrtitlefull
9562   \let\glxtrtitlefullpl\@glxtr@org@glxtrtitlefullpl
9563   \let\Glsxtrtitlefull\@glxtr@org@Glsxtrtitlefull
9564   \let\Glsxtrtitlefullpl\@glxtr@org@Glsxtrtitlefullpl
9565 }
```

Instead of using one document-wide conditional, use `headuc` attribute to determine whether or not to use the all upper case form.

`\glxtrheadshort` Command used to display short form in the page header.

```
9566 \newcommand*{\glxtrheadshort}[1]{%
9567   \protect\NoCaseChange
9568   {%
9569     \gl@ifattribute{#1}{headuc}{true}%
9570     {%
9571       \Glsxtrshort[noindex,hyper=false]{#1}[]%
9572     }%
9573     {%
9574       \glxtrshort[noindex,hyper=false]{#1}[]%
9575     }%
9576   }%
9577 }
```

`\lsxtrtitleshort` Command to display short form of abbreviation in section title and table of contents.

```

9578 \newrobustcmd*{\glxstrtitleshort}[1]{%
9579   \glxstrshort[noindex,hyper=false]{#1}[]%
9580 }

```

**sxtrheadshortpl** Command used to display plural short form in the page header. If you want the text converted to upper case, this needs to be redefined to use \GLSxtrshortpl instead. If you are using a smallcaps style, the default fonts don't provide italic smallcaps.

```

9581 \newcommand*{\glxtrheadshortpl}[1]{%
9582   \protect\NoCaseChange
9583   {%
9584     \gl@ifattribute{#1}{headuc}{true}%
9585     {%
9586       \GLSxtrshortpl[noindex,hyper=false]{#1}[]%
9587     }%
9588     {%
9589       \glxstrshortpl[noindex,hyper=false]{#1}[]%
9590     }%
9591   }%
9592 }

```

**xtrtitleshortpl** Command to display plural short form of abbreviation in section title and table of contents.

```

9593 \newrobustcmd*{\glxtrtitleshortpl}[1]{%
9594   \glxstrshortpl[noindex,hyper=false]{#1}[]%
9595 }

```

**GLsxtrheadshort** Command used to display short form in the page header with the first letter converted to upper case.

```

9596 \newcommand*{\GLsxtrheadshort}[1]{%
9597   \protect\NoCaseChange
9598   {%
9599     \gl@ifattribute{#1}{headuc}{true}%
9600     {%
9601       \GLSxtrshort[noindex,hyper=false]{#1}[]%
9602     }%
9603     {%
9604       \Glsxtrshort[noindex,hyper=false]{#1}[]%
9605     }%
9606   }%
9607 }

```

**lxsxtrtitleshort** Command to display short form of abbreviation in section title and table of contents with the first letter converted to upper case.

```

9608 \newrobustcmd*{\Glsxtrtitleshort}[1]{%
9609   \Glsxtrshort[noindex,hyper=false]{#1}[]%
9610 }

```

**sxtrheadshortpl** Command used to display plural short form in the page header with the first letter converted to upper case.

```

9611 \newcommand*{\Glsxtrheadshortpl}[1]{%
9612   \protect\NoCaseChange
9613   {%
9614     \glsifattribute{#1}{headuc}{true}%
9615     {%
9616       \Glsxtrshortpl[noindex,hyper=false]{#1}[]%
9617     }%
9618     {%
9619       \Glsxtrshortpl[noindex,hyper=false]{#1}[]%
9620     }%
9621   }%
9622 }

```

**xtrtitleshortpl** Command to display plural short form of abbreviation in section title and table of contents with the first letter converted to upper case.

```

9623 \newrobustcmd*{\Glsxtrtitleshortpl}[1]{%
9624   \Glsxtrshortpl[noindex,hyper=false]{#1}[]%
9625 }

```

**\glsxtrheadtext** As above but for the text value.

```

9626 \newcommand*{\glsxtrheadtext}[1]{%
9627   \protect\NoCaseChange
9628   {%
9629     \glsifattribute{#1}{headuc}{true}%
9630     {%
9631       \GLStext[noindex,hyper=false]{#1}[]%
9632     }%
9633     {%
9634       \glstext[noindex,hyper=false]{#1}[]%
9635     }%
9636   }%
9637 }

```

**glsxtrtitletext** Command to display text value in section title and table of contents.

```

9638 \newrobustcmd*{\glsxtrtitletext}[1]{%
9639   \glstext[noindex,hyper=false]{#1}[]%
9640 }

```

**\Glsxtrheadtext** First letter converted to upper case

```

9641 \newcommand*{\Glsxtrheadtext}[1]{%
9642   \protect\NoCaseChange
9643   {%
9644     \glsifattribute{#1}{headuc}{true}%
9645     {%
9646       \GLStext[noindex,hyper=false]{#1}[]%
9647     }%
9648     {%
9649       \Glstext[noindex,hyper=false]{#1}[]%
9650     }%

```

```

9651 }%
9652 }

```

**Glsxtrtitletext** Command to display text value in section title and table of contents with the first letter changed to upper case.

```

9653 \newrobustcmd*{\Glsxtrtitletext}[1]{%
9654   \Gls{text[noindex,hyper=false]}{#1}[]%
9655 }

```

**lsxtrheadplural** As above but for the plural value.

```

9656 \newcommand*{\Glsxtrheadplural}[1]{%
9657   \protect\NoCaseChange
9658   {%
9659     \glsifattribute{#1}{headuc}{true}%
9660     {%
9661       \GLSplural[noindex,hyper=false]{#1}[]%
9662     }%
9663     {%
9664       \glsplural[noindex,hyper=false]{#1}[]%
9665     }%
9666   }%
9667 }

```

**sxtrtitleplural** Command to display plural value in section title and table of contents.

```

9668 \newrobustcmd*{\Glsxtrtitleplural}[1]{%
9669   \glsplural[noindex,hyper=false]{#1}[]%
9670 }

```

**lsxtrheadplural** Convert first letter to upper case.

```

9671 \newcommand*{\Glsxtrheadplural}[1]{%
9672   \protect\NoCaseChange
9673   {%
9674     \glsifattribute{#1}{headuc}{true}%
9675     {%
9676       \GLSplural[noindex,hyper=false]{#1}[]%
9677     }%
9678     {%
9679       \Glsplural[noindex,hyper=false]{#1}[]%
9680     }%
9681   }%
9682 }

```

**sxtrtitleplural** Command to display plural value in section title and table of contents with the first letter changed to upper case.

```

9683 \newrobustcmd*{\Glsxtrtitleplural}[1]{%
9684   \Glsplural[noindex,hyper=false]{#1}[]%
9685 }

```

glsxtrheadfirst As above but for the first value.

```
9686 \newcommand*{\glsxtrheadfirst}[1]{%
9687   \protect\NoCaseChange
9688   {%
9689     \glsifattribute{#1}{headuc}{true}%
9690     {%
9691       \GLSfirst[noindex,hyper=false]{#1}[]%
9692     }%
9693     {%
9694       \glsfirst[noindex,hyper=false]{#1}[]%
9695     }%
9696   }%
9697 }
```

lsxtrtitlefirst Command to display first value in section title and table of contents.

```
9698 \newrobustcmd*{\glsxtrtitlefirst}[1]{%
9699   \glsfirst[noindex,hyper=false]{#1}[]%
9700 }
```

Glsxtrheadfirst First letter converted to upper case

```
9701 \newcommand*{\Glsxtrheadfirst}[1]{%
9702   \protect\NoCaseChange
9703   {%
9704     \glsifattribute{#1}{headuc}{true}%
9705     {%
9706       \GLSfirst[noindex,hyper=false]{#1}[]%
9707     }%
9708     {%
9709       \Glsfirst[noindex,hyper=false]{#1}[]%
9710     }%
9711   }%
9712 }
```

lsxtrtitlefirst Command to display first value in section title and table of contents with the first letter changed to upper case.

```
9713 \newrobustcmd*{\Glsxtrtitlefirst}[1]{%
9714   \Glsfirst[noindex,hyper=false]{#1}[]%
9715 }
```

headfirstplural As above but for the firstplural value.

```
9716 \newcommand*{\glsxtrheadfirstplural}[1]{%
9717   \protect\NoCaseChange
9718   {%
9719     \glsifattribute{#1}{headuc}{true}%
9720     {%
9721       \GLSfirstplural[noindex,hyper=false]{#1}[]%
9722     }%
9723     {%
```

```

9724     \glsfirstplural[noindex,hyper=false]{#1}[]%
9725 }%
9726 }%
9727 }

```

**titlefirstplural** Command to display firstplural value in section title and table of contents.

```

9728 \newrobustcmd*{\glsxtrtitlefirstplural}[1]{%
9729   \glsfirstplural[noindex,hyper=false]{#1}[]%
9730 }

```

**headfirstplural** First letter converted to upper case

```

9731 \newcommand*{\Glsxtrheadfirstplural}[1]{%
9732   \protect\NoCaseChange
9733   {%
9734     \glsifattribute{#1}{headuc}{true}%
9735     {%
9736       \GLSfirstplural[noindex,hyper=false]{#1}[]%
9737     }%
9738     {%
9739       \Glsfirstplural[noindex,hyper=false]{#1}[]%
9740     }%
9741   }%
9742 }

```

**titlefirstplural** Command to display first value in section title and table of contents with the first letter changed to upper case.

```

9743 \newrobustcmd*{\Glsxtrtitlefirstplural}[1]{%
9744   \Glsfirstplural[noindex,hyper=false]{#1}[]%
9745 }

```

**\glsxtrheadlong** Command used to display long form in the page header.

```

9746 \newcommand*{\glsxtrheadlong}[1]{%
9747   \protect\NoCaseChange
9748   {%
9749     \glsifattribute{#1}{headuc}{true}%
9750     {%
9751       \GLSxtrlong[noindex,hyper=false]{#1}[]%
9752     }%
9753     {%
9754       \glsxtrlong[noindex,hyper=false]{#1}[]%
9755     }%
9756   }%
9757 }

```

**glsxtrtitlelong** Command to display long form of abbreviation in section title and table of contents.

```

9758 \newrobustcmd*{\glsxtrtitlelong}[1]{%
9759   \glsxtrlong[noindex,hyper=false]{#1}[]%
9760 }

```

`lsxtrheadlongpl` Command used to display plural long form in the page header. If you want the text converted to upper case, this needs to be redefined to use `\GLSxtrlongpl` instead. If you are using a smallcaps style, the default fonts don't provide italic smallcaps.

```

9761 \newcommand*{\glxtrheadlongpl}[1]{%
9762   \protect\NoCaseChange
9763   {%
9764     \glsifattribute{#1}{headuc}{true}%
9765     {%
9766       \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
9767     }%
9768     {%
9769       \glxtrlongpl[noindex,hyper=false]{#1}[]%
9770     }%
9771   }%
9772 }
```

`sxtrtitlelongpl` Command to display plural long form of abbreviation in section title and table of contents.

```

9773 \newrobustcmd*{\glxtrtitlelongpl}[1]{%
9774   \glxtrlongpl[noindex,hyper=false]{#1}[]%
9775 }
```

`\Glsxtrheadlong` Command used to display long form in the page header with the first letter converted to upper case.

```

9776 \newcommand*{\Glsxtrheadlong}[1]{%
9777   \protect\NoCaseChange
9778   {%
9779     \glsifattribute{#1}{headuc}{true}%
9780     {%
9781       \GLSxtrlong[noindex,hyper=false]{#1}[]%
9782     }%
9783     {%
9784       \Glsxtrlong[noindex,hyper=false]{#1}[]%
9785     }%
9786   }%
9787 }
```

`Glsxtrtitlelong` Command to display long form of abbreviation in section title and table of contents with the first letter converted to upper case.

```

9788 \newrobustcmd*{\Glsxtrtitlelong}[1]{%
9789   \Glsxtrlong[noindex,hyper=false]{#1}[]%
9790 }
```

`lsxtrheadlongpl` Command used to display plural long form in the page header with the first letter converted to upper case.

```

9791 \newcommand*{\Glsxtrheadlongpl}[1]{%
9792   \protect\NoCaseChange
9793   {%
9794     \glsifattribute{#1}{headuc}{true}%
9795     {%
9796       \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
9797     }%
9798     {%
9799       \glxtrlongpl[noindex,hyper=false]{#1}[]%
9800     }%
9801   }%
9802 }
```



```

9795   {%
9796     \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
9797   }%
9798   {%
9799     \Glsxtrlongpl[noindex,hyper=false]{#1}[]%
9800   }%
9801 }%
9802 }

```

`sxtrtitlelongpl` Command to display plural long form of abbreviation in section title and table of contents with the first letter converted to upper case.

```

9803 \newrobustcmd*{\Glsxtrtitlelongpl}[1]{%
9804   \Glsxtrlongpl[noindex,hyper=false]{#1}[]%
9805 }

```

`\glsxtrheadfull` Command used to display full form in the page header.

```

9806 \newcommand*{\glsxtrheadfull}[1]{%
9807   \protect\NoCaseChange
9808   {%
9809     \glsifattribute{#1}{headuc}{true}%
9810     {%
9811       \GLSxtrfull[noindex,hyper=false]{#1}[]%
9812     }%
9813     {%
9814       \glsxtrfull[noindex,hyper=false]{#1}[]%
9815     }%
9816   }%
9817 }

```

`glsxtrtitlefull` Command to display full form of abbreviation in section title and table of contents.

```

9818 \newrobustcmd*{\glsxtrtitlefull}[1]{%
9819   \glsxtrfull[noindex,hyper=false]{#1}[]%
9820 }

```

`lsxtrheadfullpl` Command used to display plural full form in the page header. If you want the text converted to upper case, this needs to be redefined to use `\GLSxtrfullpl` instead. If you are using a smallcaps style, the default fonts don't provide italic smallcaps.

```

9821 \newcommand*{\glsxtrheadfullpl}[1]{%
9822   \protect\NoCaseChange
9823   {%
9824     \glsifattribute{#1}{headuc}{true}%
9825     {%
9826       \GLSxtrfullpl[noindex,hyper=false]{#1}[]%
9827     }%
9828     {%
9829       \glsxtrfullpl[noindex,hyper=false]{#1}[]%
9830     }%
9831   }%
9832 }

```

`sxtrtitlefullpl` Command to display plural full form of abbreviation in section title and table of contents.

```

9833 \newrobustcmd*{\Glsxtrtitlefullpl}[1]{%
9834   \Glsxtrfullpl[noindex,hyper=false]{#1}[]%
9835 }

```

`\Glsxtrheadfull` Command used to display full form in the page header with the first letter converted to upper case.

```

9836 \newcommand*{\Glsxtrheadfull}[1]{%
9837   \protect\NoCaseChange
9838   {%
9839     \glsifattribute{#1}{headuc}{true}%
9840     {%
9841       \Glsxtrfull[noindex,hyper=false]{#1}[]%
9842     }%
9843     {%
9844       \Glsxtrfull[noindex,hyper=false]{#1}[]%
9845     }%
9846   }%
9847 }

```

`Glsxtrtitlefull` Command to display full form of abbreviation in section title and table of contents with the first letter converted to upper case.

```

9848 \newrobustcmd*{\Glsxtrtitlefull}[1]{%
9849   \Glsxtrfull[noindex,hyper=false]{#1}[]%
9850 }

```

`lGlsxtrheadfullpl` Command used to display plural full form in the page header with the first letter converted to upper case.

```

9851 \newcommand*{\Glsxtrheadfullpl}[1]{%
9852   \protect\NoCaseChange
9853   {%
9854     \glsifattribute{#1}{headuc}{true}%
9855     {%
9856       \Glsxtrfullpl[noindex,hyper=false]{#1}[]%
9857     }%
9858     {%
9859       \Glsxtrfullpl[noindex,hyper=false]{#1}[]%
9860     }%
9861   }%
9862 }

```

`sxtrtitlefullpl` Command to display plural full form of abbreviation in section title and table of contents with the first letter converted to upper case.

```

9863 \newrobustcmd*{\Glsxtrtitlefullpl}[1]{%
9864   \Glsxtrfullpl[noindex,hyper=false]{#1}[]%
9865 }

```

`\glsfmtshort` Provide a way of using the formatted short form in section headings. If `hyperref` has been loaded, use `\texorpdfstring` for convenience in PDF bookmarks.

```
9866 \ifdef\texorpdfstring
9867 {
9868   \newcommand*\glsfmtshort}[1]{%
9869     \texorpdfstring
9870       {\glsxtrtitleshort{#1}}%
9871       {\glsentryshort{#1}}%
9872   }
9873 }
9874 {
9875   \newcommand*\glsfmtshort}[1]{%
9876     \glsxtrtitleshort{#1}}
9877 }
```

Similarly for the plural version.

`\glsfmtshortpl`

```
9878 \ifdef\texorpdfstring
9879 {
9880   \newcommand*\glsfmtshortpl}[1]{%
9881     \texorpdfstring
9882       {\glsxtrtitleshortpl{#1}}%
9883       {\glsentryshortpl{#1}}%
9884   }
9885 }
9886 {
9887   \newcommand*\glsfmtshortpl}[1]{%
9888     \glsxtrtitleshortpl{#1}}
9889 }
```

The case-changing version isn't suitable for PDF bookmarks, so the PDF alternative uses the non-case-changing version.

`\Glsfmtshort` Singular form (first letter uppercase).

```
9890 \ifdef\texorpdfstring
9891 {
9892   \newcommand*\Glsfmtshort}[1]{%
9893     \texorpdfstring
9894       {\Glsxtrtitleshort{#1}}%
9895       {\glsentryshort{#1}}%
9896   }
9897 }
9898 {
9899   \newcommand*\Glsfmtshort}[1]{%
9900     \Glsxtrtitleshort{#1}}
9901 }
```

`\Glsfmtshortpl` Plural form (first letter uppercase).

```

9902 \ifdef\texorpdfstring
9903 {
9904   \newcommand*\Glsfmtshortpl}[1]{%
9905     \texorpdfstring
9906     {\Glsxtrtitleshortpl{#1}}%
9907     {\glsentryshortpl{#1}}%
9908   }
9909 }
9910 {
9911   \newcommand*\Glsfmtshortpl}[1]{%
9912     \Glsxtrtitleshortpl{#1}}
9913 }

```

`\glsfmttext` As above but for the text value.

```

9914 \ifdef\texorpdfstring
9915 {
9916   \newcommand*\glsfmttext}[1]{%
9917     \texorpdfstring
9918     {\glsxtrtitletext{#1}}%
9919     {\glsentrytext{#1}}%
9920   }
9921 }
9922 {
9923   \newcommand*\glsfmttext}[1]{%
9924     \glsxtrtitletext{#1}}
9925 }

```

`\Glsfmttext` First letter converted to upper case.

```

9926 \ifdef\texorpdfstring
9927 {
9928   \newcommand*\Glsfmttext}[1]{%
9929     \texorpdfstring
9930     {\Glsxtrtitletext{#1}}%
9931     {\glsentrytext{#1}}%
9932   }
9933 }
9934 {
9935   \newcommand*\Glsfmttext}[1]{%
9936     \Glsxtrtitletext{#1}}
9937 }

```

`\glsfmtplural` As above but for the plural value.

```

9938 \ifdef\texorpdfstring
9939 {
9940   \newcommand*\glsfmtplural}[1]{%
9941     \texorpdfstring
9942     {\glsxtrtitleplural{#1}}%
9943     {\glsentryplural{#1}}%
9944   }

```

```

9945 }
9946 {
9947   \newcommand*{\glsfmtplural}[1]{%
9948     \glsxtrtitleplural{#1}}
9949 }

```

`\Glsfmtplural` First letter converted to upper case.

```

9950 \ifdef\texorpdfstring
9951 {
9952   \newcommand*{\Glsfmtplural}[1]{%
9953     \texorpdfstring
9954       {\Glsxtrtitleplural{#1}}%
9955     {\glsentryplural{#1}}%
9956   }
9957 }
9958 {
9959   \newcommand*{\Glsfmtplural}[1]{%
9960     \Glsxtrtitleplural{#1}}
9961 }

```

`\glsfmtfirst` As above but for the first value.

```

9962 \ifdef\texorpdfstring
9963 {
9964   \newcommand*{\glsfmtfirst}[1]{%
9965     \texorpdfstring
9966       {\glsxtrtitlefirst{#1}}%
9967     {\glsentryfirst{#1}}%
9968   }
9969 }
9970 {
9971   \newcommand*{\glsfmtfirst}[1]{%
9972     \glsxtrtitlefirst{#1}}
9973 }

```

`\Glsfmtfirst` First letter converted to upper case.

```

9974 \ifdef\texorpdfstring
9975 {
9976   \newcommand*{\Glsfmtfirst}[1]{%
9977     \texorpdfstring
9978       {\Glsxtrtitlefirst{#1}}%
9979     {\glsentryfirst{#1}}%
9980   }
9981 }
9982 {
9983   \newcommand*{\Glsfmtfirst}[1]{%
9984     \Glsxtrtitlefirst{#1}}
9985 }

```

`\glsfmtfirstpl` As above but for the firstplural value.

```

9986 \ifdef\teorpdfstring
9987 {
9988   \newcommand*\glsfmtfirstpl}[1]{%
9989     \teorpdfstring
9990     {\glsxtrtitlefirstplural{#1}}%
9991     {\glsentryfirstplural{#1}}%
9992   }
9993 }
9994 {
9995   \newcommand*\glsfmtfirstpl}[1]{%
9996     \glsxtrtitlefirstplural{#1}}
9997 }

```

`\Glsfmtfirstpl` First letter converted to upper case.

```

9998 \ifdef\teorpdfstring
9999 {
10000   \newcommand*\Glsfmtfirstpl}[1]{%
10001     \teorpdfstring
10002     {\Glsxtrtitlefirstplural{#1}}%
10003     {\glsentryfirstplural{#1}}%
10004   }
10005 }
10006 {
10007   \newcommand*\Glsfmtfirstpl}[1]{%
10008     \Glsxtrtitlefirstplural{#1}}
10009 }

```

`\glsfmtlong` As above but for the long value.

```

10010 \ifdef\teorpdfstring
10011 {
10012   \newcommand*\glsfmtlong}[1]{%
10013     \teorpdfstring
10014     {\glsxtrtitlelong{#1}}%
10015     {\glsentrylong{#1}}%
10016   }
10017 }
10018 {
10019   \newcommand*\glsfmtlong}[1]{%
10020     \glsxtrtitlelong{#1}}
10021 }

```

`\Glsfmtlong` First letter converted to upper case.

```

10022 \ifdef\teorpdfstring
10023 {
10024   \newcommand*\Glsfmtlong}[1]{%
10025     \teorpdfstring
10026     {\Glsxtrtitlelong{#1}}%
10027     {\glsentrylong{#1}}%
10028   }

```

```

10029 }
10030 {
10031   \newcommand*{\Glsfmtlong}[1]{%
10032     \Glsxtrtitlelong{#1}}
10033 }

```

`\glsfmtlongpl` As above but for the longplural value.

```

10034 \ifdef\texorpdfstring
10035 {
10036   \newcommand*{\glsfmtlongpl}[1]{%
10037     \texorpdfstring
10038     {\Glsxtrtitlelongpl{#1}}%
10039     {\Glsentrylongpl{#1}}%
10040   }
10041 }
10042 {
10043   \newcommand*{\glsfmtlongpl}[1]{%
10044     \Glsxtrtitlelongpl{#1}}
10045 }

```

`\Glsfmtlongpl` First letter converted to upper case.

```

10046 \ifdef\texorpdfstring
10047 {
10048   \newcommand*{\Glsfmtlongpl}[1]{%
10049     \texorpdfstring
10050     {\Glsxtrtitlelongpl{#1}}%
10051     {\Glsentrylongpl{#1}}%
10052   }
10053 }
10054 {
10055   \newcommand*{\Glsfmtlongpl}[1]{%
10056     \Glsxtrtitlelongpl{#1}}
10057 }

```

`\glsfmtfull` In-line full format.

```

10058 \ifdef\texorpdfstring
10059 {
10060   \newcommand*{\glsfmtfull}[1]{%
10061     \texorpdfstring
10062     {\Glsxtrtitlefull{#1}}%
10063     {\Glsxtrinlinefullformat{#1}{}}%
10064   }
10065 }
10066 {
10067   \newcommand*{\glsfmtfull}[1]{%
10068     \Glsxtrtitlefull{#1}}
10069 }

```

`\Glsfmtfull` First letter converted to upper case.

```

10070 \ifdef\txorpdfstring
10071 {
10072   \newcommand*{\Glsfmtfull}[1]{%
10073     \txorpdfstring
10074     {\Glsxtrtitlefull{#1}}%
10075     {\Glsxtrinlinefullformat{#1}{}}%
10076   }
10077 }
10078 {
10079   \newcommand*{\Glsfmtfull}[1]{%
10080     \Glsxtrtitlefull{#1}}
10081 }

```

`\glsfmtfullpl` In-line full plural format.

```

10082 \ifdef\txorpdfstring
10083 {
10084   \newcommand*{\glsfmtfullpl}[1]{%
10085     \txorpdfstring
10086     {\glsxtrtitlefullpl{#1}}%
10087     {\glsxtrinlinefullplformat{#1}{}}%
10088   }
10089 }
10090 {
10091   \newcommand*{\glsfmtfullpl}[1]{%
10092     \glsxtrtitlefullpl{#1}}
10093 }

```

`\Glsfmtfullpl` First letter converted to upper case.

```

10094 \ifdef\txorpdfstring
10095 {
10096   \newcommand*{\Glsfmtfullpl}[1]{%
10097     \txorpdfstring
10098     {\Glsxtrtitlefullpl{#1}}%
10099     {\Glsxtrinlinefullplformat{#1}{}}%
10100   }
10101 }
10102 {
10103   \newcommand*{\Glsfmtfullpl}[1]{%
10104     \Glsxtrtitlefullpl{#1}}
10105 }

```

## 1.8 Multi-Lingual Support

Add the facility to load language modules, if they are installed, but none are provided with this package.

`sariesExtraLang`

```

10106 \newcommand*{\RequireGlossariesExtraLang}[1]{%

```



```

10107 \ifundefined{ver@glossariesxtr-#1.ldf}{\input{glossariesxtr-#1.ldf}}{}%
10108 }

```

sariesExtraLang

```

10109 \newcommand*{\ProvidesGlossariesExtraLang}[1]{%
10110 \ProvidesFile{glossariesxtr-#1.ldf}%
10111 }

```

Load any required language modules that are available. This doesn't generate any warning if none are found, since they're not essential. (The only command that really needs defining for the document is \abbreviationsname, which can simply be redefined.)

```

10112 \ifpackageloaded{tracklang}
10113 {%
10114 \AnyTrackedLanguages
10115 {%
10116 \ForEachTrackedDialect{\this@dialect}{%
10117 \IfTrackedLanguageFileExists{\this@dialect}%
10118 {glossariesxtr-}% prefix
10119 {.ldf}%
10120 {%
10121 \RequireGlossariesExtraLang{\CurrentTrackedTag}%
10122 }%
10123 {%
10124 }%
10125 }%
10126 }%
10127 {}%
10128 }
10129 {}

```

Load glossaries-extra-stylemods if required.

```

10130 \@glsxtr@redefstyles

```

and set the style:

```

10131 \@glsxtr@do@style

```

## 2 Style Adjustments (glossaries-extra-stylemods.sty)

This package adjusts the predefined styles so that they include the post description hook. Also, some other minor adjustments may be made to make existing styles more flexible.

### 2.1 Package Initialisation

First identify package:

```
10132 \NeedsTeXFormat{LaTeX2e}
10133 \ProvidesPackage{glossaries-extra-stylemods}[2017/09/08 v1.19 (NLCT)]
```

Provide package options to automatically load required predefined styles. The simplest method is to just test for the existence of the file `glossary-⟨option⟩.sty`. Packages can't be loaded whilst the options are being processed, so save the list in `\@glsxtr@loadstyles`.

`sxtr@loadstyles`

```
10134 \newcommand*{\@glsxtr@loadstyles}{%
10135 \DeclareOption*{%
10136   \IfFileExists{glossary-\CurrentOption.sty}{
10137     {\eappto\@glsxtr@loadstyles{%
10138       \noexpand\RequirePackage{glossary-\CurrentOption}}}%
10139     {\PackageError{glossaries-extra-styles}%
10140       {Unknown option '\CurrentOption'}{}}
10141 }
```

Process the package options:

```
10142 \ProcessOptions
```

Load the required packages:

```
10143 \@glsxtr@loadstyles
```

Adjust the styles that the post description hook added, but only for styles that have already been defined. All the tree styles in `glossary-tree` include the post description hook, so they don't require adjustment. Similarly for `glossary-mcols` which builds on the tree styles.

In case we have an old version of `glossaries`:

`ewglossarystyle`

```
10144 \providecommand{\renewglossarystyle}[2]{%
10145   \ifcsundef{@glsstyle@#1}%
10146   {%
10147     \PackageError{glossaries-extra}{Glossary style '#1' isn't already defined}{}}%
```

```

10148 }%
10149 {%
10150 \csdef{@glsstyle@#1}{#2}%
10151 }%
10152 }

```

## 2.2 List-Like Styles

The list-like styles mostly already use the post description hook. Only the listdotted style need modifying.

```

10153 \ifcsdef{@glsstyle@listdotted}
10154 {%
10155 \renewglossarystyle{listdotted}{%
10156 \setglossarystyle{list}%
10157 \renewcommand*{\glossentry}[2]{%
10158 \item[]\makebox[\glslistdottedwidth][l]{%
10159 \glstryitem{##1}%
10160 \glstarget{##1}{\glossentryname{##1}}%
10161 \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}%
10162 \glossentrydesc{##1}\glspostdescription}%
10163 \renewcommand*{\subglossentry}[3]{%
10164 \item[]\makebox[\glslistdottedwidth][l]{%
10165 \glssubentryitem{##2}%
10166 \glstarget{##2}{\glossentryname{##2}}%
10167 \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}%
10168 \glossentrydesc{##2}\glspostdescription}%
10169 }
10170 }
10171 {}

```

The sublistdotted style doesn't display the description for top-level entries. Sub-level entries use the listdottedstyle.

## 2.3 Longtable Styles

The three and four column styles require adjustment, but not the two column styles.

```

10172 \ifcsdef{@glsstyle@long3col}
10173 {%
10174 \renewglossarystyle{long3col}{%
10175 \renewenvironment{theglossary}%
10176 {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
10177 {\end{longtable}}%
10178 \renewcommand*{\glossaryheader}{}%
10179 \renewcommand*{\glsgroupheading}[1]{}%
10180 \renewcommand{\glossentry}[2]{%
10181 \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10182 \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline

```

```

10183 }%
10184 \renewcommand{\subglossentry}[3]{%
10185     &
10186     \glssubentryitem{##2}%
10187     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10188     ##3\tabularnewline
10189 }%
10190 \renewcommand*\{ \glsgroupskip }{%
10191     \ifglsgnognogroupskip\else & & \tabularnewline\fi}%
10192 }
10193 }
10194 {}

```

Four column style:

```

10195 \ifcsdef{@glsstyle@long4col}
10196 {%
10197     \renewglossarystyle{long4col}{%
10198         \renewenvironment{theglossary}%
10199             {\begin{longtable}{llll}}%
10200             {\end{longtable}}%
10201         \renewcommand*\{ \glossaryheader }{%
10202             \renewcommand*\{ \glsgroupheading }[1]{%
10203                 \renewcommand{\glossentry}[2]{%
10204                     \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10205                     \glossentrydesc{##1}\glspostdescription &
10206                     \glossentrysymbol{##1} &
10207                     ##2\tabularnewline
10208                 }%
10209                 \renewcommand{\subglossentry}[3]{%
10210                     &
10211                     \glssubentryitem{##2}%
10212                     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10213                     \glossentrysymbol{##2} & ##3\tabularnewline
10214                 }%
10215                 \renewcommand*\{ \glsgroupskip }{%
10216                     \ifglsgnognogroupskip\else & & \tabularnewline\fi}%
10217             }
10218         }
10219     }

```

The styles in glossary-longbooktabs are all based on the styles in glossary-long, so no adjustments are needed for that package.

## 2.4 Long Ragged Styles

The three and four column styles require adjustment, but not the two column styles.

```

10220 \ifcsdef{@glsstyle@longragged3col}
10221 {%
10222     \renewglossarystyle{longragged3col}{%

```

```

10223 \renewenvironment{theglossary}%
10224     {\begin{longtable}{l>{\raggedright}p{\glsglwidth}%
10225       >{\raggedright}p{\glspagelistwidth}}}%
10226     {\end{longtable}}}%
10227 \renewcommand*{\glossaryheader}{}%
10228 \renewcommand*{\glsgroupheading}[1]{}%
10229 \renewcommand{\glossentry}[2]{%
10230     \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10231     \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
10232 }%
10233 \renewcommand{\subglossentry}[3]{%
10234     &
10235     \glssubentryitem{##2}%
10236     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10237     ##3\tabularnewline
10238 }%
10239 \renewcommand*{\glsgroupskip}{%
10240     \ifglsgnoglobskip\else & \tabularnewline\fi}%
10241 }
10242 }
10243 {}

```

Four column style:

```

10244 \ifcsdef{@glstyle@altlongragged4col}
10245 {%
10246     \renewglossarystyle{altlongragged4col}{%
10247         \renewenvironment{theglossary}%
10248             {\begin{longtable}{l>{\raggedright}p{\glsglwidth}l%
10249               >{\raggedright}p{\glspagelistwidth}}}%
10250             {\end{longtable}}}%
10251         \renewcommand*{\glossaryheader}{}%
10252         \renewcommand*{\glsgroupheading}[1]{}%
10253         \renewcommand{\glossentry}[2]{%
10254             \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10255             \glossentrydesc{##1}\glspostdescription & \glossentrysymbol{##1} &
10256             ##2\tabularnewline
10257         }%
10258         \renewcommand{\subglossentry}[3]{%
10259             &
10260             \glssubentryitem{##2}%
10261             \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10262             \glossentrysymbol{##2} & ##3\tabularnewline
10263         }%
10264         \renewcommand*{\glsgroupskip}{%
10265             \ifglsgnoglobskip\else & \tabularnewline\fi}%
10266     }
10267 }
10268 {}

```

## 2.5 Supertabular Styles

The three and four column styles require adjustment, but not the two column styles.

```
10269 \ifcsdef{@glsstyle@super3col}
10270 {%
10271   \renewglossarystyle{super3col}{%
10272     \renewenvironment{theglossary}%
10273       {\tablehead{}}\tabletail{}}%
10274     \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
10275     {\end{supertabular}}}%
10276   \renewcommand*{\glossaryheader}{}%
10277   \renewcommand*{\glsgroupheading}[1]{}%
10278   \renewcommand{\glossentry}[2]{%
10279     \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10280     \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
10281   }%
10282   \renewcommand{\subglossentry}[3]{%
10283     &
10284     \glssubentryitem{##2}%
10285     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10286     ##3\tabularnewline
10287   }%
10288   \renewcommand*{\glsgroupskip}{%
10289     \ifglsnogroupskip\else & \tabularnewline\fi}%
10290 }
10291 }
10292 {}
```

Four column styles:

```
10293 \ifcsdef{@glsstyle@super4col}
10294 {%
10295   \renewglossarystyle{super4col}{%
10296     \renewenvironment{theglossary}%
10297       {\tablehead{}}\tabletail{}}%
10298     \begin{supertabular}{llll}}}%
10299     \end{supertabular}}}%
10300   \renewcommand*{\glossaryheader}{}%
10301   \renewcommand*{\glsgroupheading}[1]{}%
10302   \renewcommand{\glossentry}[2]{%
10303     \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10304     \glossentrydesc{##1}\glspostdescription &
10305     \glossentrysymbol{##1} & ##2\tabularnewline
10306   }%
10307   \renewcommand{\subglossentry}[3]{%
10308     &
10309     \glssubentryitem{##2}%
10310     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10311     \glossentrysymbol{##2} & ##3\tabularnewline
10312   }%
10313   \renewcommand*{\glsgroupskip}{%
```

```

10314      \ifglsnogroupskip\else & & \tabularnewline\fi}%
10315  }
10316 }
10317 {}

```

## 2.6 Super Ragged Styles

The three and four column styles require adjustment, but not the two column styles.

```

10318 \ifcsdef{@glsstyle@superragged3col}
10319 {%
10320   \renewglossarystyle{superragged3col}{%
10321     \renewenvironment{theglossary}%
10322       {\tablehead{}\tabletail}%
10323       \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}%
10324         >{\raggedright}p{\glspagelistwidth}}}%
10325       {\end{supertabular}}}%
10326   \renewcommand*{\glossaryheader}{}%
10327   \renewcommand*{\glsgroupheading}[1]{}%
10328   \renewcommand{\glossentry}[2]{%
10329     \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10330     \glossentrydesc{##1}\glspostdescription &
10331     ##2\tabularnewline
10332   }%
10333   \renewcommand{\subglossentry}[3]{%
10334     &
10335     \glssubentryitem{##2}%
10336     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10337     ##3\tabularnewline
10338   }%
10339   \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else &
10340     &\tabularnewline\fi}%
10341 }
10342 }
10343 {}

```

Four columns:

```

10344 \ifcsdef{@glsstyle@altsuperragged4col}
10345 {%
10346   \renewglossarystyle{altsuperragged4col}{%
10347     \renewenvironment{theglossary}%
10348       {\tablehead{}\tabletail}%
10349       \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}l%
10350         >{\raggedright}p{\glspagelistwidth}}}%
10351       {\end{supertabular}}}%
10352   \renewcommand*{\glossaryheader}{}%
10353   \renewcommand{\glossentry}[2]{%
10354     \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
10355     \glossentrydesc{##1}\glspostdescription &
10356     \glossentrysymbol{##1} & ##2\tabularnewline

```

```

10357 }%
10358 \renewcommand{\subglossentry}[3]{%
10359     &
10360     \glssubentryitem{##2}%
10361     \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
10362     \glossentrysymbol{##2} & ##3\tabularnewline
10363 }%
10364 \renewcommand*{\glsgroupskip}{\ifglsgnোগroupskip\else & &
10365     &\tabularnewline\fi}%
10366 }
10367 }
10368 {}

```

## 2.7 Inline Style

The inline style is dealt with slightly differently. The `\glspostdescription` hook is actually in `\glspostinline`, which is called at the end of the glossary. The original definition of `\glspostinline` also includes a space, which is unnecessary. Here, instead of redefining the inline style, just redefine `\glspostinline` and `\glsinlinedescformat`.

```

10369 \ifdef{\@glsstyle@inline}
10370 {%
10371     \renewcommand*{\glspostinline}{.\spacefactor\sfcode'\.}
10372     Just use \glxtrpostdescription instead of \glspostdescription.
10373     \renewcommand*{\glsinlinedescformat}[3]{%
10374         \space#1\glxtrpostdescription}
10375     \renewcommand*{\glsinlinesubdescformat}[3]{%
10376         #1\glxtrpostdescription}
10377 }

```

## 2.8 Tree Styles

The `alttree` style is redefined to make it easier to made minor adjustments.

```

10378 \ifdef{\@glsstyle@alttree}
10379 {%

```

Only redefine this style if it's already been defined.

mbolDescLocation

```
\glxtraltrtreeSymbolDescLocation{\langle label \rangle}{\langle location list \rangle}
```

Layout the symbol, description and location for top-level entries.

```

10380 \newcommand{\glxtraltrtreeSymbolDescLocation}[2]{%
10381     {%
10382         \let\par\glxtrAltTreePar

```



```

10383      \ifglshassymbol{#1}{(\glossentrysymbol{#1})\space}{}%
10384      \glossentrydesc{#1}\glspostdescription \space #2\par
10385  }%
10386  }

```

`\trAltTreeIndent` Paragraph indent for subsequent paragraphs in multi-paragraph descriptions.

```

10387  \newlength\glxstrAltTreeIndent

```

`\lsxtrAltTreePar` Multi-paragraph descriptions need to keep the hanging indent.

```

10388  \newcommand{\glxstrAltTreePar}{%
10389    @@par
10390    \glxstrAltTreeSetHangIndent
10391    \setlength{\parindent}{\dimexpr\hangindent+\glxstrAltTreeIndent}%
10392  }

```

`\symbolDescLocation` `\glxstralttreeSubSymbolDescLocation{<level>}{<label>}{<location list>}`

Layout the symbol, description and location for sub-entries. Defaults to the same as the top-level.

```

10393  \newcommand{\glxstralttreeSubSymbolDescLocation}[3]{%
10394    \glxstralttreeSymbolDescLocation{#2}{#3}%
10395  }

```

`\trtreeTopindent` The original style has to keep computing the width of the name at each entry. This register allows the style to compute it once for the top-level at the start of the glossary.

```

10396  \newlength\glxstrtreeTopindent

```

`\glxstralttreeInit` User-level initialisation for the alttree style.

```

10397  \newcommand*{\glxstralttreeInit}{%
10398    \settowidth{\glxstrtreeTopindent}{\glstreenamefmt{\glsgetwidestname\space}}%
10399    \glxstrAltTreeIndent=\parindent
10400  }

```

`\eglissetwidest` The original `\glissetwidest` only uses `\def`. This uses `\protected@csedef`.

```

10401  \newcommand*{\eglissetwidest}[2][0]{%
10402    \protected@csedef{@glswidestname\romannumeral#1}{#2}%
10403  }

```

`\xglissetwidest` Like the above but uses `\protected@csxdef`.

```

10404  \newcommand*{\xglissetwidest}[2][0]{%
10405    \protected@csxdef{@glswidestname\romannumeral#1}{#2}%
10406  }

```

`\lsgetwidestname` Provide a user-level macro to obtain the widest top-level name.

```

10407  \newcommand*{\lsgetwidestname}{\@glswidestname}

```

`etwidestsubname` Provide a user-level macro to obtain the widest sub-entry name.

```

10408 \newcommand*\glsgetwidestsubname}[1]{%
10409   \ifcsundef{@glswidestname\romannumeral#1}%
10410     {\@glswidestname}%
10411     {\csuse{@glswidestname\romannumeral#1}}%
10412 }

```

`estTopLevelName` CamelCase is easier for long command names. Provide a CamelCase synonym of `\glsfindwidesttoplevelname`

```

10413 \let\glsFindWidestTopLevelName\glsfindwidesttoplevelname

```

`sedTopLevelName` Like `\glsfindwidesttoplevelname` but has an additional check that the entry has been used. Only useful if the glossaries occur at the end of the document, in which case this command should go at the start of the glossary. Alternatively, place at the end of the document and save for the next run.

```

10414 \newrobustcmd*\glsFindWidestUsedTopLevelName}[1][\@glo@types]{%
10415   \dimen@=0pt\relax
10416   \gls@tmplen=0pt\relax
10417   \foralllglossaries[#1]{\@gls@type}%
10418   {%
10419     \forglsentries[\@gls@type]{\@glo@label}%
10420     {%
10421       \ifglsused{\@glo@label}%
10422       {%
10423         \ifglshasparent{\@glo@label}%
10424         {}%
10425         {%
10426           \settowidth{\dimen@}%
10427             {\glstreenamfmt{\glsentryname{\@glo@label}}}%
10428           \ifdim\dimen@>\gls@tmplen
10429             \gls@tmplen=\dimen@
10430           \eglssetwidest{\glsentryname{\@glo@label}}%
10431           \fi
10432         }%
10433       }%
10434     }%
10435   }%
10436 }%
10437 }

```

`destUsedAnyName` Like the above but doesn't check the parent key. Useful if all levels should have the same width for the name.

```

10438 \newrobustcmd*\glsFindWidestUsedAnyName}[1][\@glo@types]{%
10439   \dimen@=0pt\relax
10440   \gls@tmplen=0pt\relax
10441   \foralllglossaries[#1]{\@gls@type}%
10442   {%
10443     \forglsentries[\@gls@type]{\@glo@label}%
10444     {%

```

```

10445     \ifglsused{\@glo@label}%
10446     {%
10447         \settowidth{\dimen@}%
10448         {\glstreenamefmt{\glentryname{\@glo@label}}}%
10449         \ifdim\dimen@>\gls@tmplen
10450             \gls@tmplen=\dimen@
10451             \eglssetwidest{\glentryname{\@glo@label}}%
10452         \fi
10453     }%
10454     {%
10455     }%
10456 }%
10457 }

```

**FindWidestAnyName** Like the above but doesn't check if the entry has been used.

```

10458 \newrobustcmd*{\glsFindWidestAnyName}[1][\@glo@types]{%
10459     \dimen@=0pt\relax
10460     \gls@tmplen=0pt\relax
10461     \forallglossaries[#1]{\@gls@type}%
10462     {%
10463         \forallglsentries[\@gls@type]{\@glo@label}%
10464         {%
10465             \settowidth{\dimen@}%
10466             {\glstreenamefmt{\glentryname{\@glo@label}}}%
10467             \ifdim\dimen@>\gls@tmplen
10468                 \gls@tmplen=\dimen@
10469                 \eglssetwidest{\glentryname{\@glo@label}}%
10470             \fi
10471         }%
10472     }%
10473 }

```

**FindUsedLevelTwo** This is like `\glsFindWidestUsedTopLevelName` but also sets the first two sub-levels as well.

Any entry that has a great-grandparent is ignored.

```

10474 \newrobustcmd*{\glsFindWidestUsedLevelTwo}[1][\@glo@types]{%
10475     \dimen@=0pt\relax
10476     \dimen@i=0pt\relax
10477     \dimen@ii=0pt\relax
10478     \forallglossaries[#1]{\@gls@type}%
10479     {%
10480         \forallglsentries[\@gls@type]{\@glo@label}%
10481         {%
10482             \ifglsused{\@glo@label}%
10483             {%
10484                 \ifglshasparent{\@glo@label}%
10485                 {%
10486                     \edef\@glo@parent{\csuse{glo@glsetoklabel}{\@glo@label}@parent}}%
10487                     \ifglshasparent{\@glo@parent}%
10488                 }%

```

```

10489         \edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@parent}@parent}}}%
10490         \ifglshasparent{\@glo@parent}%
10491         {%
10492         {%
10493             \settowidth{\gls@tmplen}%
10494             {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10495             \ifdim\gls@tmplen>\dimen@ii
10496             \dimen@ii=\gls@tmplen
10497             \eglssetwidest[2]{\glsentryname{\@glo@label}}%
10498             \fi
10499         }%
10500     }%
10501     {%
10502         \settowidth{\gls@tmplen}%
10503         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10504         \ifdim\gls@tmplen>\dimen@i
10505         \dimen@i=\gls@tmplen
10506         \eglssetwidest[1]{\glsentryname{\@glo@label}}%
10507         \fi
10508     }%
10509 }%
10510 {%
10511     \settowidth{\gls@tmplen}%
10512     {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10513     \ifdim\gls@tmplen>\dimen@
10514     \dimen@=\gls@tmplen
10515     \eglssetwidest{\glsentryname{\@glo@label}}%
10516     \fi
10517 }%
10518 }%
10519 {}%
10520 }%
10521 }%
10522 }

```

`\dWidestLevelTwo` This is like `\glsFindWidestUsedLevelTwo` but doesn't check if the entry has been used.

```

10523 \newrobustcmd*{\glsFindWidestLevelTwo}[1][\@glo@types]{%
10524     \dimen@=0pt\relax
10525     \dimen@i=0pt\relax
10526     \dimen@ii=0pt\relax
10527     \foralllglossaries[#1]{\@gls@type}%
10528     {%
10529         \forglsentries[\@gls@type]{\@glo@label}%
10530         {%
10531             \ifglshasparent{\@glo@label}%
10532             {%
10533                 \edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@label}@parent}}%
10534                 \ifglshasparent{\@glo@parent}%
10535                 {%

```

```

10536      \edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@parent}@parent}}%
10537      \ifglshasparent{\@glo@parent}%
10538      {}%
10539      {%
10540          \settowidth{\gls@tmplen}%
10541              {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10542          \ifdim\gls@tmplen>\dimen@ii
10543              \dimen@ii=\gls@tmplen
10544              \eglssetwidest[2]{\glsentryname{\@glo@label}}%
10545          \fi
10546      }%
10547  }%
10548  {%
10549      \settowidth{\gls@tmplen}%
10550          {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10551      \ifdim\gls@tmplen>\dimen@i
10552          \dimen@i=\gls@tmplen
10553          \eglssetwidest[1]{\glsentryname{\@glo@label}}%
10554      \fi
10555  }%
10556  }%
10557  {%
10558      \settowidth{\gls@tmplen}%
10559          {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10560      \ifdim\gls@tmplen>\dimen@
10561          \dimen@=\gls@tmplen
10562          \eglssetwidest{\glsentryname{\@glo@label}}%
10563      \fi
10564  }%
10565  }%
10566  }%
10567  }

```

**edAnyNameSymbol** Like the `\glsFindWidestUsedAnyName` but also measures the symbol. The length of the widest symbol is stored in the second argument should be a length register.

```

10568 \newrobustcmd*{\glsFindWidestUsedAnyNameSymbol}[2][\@glo@types]{%
10569     \dimen@=0pt\relax
10570     \gls@tmplen=0pt\relax
10571     #2=0pt\relax
10572     \forallglossaries[#1]{\@gls@type}%
10573     {%
10574         \forglsentries[\@gls@type]{\@glo@label}%
10575         {%
10576             \ifglused{\@glo@label}%
10577             {%
10578                 \settowidth{\dimen@}%
10579                     {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10580                 \ifdim\dimen@>\gls@tmplen
10581                     \gls@tmplen=\dimen@

```

```

10582         \eglssetwidest{\glsentryname{\@glo@label}}}%
10583     \fi
10584     \settowidth{\dimen@}%
10585     {\glsentrysymbol{\@glo@label}}}%
10586     \ifdim\dimen@>#2\relax
10587         #2=\dimen@
10588     \fi
10589 }%
10590 {}%
10591 }%
10592 }%
10593 }

```

**stAnyNameSymbol** Like the above but doesn't check if the entry has been used.

```

10594 \newrobustcmd*{\glsFindWidestAnyNameSymbol}[2][\@glo@types]{%
10595     \dimen@=0pt\relax
10596     \gls@tmplen=0pt\relax
10597     #2=0pt\relax
10598     \forallglossaries[#1]{\@gls@type}%
10599     {%
10600         \forglsentries[\@gls@type]{\@glo@label}%
10601         {%
10602             \settowidth{\dimen@}%
10603             {\glstreenamfmt{\glsentryname{\@glo@label}}}%
10604             \ifdim\dimen@>\gls@tmplen
10605                 \gls@tmplen=\dimen@
10606                 \eglssetwidest{\glsentryname{\@glo@label}}}%
10607             \fi
10608             \settowidth{\dimen@}%
10609             {\glsentrysymbol{\@glo@label}}}%
10610             \ifdim\dimen@>#2\relax
10611                 #2=\dimen@
10612             \fi
10613         }%
10614     }%
10615 }

```

**eSymbolLocation** Like the `\glsFindWidestUsedAnyNameSymbol` but also measures the location list. This requires `\glsentrynumberlist`. The length of the widest symbol is stored in the second argument should be a length register. The length of the widest location list is stored in the third argument, which should also be a length register.

```

10616 \newrobustcmd*{\glsFindWidestUsedAnyNameSymbolLocation}[3][\@glo@types]{%
10617     \dimen@=0pt\relax
10618     \gls@tmplen=0pt\relax
10619     #2=0pt\relax
10620     #3=0pt\relax
10621     \forallglossaries[#1]{\@gls@type}%
10622     {%
10623         \forglsentries[\@gls@type]{\@glo@label}%

```

```

10624    {%
10625        \ifglsused{\@glo@label}%
10626    {%
10627        \settowidth{\dimen@}%
10628            {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10629        \ifdim\dimen@>\gls@tmplen
10630            \gls@tmplen=\dimen@
10631            \eglssetwidest{\glsentryname{\@glo@label}}%
10632        \fi
10633        \settowidth{\dimen@}%
10634            {\glsentrysymbol{\@glo@label}}%
10635        \ifdim\dimen@>#2\relax
10636            #2=\dimen@
10637        \fi
10638        \settowidth{\dimen@}%
10639            {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
10640        \ifdim\dimen@>#3\relax
10641            #3=\dimen@
10642        \fi
10643    }%
10644    {%
10645    }%
10646    }%
10647 }

```

**eSymbolLocation** Like the `\glsFindWidestUsedAnyNameSymbol` but doesn't check if the entry has been used.

```

10648 \newrobustcmd*{\glsFindWidestAnyNameSymbolLocation}[3][\@glo@types]{%
10649     \dimen@=0pt\relax
10650     \gls@tmplen=0pt\relax
10651     #2=0pt\relax
10652     #3=0pt\relax
10653     \foralllglossaries[#1]{\@gls@type}%
10654     {%
10655         \forglsentries[\@gls@type]{\@glo@label}%
10656     {%
10657         \settowidth{\dimen@}%
10658             {\glstreenamefmt{\glsentryname{\@glo@label}}}%
10659         \ifdim\dimen@>\gls@tmplen
10660             \gls@tmplen=\dimen@
10661             \eglssetwidest{\glsentryname{\@glo@label}}%
10662         \fi
10663         \settowidth{\dimen@}%
10664             {\glsentrysymbol{\@glo@label}}%
10665         \ifdim\dimen@>#2\relax
10666             #2=\dimen@
10667         \fi
10668         \settowidth{\dimen@}%
10669             {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
10670         \ifdim\dimen@>#3\relax

```

```

10671         #3=\dimen@
10672     \fi
10673 }%
10674 }%
10675 }

```

**AnyNameLocation** Like the `\glsFindWidestUsedAnyNameSymbolLocation` but doesn't measure the symbol. The length of the widest location list is stored in the second argument, which should be a length register.

```

10676 \newrobustcmd*{\glsFindWidestUsedAnyNameLocation}[2][\@glo@types]{%
10677     \dimen@=0pt\relax
10678     \gls@tmplen=0pt\relax
10679     #2=0pt\relax
10680     \forallglossaries[#1]{\@gls@type}%
10681     {%
10682         \forallglsentries[\@gls@type]{\@glo@label}%
10683         {%
10684             \ifglsused{\@glo@label}%
10685             {%
10686                 \settowidth{\dimen@}%
10687                     {\glstreenamfmt{\glsentryname{\@glo@label}}}%
10688                 \ifdim\dimen@>\gls@tmplen
10689                     \gls@tmplen=\dimen@
10690                     \eglssetwidest{\glsentryname{\@glo@label}}%
10691                 \fi
10692                 \settowidth{\dimen@}%
10693                     {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
10694                 \ifdim\dimen@>#2\relax
10695                     #2=\dimen@
10696                 \fi
10697             }%
10698         }%
10699     }%
10700 }%
10701 }

```

**AnyNameLocation** Like the `\glsFindWidestAnyNameLocation` but doesn't check the **first use** flag.

```

10702 \newrobustcmd*{\glsFindWidestAnyNameLocation}[2][\@glo@types]{%
10703     \dimen@=0pt\relax
10704     \gls@tmplen=0pt\relax
10705     #2=0pt\relax
10706     \forallglossaries[#1]{\@gls@type}%
10707     {%
10708         \forallglsentries[\@gls@type]{\@glo@label}%
10709         {%
10710             \settowidth{\dimen@}%
10711                 {\glstreenamfmt{\glsentryname{\@glo@label}}}%
10712             \ifdim\dimen@>\gls@tmplen
10713                 \gls@tmplen=\dimen@

```



```

10714         \eglssetwidest{\glstryname{\@glo@label}}}%
10715     \fi
10716     \settowidth{\dimen@}%
10717         {\GlsXtrFormatLocationList{\glstrynumberlist{\@glo@label}}}%
10718     \ifdim\dimen@>#2\relax
10719         #2=\dimen@
10720     \fi
10721 }%
10722 }%
10723 }

```

**computeTreeIndent** Compute the value of `\glstreeindent`. Argument is the entry label. (Ignored in default definition, but this command may be redefined to take the particular entry into account.) Note that the sub-levels modify `\glstreeindent`.

```

10724 \newcommand*\glsxtrComputeTreeIndent}[1]{%
10725     \glstreeindent=\glsxtrtreetopindent\relax
10726 }

```

**computeTreeSubIndent** `\glsxtrComputeTreeSubIndent{<level>}{<label>}{<register>}`

Compute the indent for the sub-entries. The first argument is the level, the second argument is the entry label and the third argument is the length register used to store the computed indent.

```

10727 \newcommand*\glsxtrComputeTreeSubIndent}[3]{%
10728     \ifcsundef{@glswidestname\romannumeral#1}%
10729     {%
10730         \settowidth{#3}{\glstreenamfmt{\@glswidestname\space}}%
10731     }%
10732     {%
10733         \settowidth{#3}{\glstreenamfmt{%
10734             \csname @glswidestname\romannumeral#1\endcsname\space}}%
10735     }%
10736 }

```

**treeSetHangIndent** Set `\hangindent` for top-level entries:

```

10737 \newcommand*\glsxtrAltTreeSetHangIndent{\hangindent\glstreeindent}

```

**treeSetSubHangIndent** Set `\hangindent` for sub-entries:

```

10738 \newcommand*\glsxtrAltTreeSetSubHangIndent}[1]{\hangindent\glstreeindent}

```

Redefine `alttree`:

```

10739 \renewglossarystyle{alttree}{%
10740     \renewenvironment{theglossary}%
10741     {%
10742         \glsxtralttreeInit

```

```

10743     \def\@gls@prevlevel{-1}%
10744     \mbox{}\par}%
10745     {\par}%
10746     \renewcommand*\glossaryheader{}%
10747     \renewcommand*\glsgroupheading[1]{}%
10748     \renewcommand\glossentry[2]{%
10749         \ifnum\@gls@prevlevel=0\relax
10750         \else
10751             \glstrComputeTreeIndent{##1}%
10752         \fi
10753         \parindent\glstreeindent
10754         \glstrAltTreeSetHangIndent
10755         \makebox[Opt][r]%
10756         {%
10757             \glstreenamebox{\glstreeindent}%
10758             {%
10759                 \glstryitem{##1}%
10760                 \glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
10761             }%
10762         }%
10763         \glstralttreeSymbolDescLocation{##1}{##2}%
10764     \def\@gls@prevlevel{0}%
10765 }
10766 \renewcommand\subglossentry[3]{%
10767     \ifnum##1=1\relax
10768         \glssubentryitem{##2}%
10769     \fi
10770     \ifnum\@gls@prevlevel=##1\relax
10771     \else
10772         \glstrComputeTreeSubIndent{##1}{##2}{\@gls@tmplen}%
10773         \ifnum\@gls@prevlevel<##1\relax
10774             \setlength\glstreeindent\@gls@tmplen
10775             \addtolength\glstreeindent\parindent
10776             \parindent\glstreeindent
10777         \else
10778             \ifnum\@gls@prevlevel=0\relax
10779                 \glstrComputeTreeIndent{##2}%
10780             \else
10781                 \glstrComputeTreeSubIndent{\@gls@prevlevel}{##2}{\glstreeindent}%
10782             \fi
10783             \addtolength\parindent{-\glstreeindent}%
10784             \setlength\glstreeindent\parindent
10785         \fi
10786     \fi
10787     \glstrAltTreeSetSubHangIndent{##1}%
10788     \makebox[Opt][r]{\glstreenamebox{\@gls@tmplen}{%
10789         \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
10790     \glstralttreeSubSymbolDescLocation{##1}{##2}{##3}%
10791     \def\@gls@prevlevel{##1}%

```

```

10792     }%
10793     \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
10794   }
10795 }%
10796 {%

```

Assume the style isn't required if it hasn't already been defined.

```

10797 }

```

Reset the default style

```

10798 \ifx\@glossary@default@style\relax
10799 \else
10800   \setglossarystyle{\@glxtr@current@style}
10801 \fi

```

# Glossary

**First use** The first time a glossary entry is used (from the start of the document or after a reset) with one of the following commands: `\gls`, `\Gls`, `\GLS`, `\glspl`, `\Glspl`, `\GLSpl` or `\glsdisp`. *see* **first use flag** & **first use text**

**First use flag** A conditional that determines whether or not the entry has been used according to the rules of **first use**.

**First use text** The text that is displayed on **first use**, which is governed by the first and first-plural keys of `\newglossaryentry`. (May be overridden by `\glsdisp`.)

`makeindex` An indexing application.

`xindy` An flexible indexing application with multilingual support written in Perl.

## Change History

|   |     |                                       |     |
|---|-----|---------------------------------------|-----|
| 0.1 (2015-11-22)                          |     | \@Glsymbol@: added redefinition       | 57  |
| General: Initial experimental release     | 5   | \@Glsymbolplural@: added              |     |
| 0.2 (2015-11-30)                          |     | redefinition                          | 58  |
| \Glsfmtshort: new                         | 275 | \@Glstext@: added redefinition        | 53  |
| \glsfmtshort: new                         | 275 | \@Glsuseri@: added redefinition       | 58  |
| \Glsfmtshortpl: new                       | 275 | \@Glsuserii@: added redefinition      | 58  |
| \glsfmtshortpl: new                       | 275 | \@Glsuseriii@: added redefinition     | 59  |
| short: switched inline full form to short |     | \@Glsuseriv@: added redefinition      | 59  |
| (long)                                    | 183 | \@Glsuserv@: added redefinition       | 59  |
| 0.3 (2015-12-02)                          |     | \@Glsuservi@: added redefinition      | 60  |
| \@ACRlong: added redefinition             | 63  | \@acrlong: added redefinition         | 62  |
| \@ACRlongpl: added redefinition           | 64  | \@acrlongpl: added redefinition       | 63  |
| \@ACRshort: added redefinition            | 61  | \@acrshort: added redefinition        | 60  |
| \@ACRshortpl: added redefinition          | 62  | \@acrshortpl: added redefinition      | 61  |
| \@Acrlong: added redefinition             | 62  | \@gls@field@link: added optional      |     |
| \@Acrlongpl: added redefinition           | 63  | argument                              | 48  |
| \@Acrshort: added redefinition            | 60  | \@glsdescplural@: added redefinition  | 57  |
| \@Acrshortpl: added redefinition          | 61  | \@glsfirst@: added redefinition       | 54  |
| \@GLSdesc@: added redefinition            | 56  | \@glsfirstplural@: added redefinition | 55  |
| \@GLSdescplural@: added redefinition      | 57  | \@glsplural@: added redefinition      | 54  |
| \@GLSfirst@: added redefinition           | 54  | \@glssymbolplural@: added             |     |
| \@GLSfirstplural@: added redefinition     | 55  | redefinition                          | 58  |
| \@GLSname@: added redefinition            | 56  | \@glsxtr@defaultnoglossarywarning:    |     |
| \@GLSplural@: added redefinition          | 55  | new                                   | 111 |
| \@GLSsymbol@: added redefinition          | 57  | \@glsxtr@field@linkdefs: new          | 52  |
| \@GLSsymbolplural@: added                 |     | \@glsxtr@insertdots: new              | 154 |
| redefinition                              | 58  | \@print@glossary: added redefinition  | 108 |
| \@GLStext@: added redefinition            | 53  | \glsabbrvdefaultfont: renamed from    |     |
| \@GLSuseri@: added redefinition           | 58  | \abbrvdefaultfont                     | 159 |
| \@GLSuserii@: added redefinition          | 59  | \glsaccessdesc: new                   | 123 |
| \@GLSuseriii@: added redefinition         | 59  | \glsaccessdescplural: new             | 124 |
| \@GLSuseriv@: added redefinition          | 59  | \glsaccessfirst: new                  | 121 |
| \@GLSuserv@: added redefinition           | 60  | \glsaccessfirstplural: new            | 121 |
| \@GLSuservi@: added redefinition          | 60  | \Glsaccesslong: new                   | 126 |
| \@Glsdesc@: added redefinition            | 56  | \glsaccesslong: new                   | 125 |
| \@Glsdescplural@: added redefinition      | 57  | \glsaccessname: new                   | 119 |
| \@Glsfirst@: added redefinition           | 54  | \glsaccessplural: new                 | 120 |
| \@Glsfirstplural@: added redefinition     | 55  | \Glsaccessshort: new                  | 124 |
| \@Glsname@: added redefinition            | 56  | \glsaccessshort: new                  | 124 |
| \@Glsplural@: added redefinition          | 54  | \Glsaccessshortpl: new                | 125 |

|  |     |                                    |     |
|--|-----|------------------------------------|-----|
| \glsaccesssshortpl: new .....          | 125 | \cGLSpl: new .....                 | 87  |
| \glsaccessssymbol: new .....           | 122 | \cGLSpl@: new .....                | 87  |
| \glsaccessssymbolplural: new .....     | 122 | \glsxtr@setentrycountunsetattr:    |     |
| \glsaccesstext: new .....              | 120 | new .....                          | 82  |
| \glsentryfmt: added check for short .. | 47  | \cGLS: new .....                   | 86  |
| \glslongpltok: new .....               | 153 | \cGLSformat: new .....             | 87  |
| \glsshortpltok: new .....              | 153 | \cGLSpl: new .....                 | 87  |
| \glsxtr@newabbreviation: fixed family  |     | \cGLSplformat: new .....           | 87  |
| name in \setkeys .....                 | 155 | \GlossariesExtraWarningNoLine:     |     |
| \glsxtrdiscardperiod: added check      |     | new .....                          | 14  |
| for plural .....                       | 150 | \glsenableentrycount: new .....    | 82  |
| \GLSxtrlongpl: new .....               | 168 | \glsfirstabbrvdefaultfont: new ..  | 159 |
| \Glsxtrlongpl: new .....               | 168 | \glsfirstlongdefaultfont: new ...  | 159 |
| \glsxtrlongpl: new .....               | 167 | \Glsfmtfirst: new .....            | 277 |
| \glsxtrNoGlossaryWarning: new ....     | 18  | \glsfmtfirst: new .....            | 277 |
| \glsxtrpostlinkAddDescOnFirstUse:      |     | \Glsfmtfirstpl: new .....          | 278 |
| new .....                              | 150 | \glsfmtfirstpl: new .....          | 277 |
| \glsxtrpostlinkAddSymbolOnFirstUse:    |     | \Glsfmtplural: new .....           | 277 |
| new .....                              | 150 | \glsfmtplural: new .....           | 276 |
| \glsxtrpostlinkendsentence: new ..     | 150 | \Glsfmtshort: changed to use       |     |
| \GLSxtrshortpl: new .....              | 167 | \Glsxtrtitleshort .....            | 275 |
| \Glsxtrshortpl: new .....              | 166 | renamed from \Glsentryfmtshort ..  | 275 |
| \glsxtrshortpl: new .....              | 166 | \glsfmtshort: changed to use       |     |
| short-long-desc: fixed name to use     |     | \glsxtrtitleshort .....            | 275 |
| \glslabeltok .....                     | 178 | renamed from \glsentryfmtshort ..  | 275 |
| long-short-desc: fixed name to use     |     | \Glsfmtshortpl: changed to use     |     |
| \glslabeltok .....                     | 176 | \Glsxtrtitleshortpl .....          | 275 |
| 0.4 (2015-12-03)                       |     | renamed from                       |     |
| \@glsxtr@doabbreviationsdef: added     |     | \Glsentryfmtshortpl .....          | 275 |
| redefinition of \acronymtype .....     | 14  | \glsfmtshortpl: changed to use     |     |
| \Glsfmtshort: changed to use           |     | \glsxtrtitleshortpl .....          | 275 |
| \Glsxtrshort .....                     | 275 | renamed from                       |     |
| \glsfmtshort: changed to use           |     | \glsentryfmtshortpl .....          | 275 |
| \glsxtrshort .....                     | 275 | \Glsfmttext: new .....             | 276 |
| \Glsfmtshortpl: changed to use         |     | \glsfmttext: new .....             | 276 |
| \glsxtrshortpl .....                   | 275 | \glshasattribute: new .....        | 131 |
| \glsfmtshortpl: changed to use         |     | \glshascategoryattribute: new ...  | 130 |
| \glsxtrshortpl .....                   | 275 | \glsxtremsuffix: new .....         | 217 |
| \glsxtrifemptyglossary: new .....      | 21  | \GlsXtrEnableEntryCounting: new .. | 81  |
| \glsxtrnewnumber: added extra          |     | \glsxtrifcounttrigger: new .....   | 84  |
| argument .....                         | 134 | \glsxtrscfont: new .....           | 190 |
| \glsxtrnewsymbol: added extra          |     | \glsxtrscsuffix: new .....         | 190 |
| argument .....                         | 134 | \glsxtrsmfont: new .....           | 203 |
| \MakeAcronymsAbbreviations: set the    |     | \glsxtrsmsuffix: new .....         | 204 |
| default type to \acronymtype .....     | 95  | short-em: new .....                | 224 |
| \newterm: fixed name argument .....    | 133 | short-em-desc: new .....           | 225 |
| 0.5 (2015-12-07)                       |     | short-em-footnote: new .....       | 234 |
| \cGLS: new .....                       | 86  | short-em-long: new .....           | 220 |
| \cGLS@: new .....                      | 87  | short-em-long-desc: new .....      | 222 |

|                                     |     |   |     |
|-------------------------------------|-----|---|-----|
| short-em-postfootnote: new .....    | 236 | \glxtrheadshortpl: now uses headuc        |     |
| short-sc-footnote: new .....        | 200 | attribute .....                           | 267 |
| short-sc-postfootnote: new .....    | 201 | \Glsxtrheadtext: now uses headuc          |     |
| short-sm: new .....                 | 207 | attribute .....                           | 268 |
| short-sm-desc: new .....            | 208 | \glxtrheadtext: now uses headuc           |     |
| short-sm-footnote: new .....        | 213 | attribute .....                           | 268 |
| short-sm-long: new .....            | 205 | short-em-footnote: switch off regular     |     |
| short-sm-long-desc: new .....       | 206 | attribute if set .....                    | 234 |
| short-sm-postfootnote: new .....    | 215 | short-long: switch off regular attribute  |     |
| long-noshort-em: new .....          | 227 | if set .....                              | 177 |
| long-noshort-em-desc: new .....     | 230 | short-long-desc: switch off regular       |     |
| long-noshort-sm: new .....          | 210 | attribute if set .....                    | 179 |
| long-noshort-sm-desc: new .....     | 212 | short-sc-footnote: switch off regular     |     |
| long-short-em: new .....            | 217 | attribute if set .....                    | 200 |
| long-short-em-desc: new .....       | 218 | short-sm-footnote: switch off regular     |     |
| long-short-sm: new .....            | 204 | attribute if set .....                    | 213 |
| long-short-sm-desc: new .....       | 205 | long-short: switch off regular attribute  |     |
| 0.5.1 (2015-12-02)                  |     | if set .....                              | 175 |
| \Glsaccessstext: new .....          | 120 | long-short-desc: switch off regular       |     |
| 0.5.1 (2015-12-07)                  |     | attribute if set .....                    | 177 |
| \@glxtr@doaccsupp: new .....        | 17  | long-short-sc-desc: switch off regular    |     |
| General: removed \ifglxtruseuchead  | 266 | attribute if set .....                    | 191 |
| \Glsaccessdesc: new .....           | 123 | footnote: switch off regular attribute if |     |
| \Glsaccessdescplural: new .....     | 124 | set .....                                 | 180 |
| \Glsaccessfirst: new .....          | 121 | postfootnote: switch off regular          |     |
| \Glsaccessfirstplural: new .....    | 122 | attribute if set .....                    | 181 |
| \Glsaccessname: new .....           | 119 | 0.5.2 (2015-12-08)                        |     |
| \Glsaccessplural: new .....         | 120 | \@GLSdesc@: added accessibility support   | 56  |
| \Glsaccesssymbol: new .....         | 122 | \@GLSdescplural@: added accessibility     |     |
| \Glsaccesssymbolplural: new .....   | 123 | support .....                             | 57  |
| \Glsxtrheadfirst: now uses headuc   |     | \@GLSfirst@: added accessibility          |     |
| attribute .....                     | 270 | support .....                             | 54  |
| \glxtrheadfirst: now uses headuc    |     | \@GLSfirstplural@: added accessibility    |     |
| attribute .....                     | 270 | support .....                             | 55  |
| \Glsxtrheadfirstplural: now uses    |     | \@GLSname@: added accessibility support   | 56  |
| headuc attribute .....              | 271 | \@GLSplural@: added accessibility         |     |
| \glxtrheadfirstplural: now uses     |     | support .....                             | 55  |
| headuc attribute .....              | 270 | \@GLSsymbol@: added accessibility         |     |
| \Glsxtrheadplural: now uses headuc  |     | support .....                             | 57  |
| attribute .....                     | 269 | \@GLSsymbolplural@: added                 |     |
| \glxtrheadplural: now uses headuc   |     | accessibility support .....               | 58  |
| attribute .....                     | 269 | \@GLStext@: added accessibility support   | 53  |
| \Glsxtrheadshort: now uses headuc   |     | \@GLSdesc@: added accessibility support   | 56  |
| attribute .....                     | 267 | \@GLSdescplural@: added accessibility     |     |
| \glxtrheadshort: now uses headuc    |     | support .....                             | 57  |
| attribute .....                     | 266 | \@GLSfirst@: added accessibility          |     |
| \Glsxtrheadshortpl: now uses headuc |     | support .....                             | 54  |
| attribute .....                     | 267 | \@GLSfirstplural@: added accessibility    |     |
|                                     |     | support .....                             | 55  |

|  |          |                                       |          |
|--|----------|---------------------------------------|----------|
| \@Glsname@: add accessibility support ..             | 56       | \GLSaccesssymbolplural: new ..        | 123, 128 |
| \@Glsplural@: added accessibility support .....      | 54       | \GLSaccessstext: new .....            | 120, 127 |
| \@Glsymbol@: added accessibility support .....       | 57       | \glsentryfmt: moved                   |          |
| \@Glsymbolplural@: added accessibility support ..... | 58       | \glssetabbrvfmt from                  |          |
| \@Glstext@: added accessibility support              | 53       | \glsxtrabbrvfmt to here .....         | 47       |
| \@glsdesc@: added accessibility support              | 56       | \GlsXtrEnableInitialTagging: new      | 146      |
| \@glsdescplural@: added accessibility support .....  | 57       | \glsxtrfieldtitlecase: new .....      | 135      |
| \@glsfirst@: added accessibility support .....       | 54       | \GlsXtrFormatLocationList: new ...    | 45       |
| \@glsfirstplural@: added accessibility support ..... | 55       | \glsxtrnewabbrevpresetkeyhook:        |          |
| \@Glsname@: added accessibility support              | 56       | new .....                             | 157      |
| \@Glsplural@: added accessibility support .....      | 54       | \glsxtrtagfont: new .....             | 148      |
| \@Glsymbol@: added accessibility support .....       | 57       | \KV@printgloss@nonumberlist: added    | 47       |
| \@Glsymbolplural@: added accessibility support ..... | 58       | \mfu@checkword@do: added .....        | 147      |
| \@Glstext@: added accessibility support              | 53       | \setabbreviationstyle: added check    |          |
| \@glsxtr@activate@initialtagging:                    |          | for post-definition style switch .... | 172      |
| new .....  | 148      | 0.5.3 (2015-12-09)                    |          |
| \@glsxtr@do@titlecaps@warn: new ..                   | 148      | \@glsxtr@autoindex@at: new .....      | 144      |
| \@glsxtr@tag: new .....                              | 148      | \@glsxtr@autoindex@encap: new ...     | 144      |
| General: fixed typo in glossaries-accsupp            |          | \@glsxtr@autoindex@esc: new .....     | 145      |
| and tidied up code to use just one                   |          | \@glsxtr@autoindex@level: new ...     | 144      |
| \@ifpackageloaded .....                              | 119      | \@glsxtr@autoindex@setname: new ..    | 142      |
| removed \glsxtrabbrvfmt .....                        | 169      | \@glsxtr@doabbreviationsdef: new ..   | 14       |
| \glossaryentrynumbers: added .....                   | 45       | General: removed                      |          |
| \Glossentrydesc: added .....                         | 146      | \GlsXtrNoGlsWarningNoAutoMakeMain     |          |
| \Glossentryname: added .....                         | 140      | .....                                 | 110      |
| \Glossentrysymbol: added .....                       | 146      | \glsdescwidth: added .....            | 44       |
| \glossentrysymbol: added .....                       | 146      | \glspagelistwidth: added .....        | 44       |
| \GLSaccessdesc: new .....                            | 123, 128 | \glsxtrdoautoindexname: new .....     | 142      |
| \GLSaccessdescplural: new ...                        | 124, 128 | \glsxtrpostnamehook: new .....        | 141      |
| \GLSaccessfirst: new .....                           | 121, 127 | \if@glsxtr@format@override: new ..    | 141      |
| \GLSaccessfirstplural: new ..                        | 122, 127 | \ProvidesGlossariesExtraLang: new     | 281      |
| \GLSaccesslong: new .....                            | 126, 129 | \RequireGlossariesExtraLang: new      | 280      |
| \GLSaccesslongpl: new .....                          | 126, 129 | 0.5.4 (2015-12-15)                    |          |
| \Glsaccesslongpl: new .....                          | 126      | \@newglossaryentry@defunitcounters:   |          |
| \glsaccesslongpl: new .....                          | 126      | new .....                             | 88       |
| \GLSaccessname: new .....                            | 119, 127 | \@GLSxtr@p@acrlong@: new .....        | 75       |
| \GLSaccessplural: new .....                          | 121, 127 | \@GLSxtr@p@acrlongpl@: new .....      | 75       |
| \GLSaccessshort: new .....                           | 125, 129 | \@GLSxtr@p@acrshort@: new .....       | 75       |
| \GLSaccessshortpl: new .....                         | 125, 129 | \@GLSxtr@p@acrshortpl@: new .....     | 75       |
| \GLSaccesssymbol: new .....                          | 122, 128 | \@GLSxtr@p@long@: new .....           | 74       |
|  |          | \@GLSxtr@p@longpl@: new .....         | 75       |
|  |          | \@GLSxtr@p@plural@: new .....         | 73       |
|  |          | \@GLSxtr@p@short@: new .....          | 74       |
|  |          | \@GLSxtr@p@shortpl@: new .....        | 74       |
|  |          | \@GLSxtr@p@text@: new .....           | 73       |
|  |          | \@GlsXtrEnableOnTheFly: new .....     | 41       |
|  |          | \@Glsxtr: new .....                   | 41       |
|  |          | \@Glsxtr@p@acrlong@: new .....        | 75       |
|  |          | \@Glsxtr@p@acrlongpl@: new .....      | 75       |



|                                      |    |                                    |     |
|--------------------------------------|----|------------------------------------|-----|
| \@Glsxtr@p@acrshort@: new .....      | 75 | \glsenableentryunitcount: new .... | 90  |
| \@Glsxtr@p@acrshortpl@: new .....    | 75 | \glshasattribute: added check for  |     |
| \@Glsxtr@p@long@: new .....          | 74 | entry's existence .....            | 131 |
| \@Glsxtr@p@longpl@: new .....        | 74 | \glusifattribute: added check for  |     |
| \@Glsxtr@p@plural@: new .....        | 73 | entry's existence .....            | 131 |
| \@Glsxtr@p@short@: new .....         | 73 | \glspostlinkhook: added existence  |     |
| \@Glsxtr@p@shortpl@: new .....       | 74 | check .....                        | 149 |
| \@Glsxtr@p@text@: new .....          | 73 | \Glsxtr: new .....                 | 41  |
| \@Glsxtrpl: new .....                | 42 | \glxtr: new .....                  | 41  |
| \@alt@glshyp@opt: new .....          | 70 | \glxtrcat: new .....               | 41  |
| \@glscalt@hyp@opt: new .....         | 69 | \glxtrdowrglossaryhook: new ....   | 69  |
| \@glscalt@hyp@opt@char: new .....    | 70 | \GlsXtrEnableEntryUnitCounting:    |     |
| \@glscalt@hyp@opt@keys: new .....    | 70 | new .....                          | 93  |
| \@glscincrement@currunitcount:       |    | \GlsXtrEnableOnTheFly: new .....   | 40  |
| new .....                            | 89 | \Glsxtrpl: new .....               | 42  |
| \@glsclocal@increment@currunitcount: |    | \glxtrpl: new .....                | 42  |
| new .....                            | 89 | \glxtrpostlocalreset: new .....    | 81  |
| \@glscsetdefault@glslink@opts:       |    | \glxtrpostlocalunset: new .....    | 81  |
| new .....                            | 67 | \glxtrpostreset: new .....         | 81  |
| \@glsxtr: new .....                  | 41 | \glxtrpostunset: new .....         | 81  |
| \@glsxtr@addunitcounter: new ....    | 88 | \glxtrprotectlinks: new .....      | 72  |
| \@glsxtr@currunitcount: new ....     | 90 | \GlsXtrSetAltModifier: new .....   | 70  |
| \@glsxtr@ifunitcounter: new ....     | 88 | \GlsXtrSetDefaultGlsOpts: new .... | 68  |
| \@glsxtr@p@acrlong@: new .....       | 75 | \glxtrstarflywarn: new .....       | 40  |
| \@glsxtr@p@acrlongpl@: new .....     | 75 | \GlsXtrWarning: new .....          | 42  |
| \@glsxtr@p@acrshort@: new .....      | 75 | \MakeAcronymsAbbreviations: now    |     |
| \@glsxtr@p@acrshortpl@: new .....    | 75 | disables \setacronymstyle .....    | 95  |
| \@glsxtr@p@long@: new .....          | 74 | 1.0 (2016-01-24)                   |     |
| \@glsxtr@p@longpl@: new .....        | 74 | \@glsxtr@autoindexcrossrefs: new . | 13  |
| \@glsxtr@p@plural@: new .....        | 73 | \@glsxtr@idx@displaynumberlist:    |     |
| \@glsxtr@p@short@: new .....         | 73 | new .....                          | 102 |
| \@glsxtr@p@shortpl@: new .....       | 74 | \@glsxtr@idx@entrynumberlist: new  | 103 |
| \@glsxtr@p@text@: new .....          | 73 | \@glsxtr@noidx@displaynumberlist:  |     |
| \@glsxtr@prevunitcount: new ....     | 90 | new .....                          | 102 |
| \@glsxtr@setentryunitcountunsetattr: |    | \@glsxtr@noidx@entrynumberlist:    |     |
| new .....                            | 93 | new .....                          | 103 |
| \@glsxtr@unitcountlist: new ....     | 88 | \@glsxtr@noidx@numberlistloop:     |     |
| \@glsxtrpl: new .....                | 42 | new .....                          | 103 |
| \@newglossaryentryposthook: added    |    | \@glsxtr@reg@glosslist: new .....  | 96  |
| empty see value if not set and added |    | \makeglossaries: new .....         | 97  |
| 'see' to field key map .....         | 33 | 1.01 (2016-02-02)                  |     |
| \@sGlsXtrEnableOnTheFly: new ....    | 40 | \glxtrdiscardperiod: added check   |     |
| \cGlsformat: added .....             | 87 | for first use .....                | 150 |
| \cGlsformat: added .....             | 87 | short-desc: fixed typo in          |     |
| \cGlsplformat: added .....           | 88 | \glxtrinlinefullformat and         |     |
| \cGlsplformat: added .....           | 87 | added missing second argument ...  | 185 |
| \glstdisablehyper: added .....       | 72 | 1.02 (2016-04-25)                  |     |
| \glstdohyperlink: added .....        | 70 | \@glsxtr@current@style: new .....  | 43  |
| \glstdonohyperlink: added .....      | 72 | \Glsfmtfull: new .....             | 279 |

|   |     |   |    |
|---|-----|---|----|
| \glsfmtfull: new .....  | 279 | \@GLSdescplural@: set abbreviation and<br>regular format .....  | 57 |
| \Glsfmtfullpl: new .....  | 280 | \@GLSfirst@: set abbreviation format ..                         | 54 |
| \glsfmtfullpl: new .....  | 280 | \@GLSfirstplural@: set abbreviation<br>and regular format ..... | 55 |
| \Glsfmtlong: new .....  | 278 | \@GLSname@: set abbreviation and regular<br>format .....        | 56 |
| \glsfmtlong: new .....  | 278 | \@GLSplural@: set abbreviation and<br>regular format .....      | 55 |
| \Glsfmtlongpl: new .....  | 279 | \@GLSsymbol@: set regular format .....                          | 57 |
| \glsfmtlongpl: new .....  | 279 | \@GLSsymbolplural@: set regular format                          | 58 |
| \Glsxtrheadfull: new .....  | 274 | \@GLStext@: set abbreviation and regular<br>format .....        | 53 |
| \glsxtrheadfull: new .....  | 273 | \@GLSuseri@: set regular format .....                           | 58 |
| \Glsxtrheadfullpl: new .....  | 274 | \@GLSuserii@: set regular format .....                          | 59 |
| \glsxtrheadfullpl: new .....  | 273 | \@GLSuseriii@: set regular format .....                         | 59 |
| \Glsxtrheadlong: new .....  | 272 | \@GLSuseriv@: set regular format .....                          | 59 |
| \glsxtrheadlong: new .....  | 271 | \@GLSuseriv@: set regular format .....                          | 60 |
| \Glsxtrheadlongpl: new .....  | 272 | \@GLSdesc@: set abbreviation and regular<br>format .....        | 56 |
| \glsxtrheadlongpl: new .....  | 272 | \@GLSdescplural@: set abbreviation and<br>regular format .....  | 57 |
| \Glsxtrtitlefull: new .....   | 274 | \@GLSfirst@: set abbreviation and<br>regular format .....       | 54 |
| \glsxtrtitlefull: new .....   | 273 | \@GLSfirstplural@: set abbreviation<br>and regular format ..... | 55 |
| \Glsxtrtitlefullpl: new .....   | 274 | \@GLSname@: set abbreviation and regular<br>format .....        | 56 |
| \glsxtrtitlefullpl: new .....   | 274 | \@GLSplural@: set abbreviation and<br>regular format .....      | 54 |
| \Glsxtrtitlelong: new .....   | 272 | \@GLsymbol@: set regular format .....                           | 57 |
| \glsxtrtitlelong: new .....   | 271 | \@GLsymbolplural@: set regular format                           | 58 |
| \Glsxtrtitlelongpl: new .....   | 273 | \@GLstext@: set abbreviation and regular<br>format .....        | 53 |
| \glsxtrtitlelongpl: new .....   | 272 | \@GLSuseri@: set regular format .....                           | 58 |
| \ifglxtrinsertinside: new .....   | 175 | \@GLSuserii@: set regular format .....                          | 58 |
| postfootnote: added redef of  |     | \@GLSuseriii@: set regular format .....                         | 59 |
| \glsxtrsetupfulldefs .....  | 182 | \@GLSuseriv@: set regular format .....                          | 59 |
| stylemods: new .....  | 18  | \@GLSuseriv@: set regular format .....                          | 59 |
| 1.03 (2016-04-27)   |     | \@GLSuseriv@: set regular format .....                          | 60 |
| \@GLSfirstplural@: bug fix: misspelt cs<br>name .....                       | 55  | \@GLSdesc@: set abbreviation and regular<br>format .....        | 56 |
| \@GLSplural@: fixed bug \@GLSplural@<br>should be redefined not \@GLSplural | 55  | \@GLSdescplural@: set abbreviation and<br>regular format .....  | 54 |
| \@GLSfirstplural@: bug fix: misspelt cs<br>name .....                       | 55  | \@GLsymbol@: set regular format .....                           | 57 |
| \@GLsplural@: fixed bug \@GLsplural@<br>should be redefined not \@GLsplural | 54  | \@GLsymbolplural@: set regular format                           | 58 |
| \@glsplural@: fixed bug \@glsplural@<br>should be redefined not \@glsplural | 54  | \@GLstext@: set abbreviation and regular<br>format .....        | 53 |
| \glsxtrtitlelongpl: bug fix: changed<br>\glsxtrlong to \glsxtrlongpl ..     | 272 | \@GLSuseri@: set regular format .....                           | 58 |
| \glsxtrtitleshortpl: bug fix: changed<br>\glsxtrshort to \glsxtrshortpl     | 267 | \@GLSuserii@: set regular format .....                          | 58 |
| 1.04 (2015-04-30)   |     | \@GLSuseriii@: set regular format .....                         | 59 |
| short-em-footnote: renamed from<br>“footnote-em” .....                      | 234 | \@GLSuseriv@: set regular format .....                          | 59 |
| 1.04 (2016-05-02)   |     | \@GLSuseriv@: set regular format .....                          | 59 |
| \@glsxtrpostloctag: new .....   | 46  | \@GLSuseriv@: set regular format .....                          | 59 |
| \@GLSdesc@: set abbreviation and regular<br>format .....                    | 56  | \@GLSdesc@: set abbreviation and regular<br>format .....        | 56 |
|   |     | \@GLSdescplural@: set abbreviation and<br>regular format .....  | 57 |
|   |     | \@GLSfirst@: set abbreviation and<br>regular format .....       | 54 |

|  |     |   |     |
|--|-----|---|-----|
| \@glsfirstplural@: set abbreviation<br>and regular format .....          | 55  | \glsxtruserparen: new .....   | 238 |
| \@glsname@: set abbreviation and regular<br>format .....                 | 56  | \glsxtrusersuffix: new .....  | 238 |
| \@glsplural@: set abbreviation and<br>regular format .....               | 54  | \GlsXtrWarnDeprecatedAbbrStyle:<br>new .....                          | 174 |
| \@glsymbol@: set regular format .....                                    | 57  | short-em-long-em: new .....   | 222 |
| \@glsymbolplural@: set regular format .....                              | 58  | short-em-long-em-desc: new .....                                      | 223 |
| \@glstext@: set abbreviation and regular<br>format .....                 | 53  | short-em-nolong: new .....  | 225 |
| \@glsxtr@deprecated@abbrstyle:<br>new .....                              | 174 | short-em-nolong-desc: new .....                                       | 227 |
| \@glsxtr@do@style: new .....   | 19  | short-em-postfootnote: renamed<br>from “postfootnote-em” .....        | 236 |
| \@glsxtr@doloctag: new .....   | 47  | short-footnote: new .....   | 181 |
| \@glsxtr@idx@entrynumberlist:<br>switched from \let to \newcommand ..... | 103 | short-long-user: new .....  | 245 |
| \@glsxtr@pagetag: new .....  | 46  | short-long-user-desc: new .....                                       | 246 |
| \@glsxtr@pagetag: new .....  | 46  | short-nolong: new .....   | 184 |
| \@glsxtr@preloctag: new .....  | 47  | short-nolong-desc: new .....  | 186 |
| \@glsxtr@postloctag: new .....   | 47  | short-postfootnote: new .....   | 183 |
| \@glsxtr@preloctag: new .....  | 46  | short-sc-footnote: renamed from<br>“footnote-sc” .....                | 200 |
| \glossentrydesc: added glossdescfont<br>attribute check .....            | 136 | short-sc-nolong: new .....  | 195 |
| \Glossentryname: added glossnamefont<br>attribute check .....            | 140 | short-sc-nolong-desc: new .....                                       | 196 |
| \glossentryname: added glossnamefont<br>attribute check .....            | 137 | short-sc-postfootnote: renamed<br>from “postfootnote-sc” .....        | 201 |
| moved post name hook inside<br>condition .....                           | 139 | short-sm-footnote: renamed from<br>“footnote-sm” .....                | 213 |
| \glsabbrvmfont: new .....  | 217 | short-sm-nolong: new .....  | 208 |
| \glsabbrvuserfont: new .....   | 238 | short-sm-nolong-desc: new .....                                       | 210 |
| \glsfirstabbrvmfont: new .....   | 217 | short-sm-postfootnote: renamed<br>from “postfootnote-sm” .....        | 215 |
| \glsfirstabbrvuserfont: new .....  | 238 | \letabbreviationstyle: new .....                                      | 174 |
| \glsfirstlongemfont: new .....   | 217 | \newabbreviationstyle: bug fix:<br>corrected test for existence ..... | 172 |
| \glsfirstlonguserfont: new .....   | 238 | long-em-noshort-em: new .....   | 228 |
| \glsifnotregularcategory: new ...  | 132 | long-em-noshort-em-desc: new ...                                      | 232 |
| \glslongdefaultfont: new .....   | 159 | long-em-short-em: new .....   | 219 |
| \glslongemfont: new .....  | 217 | long-em-short-em-desc: new .....                                      | 220 |
| \glslongfont: new .....  | 159 | long-noshort: new .....   | 189 |
| \glslonguserfont: new .....  | 238 | long-noshort-desc: new .....  | 188 |
| \glsxtrassignfieldfont: new .....  | 53  | long-noshort-em: renamed from<br>“long-em” .....                      | 227 |
| \GlsXtrEnablePreLocationTag: new .                                       | 45  | long-noshort-em-desc: renamed from<br>“long-desc-em” .....            | 230 |
| \glsxtrfirstscfont: new .....  | 190 | long-noshort-sc: renamed from<br>“long-sc” .....                      | 196 |
| \glsxtrfirstsmfont: new .....  | 203 | long-noshort-sc-desc: renamed from<br>“long-desc-sc” .....            | 198 |
| \glsxtrlongshortdescsort: new ...  | 176 | long-noshort-sm: renamed from<br>“long-sm” .....                      | 210 |
| \glsxtrpostnamehook: added category<br>check .....                       | 141 | long-noshort-sm-desc: renamed from<br>\long-desc-sm .....             | 212 |
| \glsxtrregularfont: new .....  | 48  |   |     |
| \glsxtruserfield: new .....  | 237 |   |     |

|   |     |                                       |     |
|---|-----|---------------------------------------|-----|
| long-short-user: new .....                | 238 | docdef option changed to choice ..... | 12  |
| long-short-user-desc: new .....           | 244 | \glstr@usesee: new .....              | 34  |
| \renewabbreviationstyle: new ....         | 173 | \glstrusesee: new .....               | 33  |
| style: new .....                          | 19  | \glstruseseeformat: new .....         | 34  |
| 1.05 (2016-06-10)                         |     | \if@glstrdocdefrestricted: new ..     | 13  |
| \eglssetwidest: new .....                 | 289 | 1.07 (2016-08-15)                     |     |
| \glFindWidestAnyName: new .....           | 291 | \@@glstrp: new .....                  | 76  |
| \glFindWidestAnyNameLocation:             |     | \@GLSfirst@: added check for          |     |
| new .....                                 | 296 | nohyperfirst attribute .....          | 54  |
| \glFindWidestAnyNameSymbol: new           | 294 | \@GLSfirstplural@: added check for    |     |
| \glFindWidestAnyNameSymbolLocation:       |     | nohyperfirst attribute .....          | 55  |
| new .....                                 | 295 | \@GLSxtrp: new .....                  | 77  |
| \glFindWidestLevelTwo: new ....           | 292 | \@Glsfirst@: added check for          |     |
| \glFindWidestUsedAnyName: new ..          | 290 | nohyperfirst attribute .....          | 54  |
| \glFindWidestUsedAnyNameLocation:         |     | \@Glsfirstplural@: added check for    |     |
| new .....                                 | 296 | nohyperfirst attribute .....          | 55  |
| \glFindWidestUsedAnyNameSymbol:           |     | \@Glsxtrp: new .....                  | 76  |
| new .....                                 | 293 | \@gls@preglossaryhook: added          |     |
| \glFindWidestUsedAnyNameSymbolLocation:   |     | \glossxtrsetpopts .....               | 149 |
| new .....                                 | 294 | \@glsfirst@: added check for          |     |
| \glFindWidestUsedLevelTwo: new .          | 291 | nohyperfirst attribute .....          | 54  |
| \glFindWidestUsedTopLevelName:            |     | \@glsfirstplural@: added check for    |     |
| new .....                                 | 290 | nohyperfirst attribute .....          | 55  |
| \glsfirstlongfootnotefont: new ..         | 179 | \@glstrinmark: new .....              | 264 |
| \glsgetwidestname: new .....              | 289 | \@glstrnotinmark: new .....           | 264 |
| \glsgetwidestsubname: new .....           | 290 | \@glstrp: new .....                   | 76  |
| \glslongfootnotefont: new .....           | 179 | \@glstrp@opt: new .....               | 75  |
| \glstrAltTreeIndent: new .....            | 289 | \glossxtrsetpopts: new .....          | 76  |
| \glstrAltTreeInit: new .....              | 289 | \glsps: new .....                     | 78  |
| \glstrAltTreePar: new .....               | 289 | \glspt: new .....                     | 78  |
| \glstrAltTreeSetHangIndent: new           | 297 | \glstr@entry@p: new .....             | 77  |
| \glstrAltTreeSetSubHangIndent:            |     | \glstrabbrvfootnote: new .....        | 179 |
| new .....                                 | 297 | \glstrchecknohyperfirst: new ....     | 53  |
| \glstrAltTreeSubSymbolDescLocation:       |     | \glstrfieldtitlecasecs: new ....      | 135 |
| new .....                                 | 289 | \glstrifinmark: new .....             | 264 |
| \glstrAltTreeSymbolDescLocation:          |     | \GLSxtrp: new .....                   | 79  |
| new .....                                 | 288 | \Glsxtrp: new .....                   | 78  |
| \glstrComputeTreeIndent: new ...          | 297 | \glstrp: new .....                    | 77  |
| \glstrComputeTreeSubIndent: new           | 297 | \glstrsetpopts: new .....             | 76  |
| \glstrtreetopindent: new .....            | 289 | short-long-desc: added text key ....  | 179 |
| short-em-long: fixed incorrect font used  |     | fixed misspelling of \glsabbrvfont in |     |
| by long form .....                        | 221 | plural key .....                      | 179 |
| \xglsetwidest: new .....                  | 289 | long-short-desc: added missing text   |     |
| 1.06 (2016-06-18)                         |     | key .....                             | 177 |
| \@glsdoifexistsorwarn: new .....          | 13  | fixed misspelling of \glsabbrvfont .  | 177 |
| \@glstr@docdefval: new .....              | 12  | footnote: changed first forms to use  |     |
| \@glstr@usesee: new .....                 | 34  | \glsfirstlongfootnotefont ...         | 179 |
| General: disabled docdef key at the start |     | postfootnote: removed \footnote       |     |
| of the document .....                     | 21  | from first keys .....                 | 181 |

|   |                                       |
|---|---------------------------------------|
| switched from \glsfirstlongfont to          | 1.10 (2016-12-17)                     |
| \glsfirstlongfootnotefont ... 182           | \@GLSp1@: fixed bug caused by typo in |
| \RestoreAcronyms: modified                  | command name ..... 49                 |
| \@gls@link@checkfirsthyper to               | 1.11 (2017-01-19)                     |
| set \glsxtrifwasfirstuse ..... 96           | \@glsxtr@do@redef@forglsentries:      |
| 1.08 (2016-12-13)                           | new ..... 6                           |
| \@glsxtr@record: new ..... 7                | \@glsxtr@noidx@do: new ..... 117      |
| \@GLS@: added \@glsxtr@record ..... 49      | \@glsxtr@redef@forglsentries: new . 6 |
| \@GLSp1@: added \@glsxtr@record ... 49      | \@glsxtr@shortcutsval: new ..... 16   |
| \@Gls@: added \@glsxtr@record ..... 49      | \@glsxtr@unsrt@getgrouptitle: new 117 |
| \@GLspl@: added \@glsxtr@record ... 49      | \@print@noidx@glossary: added         |
| \@gls@: added \@glsxtr@record ..... 48      | redefinition ..... 105                |
| \@gls@@link@: added                         | \glsxtr@addloclistfield: added        |
| \@glsxtr@record ..... 50                    | group key ..... 11                    |
| \@gls@field@link: added                     | added location key ..... 10           |
| \@glsxtr@record ..... 48                    | \glsxtr@fields: new ..... 113         |
| \@gls@saveentrycounter: new ..... 21        | \glsxtr@linkprefix: new ..... 113     |
| \@glsdisp: added \@glsxtr@record .. 49      | \glsxtr@org@newignoredglossary:       |
| \@glspl@: added \@glsxtr@record ... 49      | new ..... 28                          |
| \@glsxtr@dorecord: new ..... 9              | \glsxtr@s@newignoredglossary: new 29  |
| \@glsxtr@err@undefaction: new ..... 6       | \glsxtr@shortcutsval: new ..... 113   |
| \@glsxtr@record: new ..... 7                | \glsxtr@texencoding: new ..... 113    |
| \@glsxtr@warn@onexistsordo: new ... 6       | \glsxtr@writefields: new ..... 113    |
| \@glsxtr@warn@undefaction: new .... 6       | \GlsXtrLoadResources: new ..... 113   |
| \@print@unsrt@glossary: new ..... 116       | \glsxtrresourcefile: changed          |
| General: added record package option ... 11 | extension to .glstex ..... 112        |
| \glsadd: added \@glsxtr@record .... 52      | \newignoredglossary: added starred    |
| \glsdoifexists: now defines                 | version ..... 28                      |
| \glslabel ..... 32                          | 1.12 (2017-02-03)                     |
| \glsxtr@do@wrglossary: new ..... 21         | \@glsxtr@recordcounter: new ..... 9   |
| \glsxtr@addloclistfield: new ..... 10       | \@gls@preglossaryhook: check for      |
| \glsxtr@indexonly@saveentrycounter:         | definition ..... 148                  |
| new ..... 10                                | \@glsxtr@counterrecordhook: new . 115 |
| \glsxtr@record: new ..... 115               | \@glsxtr@display@loc: new ..... 106   |
| \glsxtr@resource: new ..... 113             | \@glsxtr@docounterrecord: new ... 115 |
| \glsxtr@saveentrycounter: new .... 21       | \@glsxtr@longnewglossaryentry:        |
| \glsxtr@setup@record: new ..... 10          | new ..... 28                          |
| \glsxtrassignfieldfont: added check         | \@glsxtr@noop@recordcounter: new .. 9 |
| for existence ..... 53                      | \@glsxtr@op@recordcounter: new ... 10 |
| \glsxtrresourcefile: new ..... 112          | \@glsxtr@provide@storagekey: new . 22 |
| \printunsrtglossaries: new ..... 116        | \@glsxtr@s@longnewglossaryentry:      |
| \printunsrtglossary: new ..... 115          | new ..... 27                          |
| 1.09 (2016-12-16)                           | \@glsxtrentryfmt: new ..... 23        |
| \@glsxtr@gettype: new ..... 101             | \@glsxtrindexaliased: new ..... 68    |
| \@glsxtr@mixed@assign@sortkey:              | \@glsxtrsetaliasnoindex: new .... 68  |
| new ..... 102                               | \@newglossaryentryposthook: added     |
| \@printglossary: redefined to save          | check for alias key ..... 37          |
| options ..... 101                           | \@no@glsxtrindexaliased: new ..... 68 |
| \glsxtr@makeglossaries: new ..... 101       | \@printunsrtglossary: new ..... 115   |

|  |     |  |     |
|--|-----|--|-----|
| General: added target key to printgloss family .....           | 101 | \GlsXtrLoadResources: removed restriction on only one per document | 113 |
| \apptoglossarypreamble: new .....                              | 27  | \glxtrlocrangefmt: new .....                                       | 107 |
| \csGlsXtrLetField: new .....                                   | 26  | \glxtrpostlongdescription: new ..                                  | 28  |
| \eGlsXtrSetField: new .....                                    | 26  | \glxtrprovidestoragekey: new ....                                  | 22  |
| \gGlsXtrSetField: new .....                                    | 26  | \GlsXtrRecordCounter: new .....                                    | 115 |
| \glsdohyperlink: added check for alias field .....             | 71  | \glxtrresourcecount: new .....                                     | 113 |
| \glsnoidxdisplayloc: added redefinition .....                  | 106 | \glxtrresourcefile: added catcode change for @ .....               | 112 |
| \glissettoctitle: added patch .....                            | 29  | \glxtrsetaliasnoindex: new .....                                   | 67  |
| \glxtr@counterrecord: new .....                                | 115 | \GlsXtrSetField: new .....   | 25  |
| \glxtr@langtag: new .....                                      | 113 | \glxtrsetfieldifexists: new ....                                   | 25  |
| \glxtr@newabbreviation: new ....                               | 155 | \glxtrunsrtdo: new .....   | 117 |
| \glxtr@org@newignoredglossary: Added check for existence ..... | 28  | \Glsxtrusefield: new .....   | 25  |
| \glxtr@pluralsuffixes: new .....                               | 113 | \glxtrusefield: new .....  | 25  |
| \glxtr@provideignoredglossary: new .....                       | 30  | short-postlong-user: new .....                                     | 242 |
| \glxtr@s@newignoredglossary: Added check for existence .....   | 29  | short-postlong-user-desc: new ...                                  | 243 |
| \glxtr@s@provideignoredglossary: new .....                     | 31  | \longnewglossaryentry: added starred version .....                 | 27  |
| \glxtrabbrvpluralsuffix: new ...                               | 159 | long-postshort-user: new .....                                     | 240 |
| \glxtralias: new .....   | 37  | long-postshort-user-desc: new ...                                  | 241 |
| \glxtrcopytoglossary: new .....                                | 31  | postdot: new .....   | 14  |
| \glxtrdeffield: new .....                                      | 25  | \pretoglossarypreamble: new .....                                  | 27  |
| \glxtrdisplayendloc: new .....                                 | 106 | \print@noop@unsrtglossaryunit: new .....                           | 117 |
| \glxtrdisplayendlochook: new ...                               | 107 | \print@op@unsrtglossaryunit: new                                   | 117 |
| \glxtrdisplaysingleloc: new ....                               | 106 | \printunsrtglossary: added starred form .....                      | 115 |
| \glxtrdisplaystartloc: new ....                                | 106 | \printunsrtglossaryhandler: new .                                  | 116 |
| \glxtrdeffield: new .....                                      | 25  | \printunsrtglossaryunit: new ....                                  | 10  |
| \glxtrentryfmt: new .....                                      | 23  | \printunsrtglossaryunitsetup: new                                  | 117 |
| \glxtrfielddolistloop: new .....                               | 24  | \provideignoredglossary: new ....                                  | 30  |
| \glxtrfieldforlistloop: new ....                               | 24  | \s@glxtr@provide@storagekey: new                                   | 22  |
| \glxtrfielddifylist: new .....                                 | 24  | \s@printunsrtglossary: new .....                                   | 115 |
| \glxtrfieldlistadd: new .....                                  | 24  | \xGlsXtrSetField: new .....  | 26  |
| \glxtrfieldlistead: new .....                                  | 24  | 1.13 (2017-02-07)  |     |
| \glxtrfieldlistgadd: new .....                                 | 24  | \@glsdisp: removed   |     |
| \glxtrfieldlistxadd: new .....                                 | 24  | \@glxtr@org@glsdisp .....  | 49  |
| \glxtrfieldxifylist: new .....                                 | 24  | \glxtrsetaliasnoindex: switched to \providecommand .....           | 67  |
| \glxtrfmt: new .....   | 23  | 1.14 (2017-04-18)  |     |
| \GlsXtrFmtDefaultOptions: new ....                             | 23  | \@gls@link: added redefinition .....                               | 51  |
| \GlsXtrFmtField: new .....                                     | 23  | \@gls@noidx@getgrouptitle: new ..                                  | 103 |
| \glxtrifkeydefined: new .....                                  | 21  | \@gls@removespaces: new .....                                      | 107 |
| \glxtrindexaliased: new .....                                  | 68  | \@glxtr@do@automake@err: new ...                                   | 114 |
| \GlsXtrLetField: new .....                                     | 26  | \@glxtr@org@gloautosee: new ....                                   | 19  |
| \GlsXtrLetFieldToField: new .....                              | 26  | \@glxtr@record: added third arg ....                               | 7   |
|  |     | \@glxtr@recordsee: new .....                                       | 10  |

|  |     |  |     |
|--|-----|--|-----|
| General: added \glsadd option          |     | 1.16 (2017-06-15)                      |     |
| theHvalue .....                        | 52  | \@glo@autosee: added redefinition .... | 20  |
| added \glsadd option thevalue ....     | 52  | \@gls@noidx@getgrouptitle: fixed       |     |
| \glsdisablehyper: added redefinition . | 71  | bug .....                              | 103 |
| \glsenableentrycount: fixed            |     | \@glsxtr@addunusedxrefs: added         |     |
| assignment of \@cGls@ .....            | 83  | check for seealso field .....          | 38  |
| \glsenableentryunitcount: fixed        |     | \@glsxtr@dorecordnodefer: new .....    | 9   |
| assignment of \@cGls@ .....            | 91  | \@glsxtr@noidx@do: use \csuse instead  |     |
| \glsnavigation: new .....              | 104 | of \csname .....                       | 118 |
| \glsxtr@org@getgrouptitle: new ..      | 104 | \@print@unsrt@glossary: corrected      |     |
| \glsxtr@recordsee: new .....           | 7   | misspelt command .....                 | 116 |
| \glsxtr@writefields: added check for   |     | \@printunsrt@glossary@handler:         |     |
| automake .....                         | 114 | new .....                              | 116 |
| \glsxtrdisplayendloc: added check      |     | General: added check for               |     |
| for empty format .....                 | 106 | \@gls@setupsort@none .....             | 12  |
| \glsxtrgetgrouptitle: new .....        | 104 | \@gls@checkseeallowed: added           |     |
| \glsxtrinitwrgloss: new .....          | 50  | redefinition .....                     | 20  |
| \glsxtrlocationhyperlink: new ...      | 107 | \@glsxtr@writefields: added            |     |
| \glsxtrsetgrouptitle: new .....        | 104 | \providecommand lines .....            | 113 |
| \glsxtrsupphypernumber: new .....      | 107 | \@glsxtrautoindex: new .....           | 142 |
| \ifglsxtrwrglossbefore: new .....      | 50  | \@glsxtrautoindexentry: new .....      | 142 |
| 1.15 (2017-05-10)                      |     | \@glsxtrautoindexsort: new .....       | 143 |
| \@glsxtr@dorecord: corrected           |     | \@glsxtrindexseealso: new .....        | 35  |
| premature expansion of \@glslocref     | 9   | \@glsxtrseealsolabels: new .....       | 37  |
| short-em-long-em: fixed spelling of    |     | \@glsxtrseelist: new .....             | 34  |
| \glsabbrvfont .....                    | 222 | \@glsxtruseseealso: new .....          | 34  |
| short-long: fixed spelling of          |     | \@glsxtruseseealsoformat: new .....    | 34  |
| \glsabbrvfont .....                    | 177 | \@seealso: new .....                   | 34  |
| short-long-user: fixed spelling of     |     | autoseeindex: new .....                | 13  |
| \glsabbrvfont .....                    | 245 | 1.17 (2017-08-09)                      |     |
| short-postlong-user: fixed spelling of |     | \@glsxtr@mark@wordseps: new .....      | 154 |
| \glsabbrvfont .....                    | 242 | \@glsxtr@markwordseps: new .....       | 154 |
| short-postlong-user-desc: fixed        |     | \@glsxtr@noidx@displaynumberlist:      |     |
| spelling of \glsabbrvfont .....        | 244 | replace hard-coded ?? with             |     |
| long-em-short-em: fixed spelling of    |     | \glsxtrundeftag .....                  | 102 |
| \glsabbrvfont .....                    | 219 | \@glsxtr@noidx@entrynumberlist:        |     |
| long-postshort-user: fixed spelling of |     | replace hard-coded ?? with             |     |
| \glsabbrvfont .....                    | 240 | \glsxtrundeftag .....                  | 103 |
| long-postshort-user-desc: fixed        |     | \@glsxtr@noidx@numberlistloop:         |     |
| spelling of \glsabbrvfont .....        | 241 | replace hard-coded ?? with             |     |
| long-short: fixed spelling of          |     | \glsxtrundeftag .....                  | 103 |
| \glsabbrvfont .....                    | 175 | \@glsxtrifhyphenstart: new .....       | 247 |
| long-short-user: fixed spelling of     |     | General: removed some inconsistencies  |     |
| \glsabbrvfont .....                    | 239 | in the abbreviation styles .....       | 175 |
| footnote: fixed spelling of            |     | \glsabbrvhyphenfont: new .....         | 247 |
| \glsabbrvfont .....                    | 180 | \glsabbrvonlyfont: new .....           | 261 |
| postfootnote: fixed spelling of        |     | \glsabbrvscfont: new .....             | 190 |
| \glsabbrvfont .....                    | 181 | \glsabbrvsmfont: new .....             | 203 |

|   |     |  |     |
|---|-----|--|-----|
| \glsabbrvuserfont: initialised to<br>default font .....                       | 238 | short-long-user-desc: corrected first<br>forms .....   | 246 |
| \glsfirstabbrvhyphenfont: new ...   | 247 | short-nolong-desc-noreg: new ....  | 186 |
| \glsfirstabbrvonlyfont: new .....   | 261 | short-nolong-noreg: new .....  | 184 |
| \glsfirstabbrvscfont: new .....   | 190 | long-em-noshort-em-desc-noreg:<br>new .....  | 233 |
| \glsfirstabbrvsmfont: new .....   | 203 | long-em-noshort-em-noreg: new ...  | 230 |
| \glsfirstlonghyphenfont: new ....   | 248 | long-hyphen-noshort-desc-noreg:<br>new .....   | 250 |
| \glsfirstlongonlyfont: new .....  | 261 | long-hyphen-postshort-hyphen: new  | 253 |
| \glslonghyphenfont: new .....   | 248 | long-hyphen-postshort-hyphen-desc:<br>new .....  | 255 |
| \glslongonlyfont: new .....   | 261 | long-hyphen-short-hyphen: new ...  | 248 |
| \glslonguserfont: initialised to default<br>font .....                        | 238 | long-hyphen-short-hyphen-desc:<br>new .....  | 249 |
| \glsxtr@newabbreviation: added<br>\glsxtrorgshort and<br>\glsxtrorglong ..... | 155 | long-noshort-desc-noreg: new ....  | 188 |
| \GlsXtrDefineAcShortcuts: new ....  | 15  | long-noshort-noreg: new .....  | 189 |
| \glsxtrgenabbrvfmt: added check for<br>\ifglsxtrininsertinside .....          | 169 | long-only-short-only: new .....  | 261 |
| \glsxtrhyphensuffix: new .....  | 248 | long-only-short-only-desc: new ..  | 263 |
| \glsxtrifhyphenstart: new .....   | 247 | long-short-user-desc: corrected first<br>forms .....   | 244 |
| \glsxtrlonghyphen: new .....  | 252 | 1.18 (2017-08-10)<br>stylemods: changed default value to<br>“default” .....                          | 18  |
| \glsxtrlonghyphennoshort: new ...   | 249 | 1.19 (2017-09-09)<br>\@glsxtr@defaultnumberformat: new .   | 7   |
| \glsxtrlonghyphenshort: new ....  | 247 | \@glsxtr@dorecord: Use<br>\@glsrecordlocref instead of<br>\@glslocref .....                          | 9   |
| \glsxtrlongshortdescname: new ...   | 176 | \@glsxtr@dorecordnodefer: Use<br>\theglsentrycounter for the<br>location rather than \@glslocref ... | 9   |
| \glsxtronlydescname: new .....  | 262 | \@glsxtr@record@setting: new ....  | 11  |
| \glsxtronlydescsort: new .....  | 262 | \@glsxtr@record@setting@alsoindex:<br>new .....  | 11  |
| \glsxtronlysuffix: new .....  | 261 | \@glsxtrifhasfield: new .....  | 25  |
| \glsxtrparen: new .....   | 157 | General: added \glslink option<br>theHvalue .....  | 51  |
| \glsxtrposthyphenlong: new .....  | 258 | added \glslink option thevalue ...   | 50  |
| \glsxtrposthyphenshort: new ....  | 252 | \glsxtr@writefields: removed<br>double-quotes around \jobname ..                                     | 114 |
| \glsxtrposthyphensubsequent: new  | 253 | \glsxtrdoautoindexname: changed<br>format test .....   | 142 |
| \glsxtrshortdescname: new .....   | 185 | \glsxtrhyperlink: new .....  | 71  |
| \glsxtrshorthyphen: new .....   | 257 | \glsxtrifhasfield: new .....   | 25  |
| \glsxtrshorthyphenlong: new ....  | 255 | \GlsXtrSetDefaultNumberFormat:<br>new .....  | 7   |
| \glsxtrshortlongdescname: new ...   | 178 | \s@glsxtrifhasfield: new .....   | 25  |
| \glsxtrshortlongdescsort: new ...   | 178 |  |     |
| \Glsxtrsubsequentfmt: new .....   | 171 |  |     |
| \glsxtrsubsequentfmt: new .....   | 171 |  |     |
| \Glsxtrsubsequentplfmt: new ....  | 171 |  |     |
| \glsxtrsubsequentplfmt: new ....  | 171 |  |     |
| \glsxtrword: new .....  | 154 |  |     |
| \glsxtrwordsep: new .....   | 154 |  |     |
| short-hyphen-long-hyphen: new ...   | 256 |  |     |
| short-hyphen-long-hyphen-desc:<br>new .....                                   | 257 |  |     |
| short-hyphen-postlong-hyphen: new   | 258 |  |     |
| short-hyphen-postlong-hyphen-desc:<br>new .....                               | 260 |  |     |



# 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                            |                    |
|------------------------------------|--------------------|
| \.                                 | 150, 288           |
| \@                                 | 112                |
| \@cGLS@                            | 83, 92             |
| \@cGLSpl@                          | 83, 92             |
| \@cGLs@                            | 83, 91             |
| \@cGLspl@                          | 83, 92             |
| \@cgls@                            | 83, 91             |
| \@cglspl@                          | 83, 85, 91         |
| \@do@wrglossary                    | 8, 98              |
| \@do@wrglossary                    | 11, 12, 21, 52, 68 |
| \@glo@assign@sortkey               | 102                |
| \@glo@list                         | 6                  |
| \@glo@type                         | 116                |
| \@gls@expand@field                 | 22                 |
| \@glslocalreset                    | 81                 |
| \@glslocalunset                    | 81                 |
| \@glsreset                         | 81                 |
| \@glsunset                         | 81                 |
| \@glsxtr@autoindex@escspch         | 144, 145           |
| \@glsxtr@checkspch                 | 143–146            |
| \@glsxtr@disabledflycommand        | 43                 |
| \@glsxtr@record                    | 12                 |
| \@glsxtr@recordcounter             | 11, 12, 115        |
| \@glsxtrp                          | 76, 77             |
| \@glsxtrpostloctag                 | 45                 |
| \@glsxtrpreloctag                  | 45, 46             |
| \@newglossaryentry@defcounters     | 82                 |
| \@newglossaryentry@defunitcounters | 90                 |
| \@par                              | 289                |
| \@ACRlong                          | 73                 |
| \@ACRlongpl                        | 73                 |
| \@ACRshort                         | 73                 |
| \@ACRshortpl                       | 73                 |
| \@Acrlong                          | 73                 |
| \@Acrlongpl                        | 73                 |
| \@Acrshort                         | 73                 |
| \@Acrshortpl                       | 73                 |
| \@GLS@                             | 72, 86, 87         |
| \@GLSdesc@                         | 57                 |
| \@GLSpl@                           | 72, 86, 87         |
| \@GLSplural@                       | 73                 |
| \@GLSsymbol@                       | 58                 |
| \@GLStext@                         | 73                 |
| \@GLSxtr@full                      | 160                |
| \@GLSxtr@fullpl                    | 162                |
| \@GLSxtr@p@acrlong@                | 73                 |
| \@GLSxtr@p@acrlongpl@              | 73                 |
| \@GLSxtr@p@acrshort@               | 73                 |
| \@GLSxtr@p@acrshortpl@             | 73                 |
| \@GLSxtr@p@long@                   | 72                 |
| \@GLSxtr@p@longpl@                 | 72                 |
| \@GLSxtr@p@plural@                 | 72                 |
| \@GLSxtr@p@short@                  | 72                 |
| \@GLSxtr@p@shortpl@                | 72                 |
| \@GLSxtr@p@text@                   | 72                 |
| \@GLSxtrlong                       | 72, 165            |
| \@GLSxtrlongpl                     | 72, 168            |
| \@GLSxtrp                          | 80                 |
| \@GLSxtrshort                      | 72, 163            |
| \@GLSxtrshortpl                    | 72, 167            |
| \@Gls@                             | 72, 85, 86         |
| \@Gls@acentryname                  | 94                 |
| \@Gls@entry@field                  | 65, 79             |
| \@Gls@entryname                    | 94                 |
| \@GlsXtrEnableOnTheFly             | 40                 |
| \@Glspl@                           | 72, 85, 86         |
| \@Glsplural@                       | 73                 |
| \@Glstext@                         | 73                 |
| \@Glsxtr                           | 41, 43             |
| \@Glsxtr@full                      | 160                |
| \@Glsxtr@fullpl                    | 161                |
| \@Glsxtr@p@acrlong@                | 73                 |
| \@Glsxtr@p@acrlongpl@              | 73                 |

|                          |   |                          |  |
|--------------------------|---|--------------------------|--|
| \@Glsxtr@p@acrshort@     | 73  | \@glo@assign@sortkey     | 99   |
| \@Glsxtr@p@acrshortpl@   | 73  | \@glo@autosee            | 19   |
| \@Glsxtr@p@long@         | 72  | \@glo@autoseehook        | 36   |
| \@Glsxtr@p@longpl@       | 72  | \@glo@category           | 88   |
| \@Glsxtr@p@plural@       | 72  | \@glo@check@sortallowed  | 100  |
| \@Glsxtr@p@short@        | 72  | \@glo@counterprefix      | 9, 107                                       |
| \@Glsxtr@p@shortpl@      | 72  | \@glo@countunit          | 88   |
| \@Glsxtr@p@text@         | 72  | \@glo@default@sorttype   | 99   |
| \@Glsxtrlong             | 72, 165   | \@glo@desc               | 28   |
| \@Glsxtrlongpl           | 72, 168   | \@glo@descplural         | 28   |
| \@Glsxtrp                | 79  | \@glo@group              | 11   |
| \@Glsxtrpl               | 42, 43  | \@glo@label              | 10, 11, 22, 33, 36–38, 64, 71, 290–297       |
| \@Glsxtrshort            | 72, 163   | \@glo@location           | 10, 11                                       |
| \@Glsxtrshortpl          | 72, 166   | \@glo@loclist            | 10   |
| \@acrlong                | 73  | \@glo@name               | 142  |
| \@acrlongpl              | 73  | \@glo@no@assign@sortkey  | 102  |
| \@acrshort               | 73  | \@glo@parent             | 291–293                                      |
| \@acrshortpl             | 73  | \@glo@see                | 33, 34, 37, 38                               |
| \@alt@glshyp@opt         | 70  | \@glo@seealso            | 36   |
| \@auxout                 | 9, 10, 47, 84, 92, 97, 98, 108, 112–115                                 | \@glo@sort               | 142  |
| \@bibgls@restoreat       | 112   | \@glo@sorttype           | 99, 105                                      |
| \@cGLS                   | 86  | \@glo@text               | 49   |
| \@cGLS@                  | 83, 86, 92  | \@glo@thislettergrp      | 118  |
| \@cGLSpl                 | 87  | \@glo@thisvalue          | 238  |
| \@cGLSpl@                | 83, 87, 92  | \@glo@tmp                | 22, 34, 64                                   |
| \@cGls@                  | 83, 91  | \@glo@type               | 38, 94, 97, 100–102, 105, 108, 111, 112, 116 |
| \@cGlspl@                | 83, 92  | \@glo@types              | 133, 290–296                                 |
| \@cgls@                  | 83, 91  | \@glossary@default@style | 43, 44, 100, 299                             |
| \@cglspl@                | 83, 91  | \@glossarystyle          | 100, 101                                     |
| \@disable@onlypremakeg   | 97  | \@gls@                   | 72, 85, 86                                   |
| \@do@auxoutstuff         | 108   | \@gls@@link              | 50   |
| \@do@glsgetcounterprefix | 9   | \@gls@ReturnAfterFi      | 107  |
| \@do@glsee               | 36, 37  | \@gls@actualchar         | 143  |
| \@do@newglossaryentry    | 94, 95, 157   | \@gls@adjustmode         | 52   |
| \@do@seeglossary         | 11, 12, 19, 97  | \@gls@alt@hyp@opt        | 70   |
| \@do@wrglossary          | 51, 52  | \@gls@alt@hyp@opt@char   | 70   |
| \@empty                  | 52, 60–64, 143, 144, 159–169  | \@gls@alt@hyp@opt@keys   | 70   |
| \@end@glxtr@addunused    | 38  | \@gls@automake           | 100  |
| \@end@glxtr@gettype      | 99, 101   | \@gls@between            | 105  |
| \@end@glxtr@usesee       | 33, 34  | \@gls@checkedmkidx       | 143–146                                      |
| \@end@glxtrifhyphenstart | 247   | \@gls@checkedmkidxchars  | 35, 142                                      |
| \@endfortrue             | 172   | \@gls@codepage           | 108  |
| \@firstofone             | 53, 136, 137, 141, 147  | \@gls@counter            | 8, 9, 51, 52, 68                             |
| \@firstofthree           | 49, 52, 60–63, 69, 70, 159, 161, 163, 164, 166, 167                     | \@gls@currentlettergroup | 105, 116, 118                                |
| \@firstoftwo             | 54, 55, 57, 58, 61–64, 66, 70, 96, 151, 152, 160–162, 166–168, 264, 265 | \@gls@declareoption      | 5  |
| \@for                    | 6, 18, 38, 82, 94, 97, 100, 105, 116, 135, 147                          | \@gls@default@longpl     | 155, 156                                     |
| \@glo@alias              | 36, 37  | \@gls@doautomake         | 100, 114                                     |

|  |                             |  |                            |
|--|-----------------------------|--|----------------------------|
| <code>\@gls@doautomake@err</code> .....                | 114                         | <code>\@gls@setupsort@none</code> .....            | 12                         |
| <code>\@gls@encapchar</code> .....                     | 143                         | <code>\@gls@short</code> .....                     | 156                        |
| <code>\@gls@entry@count</code> .....                   | 83, 84                      | <code>\@gls@shortpl</code> .....                   | 153, 156, 157              |
| <code>\@gls@entry@field</code> ..                      | 22, 25, 36, 64, 77–80, 83   | <code>\@gls@sort</code> .....                      | 118                        |
| <code>\@gls@entry@unitcount</code> .....               | 92                          | <code>\@gls@tmp</code> .....                       | 105                        |
| <code>\@gls@field@font</code> .....                    | 53–60                       | <code>\@gls@tmpb</code> .....                      | 145, 146                   |
| <code>\@gls@field@link</code> .....                    | 53–60, 65, 66               | <code>\@gls@type</code> .....                      | 97, 98, 100, 172, 290–296  |
| <code>\@gls@getcounterprefix</code> .....              | 9                           | <code>\@gls@write@entrycounts</code> .....         | 83                         |
| <code>\@gls@getgrouptitle</code> .....                 | 104, 116                    | <code>\@gls@write@entryunitcounts</code> .....     | 92                         |
| <code>\@gls@grptitle</code> .....                      | 105                         | <code>\@gls@write@entryunitcounts@do</code> .....  | 93                         |
| <code>\@gls@hyp@opt</code> ....                        | 65, 66, 70, 86, 87, 159–168 | <code>\@gls@xref</code> .....                      | 10, 35                     |
| <code>\@gls@hyp@opt@cs</code> .....                    | 69, 70                      | <code>\@glsabbrv@current@abbreviation</code> ..... | 155, 169                   |
| <code>\@gls@increment@currcount</code> .....           | 83                          | <code>\@glsacronymlists</code> .....               | 94                         |
| <code>\@gls@increment@currunitcount</code> .....       | 91                          | <code>\@glsdoifexistsorwarn</code> ...             | 13, 137, 138, 140          |
| <code>\@gls@keymap</code> .....                        | 10, 11, 22, 33, 36, 64, 113 | <code>\@glsentry</code> .....                      | 84, 92, 93                 |
| <code>\@gls@label</code> .....                         | 8, 9, 69, 97, 98, 115, 172  | <code>\@glslink</code> .....                       | 51, 71, 72                 |
| <code>\@gls@levelchar</code> .....                     | 143                         | <code>\@glsnextpages</code> .....                  | 101                        |
| <code>\@gls@link</code> .....                          | 23, 48–50, 60–64, 160–169   | <code>\@glsnonextpages</code> .....                | 101                        |
| <code>\@gls@link@checkfirsthyper</code> .....          | 49, 96                      | <code>\@glsnumberformat</code> ..                  | 7, 9, 51, 52, 68, 141, 142 |
| <code>\@gls@link@label</code> .....                    | 51                          | <code>\@glsorder</code> .....                      | 97                         |
| <code>\@gls@link@nocheckfirsthyper</code> .....        | 48, 60–64, 159–168          | <code>\@glspl@</code> .....                        | 72, 85, 86                 |
| <code>\@gls@link@opts</code> .....                     | 51                          | <code>\@glsplural@</code> .....                    | 73                         |
| <code>\@gls@list</code> .....                          | 105                         | <code>\@glspl@token</code> .....                   | 152                        |
| <code>\@gls@local@increment@currcount</code> ....      | 83                          | <code>\@glsrecordlocref</code> .....               | 9                          |
| <code>\@gls@local@increment@currunitcount</code> ..... | 91                          | <code>\@glsshowtarget</code> .....                 | 71                         |
| <code>\@gls@location</code> .....                      | 118, 119                    | <code>\@glsstyle@alttree</code> .....              | 288                        |
| <code>\@gls@loclist</code> .....                       | 102, 103, 118, 119          | <code>\@glsstyle@inline</code> .....               | 288                        |
| <code>\@gls@long</code> .....                          | 155                         | <code>\@glsstyle@listdotted</code> .....           | 283                        |
| <code>\@gls@longpl</code> .....                        | 153, 155–157                | <code>\@gls@target</code> .....                    | 72, 101                    |
| <code>\@gls@nohyperlist</code> .....                   | 29–31                       | <code>\@gls@text@</code> .....                     | 73                         |
| <code>\@gls@noidx@do</code> .....                      | 105                         | <code>\@glswidestname</code> .....                 | 289, 290, 297              |
| <code>\@gls@noidx@getgrouptitle</code> .....           | 116                         | <code>\@glsxtr</code> .....                        | 41, 43                     |
| <code>\@gls@noidx@nosanitizesort</code> .....          | 99                          | <code>\@glsxtr@@do@@wrglossary</code> .....        | 98                         |
| <code>\@gls@noidx@sanitizesort</code> .....            | 99                          | <code>\@glsxtr@abbreviationsdef</code> .....       | 14, 20                     |
| <code>\@gls@noidx@loclist@finalsep</code> .....        | 102                         | <code>\@glsxtr@activate@initialtagging</code> ..   | 147, 148                   |
| <code>\@gls@noidx@loclist@prev</code> .....            | 102                         | <code>\@glsxtr@addunitcounter</code> .....         | 88                         |
| <code>\@gls@noidx@loclist@sep</code> .....             | 102                         | <code>\@glsxtr@addunused</code> .....              | 38                         |
| <code>\@gls@noref@warn</code> .....                    | 98, 105                     | <code>\@glsxtr@addunusedxrefs</code> .....         | 38                         |
| <code>\@gls@org@glsnoidx@displayloc</code> .....       | 103                         | <code>\@glsxtr@attrval</code> ...                  | 136, 137, 139, 140, 142    |
| <code>\@gls@org@glssee@format</code> .....             | 103                         | <code>\@glsxtr@autoindex@at</code> .....           | 142–144                    |
| <code>\@gls@preglossaryhook</code> .....               | 101, 147                    | <code>\@glsxtr@autoindex@doextra@esc</code> ....   | 142                        |
| <code>\@gls@prevlevel</code> .....                     | 298                         | <code>\@glsxtr@autoindex@encap</code> .....        | 142–144                    |
| <code>\@gls@quotechar</code> .....                     | 143                         | <code>\@glsxtr@autoindex@esc</code> .....          | 143, 145, 146              |
| <code>\@gls@reference</code> .....                     | 39, 97, 98                  | <code>\@glsxtr@autoindex@escat</code> .....        | 143, 144                   |
| <code>\@gls@saveentrycounter</code> ..                 | 11, 12, 21, 51, 52          | <code>\@glsxtr@autoindex@escencap</code> ...       | 143, 144                   |
| <code>\@gls@see@noindex</code> .....                   | 20, 112                     | <code>\@glsxtr@autoindex@esclevel</code> ...       | 143, 144                   |
| <code>\@gls@setdefault@glslink@opts</code> ...         | 51, 68                      | <code>\@glsxtr@autoindex@escquote</code> ...       | 143, 145                   |
| <code>\@gls@setsort</code> .....                       | 51                          | <code>\@glsxtr@autoindex@level</code> .....        | 143, 144                   |

|   |                         |   |          |
|---|-------------------------|---|----------|
| <code>\@glsxtr@autoindex@setname</code> .....             | 142                     | <code>\@glsxtr@format@overridefalse</code> .....        | 141      |
| <code>\@glsxtr@autoindexcrossrefs</code> .....            | 12, 13, 33, 36          | <code>\@glsxtr@format@overridetrue</code> ..            | 141, 142 |
| <code>\@glsxtr@autoseeindexfalse</code> .....             | 12                      | <code>\@glsxtr@foundinlist</code> .....                 | 152      |
| <code>\@glsxtr@autoseeindextrue</code> .....              | 13                      | <code>\@glsxtr@full</code> .....                        | 159      |
| <code>\@glsxtr@cat</code> .....                           | 82, 94, 147             | <code>\@glsxtr@fullpl</code> .....                      | 161      |
| <code>\@glsxtr@counterrecordhook</code> .....             | 9, 10                   | <code>\@glsxtr@gettype</code> .....                     | 99       |
| <code>\@glsxtr@csname</code> .....                        | 89, 91                  | <code>\@glsxtr@glossdescfont</code> .....               | 136, 137 |
| <code>\@glsxtr@current@style</code> .....                 | 43, 299                 | <code>\@glsxtr@glossnamefont</code> .....               | 137–141  |
| <code>\@glsxtr@currentunitcount</code> .....              | 89, 91                  | <code>\@glsxtr@gobbleto@endescspch</code> .....         | 145      |
| <code>\@glsxtr@currunitcount</code> .....                 | 90, 92                  | <code>\@glsxtr@idx@displaynumberlist</code> .....       | 98       |
| <code>\@glsxtr@declareoption</code> .....                 | 5, 14, 18               | <code>\@glsxtr@idx@entrynumberlist</code> .....         | 99       |
| <code>\@glsxtr@defaultnoglossarywarning</code> ..         | 18                      | <code>\@glsxtr@ifcsstart</code> .....                   | 40       |
| <code>\@glsxtr@defaultnumberformat</code> .....           | 7, 51, 52, 68, 141, 142 | <code>\@glsxtr@ifpunctoken</code> .....                 | 152      |
| <code>\@glsxtr@deprecated@abbrstyle</code> .....          | 198, 200, 201,          | <code>\@glsxtr@ifunitcounter</code> .....               | 88       |
| 203, 212, 213, 215, 217, 228, 232, 236, 237               |                         | <code>\@glsxtr@insert@dots</code> .....                 | 154      |
| <code>\@glsxtr@disabledflycommand</code> .....            | 43                      | <code>\@glsxtr@insert@dots@next</code> .....            | 154      |
| <code>\@glsxtr@display@loc</code> .....                   | 106                     | <code>\@glsxtr@insertdots</code> .....                  | 156      |
| <code>\@glsxtr@do@w@rindex</code> .....                   | 69                      | <code>\@glsxtr@label</code> .....                       | 38, 135  |
| <code>\@glsxtr@do@gl@sd@is@ab@le@hyper@in@list</code> ..  | 67                      | <code>\@glsxtr@loadstyles</code> .....                  | 282      |
| <code>\@glsxtr@do@re@def@for@gl@s@en@tries</code> .....   | 7                       | <code>\@glsxtr@longnewglossaryentry</code> .....        | 27       |
| <code>\@glsxtr@do@style</code> .....                      | 19, 281                 | <code>\@glsxtr@mark@wordseps</code> .....               | 154      |
| <code>\@glsxtr@do@titlecaps@warn</code> .....             | 136–139, 147, 148       | <code>\@glsxtr@mark@wordseps@next</code> .....          | 155      |
| <code>\@glsxtr@do@abbreviations@def</code> .....          | 14                      | <code>\@glsxtr@markwordseps</code> .....                | 155, 156 |
| <code>\@glsxtr@do@acc@supp</code> .....                   | 18, 19                  | <code>\@glsxtr@mixed@assign@sortkey</code> .....        | 99       |
| <code>\@glsxtr@doc@def@val</code> .....                   | 12, 13, 39              | <code>\@glsxtr@noidx@displaynumberlist</code> ...       | 98       |
| <code>\@glsxtr@doc@counter@record</code> .....            | 10                      | <code>\@glsxtr@noidx@do</code> .....                    | 117      |
| <code>\@glsxtr@doglossary</code> .....                    | 116                     | <code>\@glsxtr@noidx@entrynumberlist</code> .....       | 99       |
| <code>\@glsxtr@doloctag</code> .....                      | 46                      | <code>\@glsxtr@noidx@numberlistloop</code> .....        | 99       |
| <code>\@glsxtr@dorecord</code> .....                      | 8                       | <code>\@glsxtr@noop@recordcounter</code> .....          | 9, 11    |
| <code>\@glsxtr@dorecord@nodefer</code> .....              | 8                       | <code>\@glsxtr@notfoundinlist</code> .....              | 152      |
| <code>\@glsxtr@dostylewarn</code> .....                   | 172                     | <code>\@glsxtr@op@recordcounter</code> .....            | 12       |
| <code>\@glsxtr@enabletagging</code> .....                 | 146                     | <code>\@glsxtr@optlist</code> .....                     | 42       |
| <code>\@glsxtr@end@</code> .....                          | 40                      | <code>\@glsxtr@org@GLS@</code> .....                    | 49       |
| <code>\@glsxtr@endescspch</code> .....                    | 143–146                 | <code>\@glsxtr@org@GLSpl@</code> .....                  | 49       |
| <code>\@glsxtr@entrycount@org@localreset</code> ..        | 83                      | <code>\@glsxtr@org@Gls@</code> .....                    | 49       |
| <code>\@glsxtr@entrycount@org@localunset</code> ..        | 83                      | <code>\@glsxtr@org@Glspl@</code> .....                  | 49       |
| <code>\@glsxtr@entrycount@org@reset</code> .....          | 83                      | <code>\@glsxtr@org@Glsxtr@titlefirst</code> ..          | 265, 266 |
| <code>\@glsxtr@entrycount@org@unset</code> .....          | 83                      | <code>\@glsxtr@org@Glsxtr@titlefirstplural</code> ..... | 265, 266 |
| <code>\@glsxtr@entryunitcount@org@localreset</code> ..... | 91                      | <code>\@glsxtr@org@Glsxtr@titlefull</code> ..           | 265, 266 |
| <code>\@glsxtr@entryunitcount@org@localunset</code> ..... | 91                      | <code>\@glsxtr@org@Glsxtr@titlefullpl</code> ..         | 265, 266 |
| <code>\@glsxtr@entryunitcount@org@reset</code> ..         | 91                      | <code>\@glsxtr@org@Glsxtr@titlelong</code> ..           | 265, 266 |
| <code>\@glsxtr@entryunitcount@org@unset</code> ..         | 91                      | <code>\@glsxtr@org@Glsxtr@titlelongpl</code> ..         | 265, 266 |
| <code>\@glsxtr@err@undefaction</code> .....               | 7, 11                   | <code>\@glsxtr@org@Glsxtr@titleplural</code> ..         | 265, 266 |
| <code>\@glsxtr@field@linkdefs</code> .....                | 48                      | <code>\@glsxtr@org@Glsxtr@titleshort</code> ..          | 265, 266 |
|   |                         | <code>\@glsxtr@org@Glsxtr@titleshortpl</code> ..        | 265, 266 |
|   |                         | <code>\@glsxtr@org@Glsxtr@titletext</code> ..           | 265, 266 |
|   |                         | <code>\@glsxtr@org@MakeUppercase</code> ....            | 265, 266 |
|   |                         | <code>\@glsxtr@org@checkfirsthyper</code> ....          | 66, 96   |

|  |                   |   |                                      |
|--|-------------------|---|--------------------------------------|
| <code>\@glsxtr@org@delimN</code> .....                       | 46                | <code>\@glsxtr@redef@for@gl@sentries</code> .....         | 7, 20                                |
| <code>\@glsxtr@org@delimR</code> .....                       | 46                | <code>\@glsxtr@redef@styles</code> .....                  | 18, 19, 281                          |
| <code>\@glsxtr@org@doseeglossary</code> ....                 | 11, 12, 98        | <code>\@glsxtr@reg@glosslist</code> .....                 | 97–100, 102                          |
| <code>\@glsxtr@org@gloauto@see</code> .....                  | 20                | <code>\@glsxtr@s@longnewglossaryentry</code> ....         | 27                                   |
| <code>\@glsxtr@org@gls@</code> .....                         | 48                | <code>\@glsxtr@save@preloctag</code> .....                | 45, 47                               |
| <code>\@glsxtr@org@glsignore</code> .....                    | 46                | <code>\@glsxtr@set@entrycount@unset@attr</code> ....      | 82                                   |
| <code>\@glsxtr@org@glspl@</code> .....                       | 49                | <code>\@glsxtr@set@entryunitcount@unset@attr</code> ..... | 93                                   |
| <code>\@glsxtr@org@glsxtr@title@first</code> .               | 265, 266          | <code>\@glsxtr@setup@shortcuts</code> .....               | 17, 20                               |
| <code>\@glsxtr@org@glsxtr@title@first@plural</code><br>..... | 265, 266          | <code>\@glsxtr@shortcuts@val</code> .....                 | 17, 114                              |
| <code>\@glsxtr@org@glsxtr@title@full</code> ..               | 265, 266          | <code>\@glsxtr@swaptwo</code> .....                       | 153                                  |
| <code>\@glsxtr@org@glsxtr@title@full@pl</code> ..            | 265, 266          | <code>\@glsxtr@tag</code> .....                           | 147                                  |
| <code>\@glsxtr@org@glsxtr@title@long</code> ..               | 265, 266          | <code>\@glsxtr@tagging@cs</code> .....                    | 147                                  |
| <code>\@glsxtr@org@glsxtr@title@long@pl</code> ..            | 265, 266          | <code>\@glsxtr@the@H@value</code> .....                   | 7, 8, 51, 52                         |
| <code>\@glsxtr@org@glsxtr@title@plural</code> ..             | 265, 266          | <code>\@glsxtr@the@value</code> .....                     | 7, 8, 50–52                          |
| <code>\@glsxtr@org@glsxtr@title@short</code> .               | 265, 266          | <code>\@glsxtr@this@loctag</code> .....                   | 46, 47                               |
| <code>\@glsxtr@org@glsxtr@title@short@pl</code> ..           | 265, 266          | <code>\@glsxtr@title@label</code> .....                   | 103, 104, 117                        |
| <code>\@glsxtr@org@glsxtr@title@text</code> ..               | 265, 266          | <code>\@glsxtr@tmp</code> .....                           | 18, 106                              |
| <code>\@glsxtr@org@make@glossaries</code> .....              | 97                | <code>\@glsxtr@type</code> .....                          | 135                                  |
| <code>\@glsxtr@org@mark@both</code> .....                    | 264               | <code>\@glsxtr@unitcount@list</code> .....                | 88                                   |
| <code>\@glsxtr@org@mark@right</code> .....                   | 264               | <code>\@glsxtr@unsrt@get@group@title</code> .....         | 116                                  |
| <code>\@glsxtr@org@new@acronym@style</code> ....             | 95, 96            | <code>\@glsxtr@use@see</code> .....                       | 34                                   |
| <code>\@glsxtr@org@post@description</code> .....             | 148               | <code>\@glsxtr@warn@on@exists@sordo</code> .....          | 7, 12                                |
| <code>\@glsxtr@org@see@no@index</code> .....                 | 112               | <code>\@glsxtr@warn@undef@action</code> .....             | 7, 12                                |
| <code>\@glsxtr@org@set@acronym@style</code> ....             | 95, 96            | <code>\@glsxtr@doc@def@false</code> .....                 | 39                                   |
| <code>\@glsxtr@org@the@H@value</code> .....                  | 7, 8              | <code>\@glsxtr@entry@fmt</code> .....                     | 23                                   |
| <code>\@glsxtr@org@prefix</code> .....                       | 9                 | <code>\@glsxtr@rif@has@field</code> .....                 | 25                                   |
| <code>\@glsxtr@org@print@glossary</code> .....               | 43, 101           | <code>\@glsxtr@rif@hyphen@start</code> .....              | 247                                  |
| <code>\@glsxtr@org@warn@dep</code> .....                     | 153               | <code>\@glsxtr@index@aliased</code> .....                 | 68                                   |
| <code>\@glsxtr@p@acrlong@</code> .....                       | 73                | <code>\@glsxtr@index@crossref@false</code> .....          | 13                                   |
| <code>\@glsxtr@p@acrlong@pl@</code> .....                    | 73                | <code>\@glsxtr@index@crossref@true</code> .....           | 13                                   |
| <code>\@glsxtr@p@acrshort@</code> .....                      | 73                | <code>\@glsxtr@in@mark</code> .....                       | 264                                  |
| <code>\@glsxtr@p@acrshort@pl@</code> .....                   | 73                | <code>\@glsxtr@long</code> .....                          | 72, 164                              |
| <code>\@glsxtr@p@long@</code> .....                          | 72                | <code>\@glsxtr@long@pl</code> .....                       | 72, 167                              |
| <code>\@glsxtr@p@long@pl@</code> .....                       | 72                | <code>\@glsxtr@not@in@mark</code> .....                   | 264                                  |
| <code>\@glsxtr@p@plural@</code> .....                        | 72                | <code>\@glsxtr@p</code> .....                             | 78                                   |
| <code>\@glsxtr@p@short@</code> .....                         | 72                | <code>\@glsxtr@p@opt</code> .....                         | 76                                   |
| <code>\@glsxtr@p@short@pl@</code> .....                      | 72                | <code>\@glsxtr@pl</code> .....                            | 42, 43                               |
| <code>\@glsxtr@p@text@</code> .....                          | 72                | <code>\@glsxtr@post@loctag</code> .....                   | 45, 47                               |
| <code>\@glsxtr@page@tag</code> .....                         | 45, 46            | <code>\@glsxtr@preloctag</code> .....                     | 45, 47                               |
| <code>\@glsxtr@page@tag</code> .....                         | 45, 46            | <code>\@glsxtr@set@alias@no@index</code> .....            | 67, 68                               |
| <code>\@glsxtr@prev@unit@count</code> .....                  | 90                | <code>\@glsxtr@short</code> .....                         | 72, 162                              |
| <code>\@glsxtr@print@gloss@opts</code> .....                 | 43, 99, 101       | <code>\@glsxtr@short@pl</code> .....                      | 72, 166                              |
| <code>\@glsxtr@provide@add@storage@key</code> ....           | 22                | <code>\@glsxtr@undef@tag</code> .....                     | 6, 21                                |
| <code>\@glsxtr@provide@storage@key</code> .....              | 22                | <code>\@gobble</code> .....                               | 7, 11, 12, 14, 53, 154               |
| <code>\@glsxtr@record</code> .....                           | 11, 12, 48–50, 52 | <code>\@gobbletwo</code> .....                            | 153                                  |
| <code>\@glsxtr@record@setting</code> .....                   | 8, 11             | <code>\@ifnextchar</code> .....                           | 69, 70                               |
| <code>\@glsxtr@record@setting@also@index</code> ...          | 8                 | <code>\@ifpackage@loaded</code> .....                     | ...                                  |
| <code>\@glsxtr@record@see</code> .....                       | 12                | ... 5, 14, 114, 119, 135, 137, 140, 141, 281              |                                      |
|  |                   | <code>\@ifstar</code> ...                                 | 22, 25, 27, 28, 30, 40, 69, 115, 146 |

|                                    |   |          |  |
|------------------------------------|---|----------|--|
| \@ifundefined .....                | 281   |          |  |
| \@ignored@glossaries .....         | 29–31                                       |          |  |
| \@input .....                      | 112   |          |  |
| \@input@ .....                     | 108   |          |  |
| \@istfilename .....                | 97  |          |  |
| \@makeglossary .....               | 97  |          |  |
| \@mfu@domakefirsttuc .....         | 147, 148                                    |          |  |
| \@mfu@nocaplist .....              | 148   |          |  |
| \@ne .....                         | 84, 93                                      |          |  |
| \@newglossaryentry@defcounters ..  | 82, 90                                      |          |  |
| \@newglossaryentryposthook .....   |   |          |  |
| .....                              | 10, 11, 22, 36, 64                          |          |  |
| \@newglossaryentryprehook .....    |   |          |  |
| .....                              | 10, 11, 22, 27, 28, 36, 64                  |          |  |
| \@nil .....                        | 107, 118                                    |          |  |
| \@nnil .....                       | 143, 145, 146, 152, 154                     |          |  |
| \@no@glstrindexaliased .....       | 68  |          |  |
| \@no@makeglossaries .....          | 112   |          |  |
| \@nopostdesc .....                 | 101   |          |  |
| \@onelevel@sanitize ...            | 10, 35, 42, 104, 117                        |          |  |
| \@onlypreamble .....               |   |          |  |
| .....                              | 43, 46, 92, 113, 115, 142, 144–146          |          |  |
| \@org@glossaryentrynumbers ....    | 100, 101                                    |          |  |
| \@org@newglossaryentryprehook ...  | 27, 28                                      |          |  |
| \@print@unsrt@glossary .....       | 115   |          |  |
| \@printgloss@setsort .....         | 99, 100                                     |          |  |
| \@printglossary .....              | 42, 115                                     |          |  |
| \@printunsrt@glossary@handler .... | 116   |          |  |
| \@printunsrtglossary .....         | 115   |          |  |
| \@sGlsXtrEnableOnTheFly .....      | 40  |          |  |
| \@secondofthree .....              |   |          |  |
| .....                              | 53–55, 61–65, 160, 162, 163, 165, 166, 168  |          |  |
| \@secondoftwo .....                | 49,   |          |  |
| .....                              | 52, 56–64, 66, 72, 96, 101, 152, 159–       |          |  |
| .....                              | 161, 163–168, 182, 202, 215, 236, 264, 266  |          |  |
| \@sglsxtr@provide@storagekey ..... | 22  |          |  |
| \@thirdofthree .....               | 53–   |          |  |
| .....                              | 56, 61–64, 66, 161, 162, 164, 165, 167, 168 |          |  |
| \@thirdoftwo .....                 | 56–60                                       |          |  |
| \@warn@nomakeglossaries .....      | 108   |          |  |
| \@xdy@main@language .....          | 108   |          |  |
| \@xdycrossrefhook .....            | 35  |          |  |
| \@xdy@language .....               | 108   |          |  |
| \@xdy@locationclassorder .....     | 35  |          |  |
| \\ .....                           | 107   |          |  |
| \_ .....                           | 110, 111                                    |          |  |
|                                    |   | <b>A</b> |  |
| \AB .....                          | 15  |          |  |
| \Ab .....                          | 15  |          |  |
| \ab .....                          | 15  |          |  |
| abbreviation styles:               |   |          |  |
| long-hyphen-postshort-hyphen ...   | 252, 255                                    |          |  |
| long-hyphen-short-hyphen .....     | 249, 253                                    |          |  |
| long-postshort-user .....          | 241   |          |  |
| long-short-user .....              | 240   |          |  |
| short .....                        | 185   |          |  |
| short-hyphen-long-hyphen .....     | 257, 258                                    |          |  |
| short-hyphen-postlong-hyphen       | 257, 258, 260                               |          |  |
| short-long-user .....              | 242   |          |  |
| short-nolong .....                 | 184   |          |  |
| short-nolong-desc .....            | 186   |          |  |
| short-postlong-user .....          | 243   |          |  |
| \abbreviationsname .....           | 14  |          |  |
| \abbrvpluralsuffix .....           |   |          |  |
| .....                              | 114, 156, 175, 177, 180, 182,               |          |  |
| .....                              | 183, 185, 187, 191, 192, 194, 195, 197,     |          |  |
| .....                              | 198, 200, 202, 204, 206, 207, 209, 210,     |          |  |
| .....                              | 212, 214, 216, 217, 219, 221, 223, 224,     |          |  |
| .....                              | 226, 227, 229, 230, 232, 234, 236, 239,     |          |  |
| .....                              | 240, 242, 245, 248, 250, 253, 256, 259, 261 |          |  |
| \ABP .....                         | 15  |          |  |
| \Abp .....                         | 15  |          |  |
| \abp .....                         | 15  |          |  |
| \AC .....                          | 16  |          |  |
| \Ac .....                          | 15  |          |  |
| \ac .....                          | 15  |          |  |
| \ACF .....                         | 16  |          |  |
| \Acf .....                         | 16  |          |  |
| \acf .....                         | 15  |          |  |
| \ACFP .....                        | 16  |          |  |
| \Acfp .....                        | 16  |          |  |
| \acfp .....                        | 15  |          |  |
| \ACL .....                         | 16  |          |  |
| \Acl .....                         | 15  |          |  |
| \acl .....                         | 15  |          |  |
| \ACLP .....                        | 16  |          |  |
| \Aclp .....                        | 16  |          |  |
| \aclp .....                        | 15  |          |  |
| \ACP .....                         | 16  |          |  |
| \Acp .....                         | 15  |          |  |
| \acp .....                         | 15  |          |  |
| \ACRfullfmt .....                  | 95  |          |  |
| \Acrfullfmt .....                  | 95  |          |  |
| \acrfullfmt .....                  | 95  |          |  |
| \ACRfullplfmt .....                | 95  |          |  |

|   |  |                                  |  |
|---|--|----------------------------------|--|
| <code>\Acfullplfmt</code> .....         | 95   | category attributes:             |  |
| <code>\acrfullplfmt</code> .....        | 95   | apospplural .....                | 156  |
| <code>\acronymentry</code> .....        | 94   | discardperiod .....              | 150  |
| <code>\acronymfont</code> .....         | 60–64, 75, 95, 96                              | entrycount .....                 | 81, 82, 84, 93                             |
| <code>\acronymname</code> .....         | 14   | firstuc .....                    | 139  |
| <code>\acronymsort</code> .....         | 94   | glossdesc .....                  | 135  |
| <code>\acronymtype</code> .....         | 14, 94, 95                                     | glossdescfont .....              | 136  |
| <code>\acrpluralsuffix</code> .....     | 94, 114  | glossname .....                  | 137  |
| <code>\ACS</code> .....                 | 16   | glossnamefont .....              | 137, 140                                   |
| <code>\Acs</code> .....                 | 15   | headuc .....                     | 266  |
| <code>\acs</code> .....                 | 15   | indexname .....                  | 142  |
| <code>\ACSP</code> .....                | 16   | indexonlyfirst .....             | 68   |
| <code>\Acsp</code> .....                | 15   | insertdots .....                 | 155, 156                                   |
| <code>\acsp</code> .....                | 15   | markshortwords .....             | 155  |
| <code>\actualchar</code> .....          | 145  | markwords .....                  | 155, 156, 246, 248, 256                    |
| <code>\addtolength</code> .....         | 298  | nohyper .....                    | 67   |
| <code>\advance</code> .....             | 84, 93, 113                                    | nohyperfirst .....               | 53–55                                      |
| <code>\AF</code> .....                  | 15   | noshortplural .....              | 156  |
| <code>\Af</code> .....                  | 15   | regular .....                    | 47,  |
| <code>\af</code> .....                  | 15   |                                  | 87, 175, 177, 179–181, 184–186, 188,       |
| <code>\AFP</code> .....                 | 15   |                                  | 189, 191–193, 195, 196, 198–200, 202,      |
| <code>\Afp</code> .....                 | 15   |                                  | 205–209, 211, 213, 215, 218–222, 224–      |
| <code>\afp</code> .....                 | 15   |                                  | 226, 228, 230, 231, 233, 234, 236, 239,    |
| <code>\AL</code> .....                  | 15   |                                  | 244–246, 248–250, 252, 256, 257, 261, 263  |
| <code>\Al</code> .....                  | 15   | <code>\cGLS</code> .....         | 15, 16, 82, 93                             |
| <code>\al</code> .....                  | 15   | <code>\cGls</code> .....         | 15, 82, 93                                 |
| <code>\ALP</code> .....                 | 15   | <code>\cgls</code> .....         | 15, 82, 93                                 |
| <code>\Alp</code> .....                 | 15   | <code>\cGLSformat</code> .....   | 86   |
| <code>\alp</code> .....                 | 15   | <code>\cGlsformat</code> .....   | 85   |
| <code>\AnyTrackedLanguages</code> ..... | 281  | <code>\cglsformat</code> .....   | 85, 87                                     |
| <code>\appto</code> .....               | 10, 11, 19, 22, 33,                            | <code>\cGLSpl</code> .....       | 15, 16, 82, 93                             |
|   | 35–37, 64, 69, 82, 90, 116, 117, 141, 151, 154 | <code>\cGlspl</code> .....       | 15, 82, 93                                 |
| <code>\AS</code> .....                  | 15   | <code>\cglspl</code> .....       | 15, 82, 93                                 |
| <code>\As</code> .....                  | 15   | <code>\cGLSplformat</code> ..... | 86   |
| <code>\as</code> .....                  | 15   | <code>\cGlsplformat</code> ..... | 85   |
| <code>\ASP</code> .....                 | 15   | <code>\cglsplformat</code> ..... | 85, 87                                     |
| <code>\Asp</code> .....                 | 15   | <code>\char</code> .....         | 104  |
| <code>\asp</code> .....                 | 15   | <code>\columnwidth</code> .....  | 44, 45                                     |
| <code>\AtBeginDocument</code> .....     | 21, 44, 45, 114                                | <code>\count@</code> .....       | 84, 92, 93                                 |
| <code>\AtEndDocument</code> .....       | 38, 83, 92, 108                                | <code>\csappto</code> .....      | 27   |
| <b>B</b>                                |  | <code>\csdef</code> .....        | 22, 25, 27, 64–66, 83,                     |
| <code>babel package</code> .....        | 142, 144, 151                                  |                                  | 88, 89, 91, 130, 172–174, 181, 202, 215,   |
| <code>\begin</code> .....               | 105, 110, 116, 283–287                         |                                  | 236, 240–242, 244, 253, 255, 258, 260, 283 |
| <code>\begingroup</code> .....          | 7, 68, 115                                     | <code>\cseappto</code> .....     | 31   |
| <code>\bgroup</code> .....              | 27, 28, 100                                    | <code>\csedef</code> .....       | 25, 89                                     |
| <b>C</b>                                |  | <code>\csgdef</code> .....       | 26, 29–31, 39, 46, 83, 89, 91, 92          |
| <code>\catcode</code> .....             | 112  | <code>\cslet</code> .....        | 26, 28, 101                                |
|   |  | <code>\csletcs</code> .....      | 26, 174                                    |



|  |  |
|--|--|
| <code>\csname</code> . . . . .                     | 6, 8, 22, 30, 35, 43, 47, 49, 51,<br>52, 60–66, 68, 76, 89, 97, 98, 105, 108,<br>111, 112, 116, 135, 153, 160–169, 174, 297  |
| <code>\cspreto</code> . . . . .                    | 27   |
| <code>\csuse</code> . . . . .                      | 23, 30, 37, 46, 65, 66,<br>77–79, 88–92, 100, 104–106, 115, 117,<br>118, 130, 141, 149, 150, 172, 174, 290–293   |
| <code>\csxdef</code> . . . . .                     | 33, 36, 89, 92, 104  |
| <code>\currentglossary</code> . . . . .            | 101  |
| <code>\CurrentOption</code> . . . . .              | 19, 282  |
| <code>\CurrentTrackedLanguageTag</code> . . . . .  | 114  |
| <code>\CurrentTrackedTag</code> . . . . .          | 281  |
| <code>\CustomAbbreviationFields</code> . . . . .   | 157, 175–179, 181, 183,<br>185, 187, 189–196, 200, 201, 204, 205,<br>207, 209, 210, 213, 215, 217–220, 222–<br>225, 227, 228, 232, 234, 236, 238, 240–<br>246, 248–251, 253, 255–258, 260, 261, 263  |
| <b>D</b>   |  |
| <code>\DeclareAcronymList</code> . . . . .         | 94   |
| <code>\DeclareOption</code> . . . . .              | 5, 282   |
| <code>\DeclareOptionX</code> . . . . .             | 5, 19  |
| <code>\def</code> . . . . .                        | 7–12, 21, 28, 30, 34–36, 38, 40–<br>43, 45, 47–64, 70–75, 85–87, 94, 98–102,<br>104–107, 116, 117, 143–148, 151–156,<br>159–169, 172, 247, 250, 252, 256, 258, 298   |
| <code>\defglstentryfmt</code> . . . . .            | 29–31  |
| <code>\define@boolkey</code> . . . . .             | 13, 67   |
| <code>\define@choicekey</code> . . . . .           | 7, 11, 13, 16, 18, 50, 101   |
| <code>\define@key</code> . . . . .                 | 10, 11, 18, 19, 22, 36, 50–52, 64, 153   |
| <code>\DefineAcronymSynonyms</code> . . . . .      | 17   |
| <code>\delimN</code> . . . . .                     | 46   |
| <code>\delimR</code> . . . . .                     | 46   |
| <code>\detokenize</code> . . . . .                 | 40   |
| <code>\dimen@</code> . . . . .                     | 96, 290–297  |
| <code>\dimen@i</code> . . . . .                    | 291–293  |
| <code>\dimen@ii</code> . . . . .                   | 291–293  |
| <code>\dimexpr</code> . . . . .                    | 44, 45, 289  |
| <code>\disable@keys</code> . . . . .               | 14, 21, 39   |
| <code>\do</code> . . . . .                         | 6, 18, 38, 82, 94, 97, 100, 105, 116, 135, 147   |
| <code>\do@gl@link@checkfirsthyper</code> . . . . . | 23, 48–51, 60–64, 159–168  |
| <code>\do@gl@disablehyperinlist</code> . . . . .   | 51, 67   |
| <code>doc package</code> . . . . .                 | 145  |
| <code>\dolistcsloop</code> . . . . .               | 24   |
| <code>\DTLifinlist</code> . . . . .                | 97–99, 102   |
| <code>\DTLifint</code> . . . . .                   | 104  |
| <b>E</b>   |  |
| <code>\eappto</code> . . . . .                     | 10, 18, 29–31, 116, 142, 282   |
| <code>\edef</code> . . . . .                       | 6, 8, 9, 29–<br>32, 34, 36, 37, 51, 52, 66, 68, 71, 88, 89,<br>91, 97, 98, 102, 104, 106–108, 112, 136,<br>137, 139, 140, 143, 145, 146, 153, 291–293  |
| <code>\eglssetwidest</code> . . . . .              | 290–297  |
| <code>\egroup</code> . . . . .                     | 28, 101  |
| <code>\else</code> . . . . .                       | 8–10, 13, 14, 16,<br>18, 19, 26, 39, 40, 45, 47, 50–52, 68, 69,<br>84, 96, 99–101, 104, 106, 107, 110, 111,<br>113, 141–143, 145, 146, 152, 154–156,<br>163–169, 171, 176, 178, 180–188, 191–<br>206, 208–216, 218–221, 223–233, 235,<br>237, 239–241, 243, 245–247, 250, 252–<br>254, 256, 258–260, 262, 284–288, 298, 299            |
| <code>\emph</code> . . . . .                       | 217  |
| <code>\empty</code> . . . . .                      | 106, 107   |
| <code>\encapchar</code> . . . . .                  | 145  |
| <code>\end</code> . . . . .                        | 105, 110, 111, 116, 283–287  |
| <code>\end@glxtr@display@loc</code> . . . . .      | 106  |
| <code>\endcsname</code> . . . . .                  | 6, 8, 22, 30, 35, 43, 47, 49, 51,<br>52, 60–66, 68, 76, 89, 97, 98, 105, 108,<br>111, 112, 116, 135, 153, 160–169, 174, 297  |
| <code>\endgroup</code> . . . . .                   | 8, 68, 115   |
| entry categories:                                  |  |
| abbreviation . . . . .                             | 169  |
| general . . . . .                                  | 130, 132   |
| index . . . . .                                    | 133  |
| <code>\epreto</code> . . . . .                     | 142  |
| <code>\equal</code> . . . . .                      | 111  |
| <code>etoolbox package</code> . . . . .            | 5  |
| <code>\expandafter</code> . . . . .                | 19,<br>22, 23, 33, 34, 38, 40–42, 65, 66, 70, 76,<br>87, 88, 97–99, 102, 105–107, 116, 118,<br>135, 138, 139, 142, 145, 152, 155–157, 247  |
| <code>\expandonce</code> . . . . .                 | 94, 143, 176, 250  |
| <b>F</b>   |  |
| <code>\fi</code> . . . . .                         | 7–10, 12–14, 16–18, 20, 26, 33,<br>36, 38–40, 44, 45, 47, 50–52, 68, 69, 84,<br>92, 93, 96, 99–101, 104, 106–108, 110–<br>114, 141–144, 146, 152, 154, 155, 157,<br>163–169, 171, 176, 178, 180–188, 191–<br>206, 208–216, 218–221, 223–233, 235,<br>237, 239–241, 243, 245–247, 250, 252–<br>254, 256, 258–260, 262, 284–288, 290–299 |
| first use . . . . .                                | 300  |
| flag . . . . .                                     | 300  |
| text . . . . .                                     | 300  |



|  |                          |  |                             |
|--|--------------------------|--|-----------------------------|
| <code>\firstacronymfont</code> .....                             | 95, 96                   | <code>\glossarysection</code> .....                    | 105, 111, 116, 117          |
| <code>fontspec</code> package .....                              | 114                      | <code>\glossarytitle</code> ...                        | 30, 100, 101, 105, 111, 116 |
| <code>\footnote</code> .....                                     | 179                      | <code>\glossarytoctitle</code> ...                     | 30, 100, 105, 111, 116      |
| <code>\forall glossaries</code> 38, 116, 133, 135, 290–296       |                          | <code>\glossentry</code> .....                         | 101, 118, 119, 283–287, 298 |
| <code>\forall glsentries</code> .....                            | 84, 93                   | <code>\glossentrydesc</code> .....                     | 283–289                     |
| <code>\ForEachTrackedDialect</code> .....                        | 281                      | <code>\glossentryname</code> .....                     | 283–287, 298                |
| <code>\forall glsentries</code> .....                            | 6, 38, 133, 135, 290–296 | <code>\glossentrysymbol</code> .....                   | 284–289                     |
| <code>\forall listcsloop</code> .....                            | 24, 93, 105              | <code>\glossxtrsetpopts</code> .....                   | 149                         |
| <code>\forall listloop</code> .....                              | 102, 103, 148            | <code>\GLS</code> .....                                | 82, 93                      |
| <code>\futurelet</code> .....                                    | 152                      | <code>\Gls</code> .....                                | 41, 82, 93                  |
| <b>G</b>   |                          |  |                             |
| <code>\gdef</code> .....   | 46, 144, 145             | <code>\gls</code> .....                                | 26, 41, 43, 82, 93, 98, 109 |
| <code>\Genacrfullformat</code> .....                             | 95                       | <code>\gls@assign@desc</code> .....                    | 28                          |
| <code>\genacrfullformat</code> .....                             | 95                       | <code>\gls@assign@field</code> .....                   | 10, 11, 22, 64              |
| <code>\GenericAcronymFields</code> .....                         | 95                       | <code>\gls@checkseeallowed</code> .....                | 40, 97                      |
| <code>\Genplacrfullformat</code> .....                           | 95                       | <code>\gls@codepage</code> .....                       | 108                         |
| <code>\genplacrfullformat</code> .....                           | 95                       | <code>\gls@defdocnewglossaryentry</code> .....         | 82, 90                      |
| <code>\glo@grabfirst</code> .....                                | 118                      | <code>\gls@defglossaryentry</code> .....               | 28, 41, 42                  |
| <code>\glo@name</code> .....                                     | 138, 139                 | <code>\gls@dotocitle</code> .....                      | 100, 101                    |
| <code>\gloaliaslabel</code> .....                                | 71                       | <code>\gls@glossary</code> .....                       | 35                          |
| <code>\global</code> .....                                       | 9, 28, 101, 118          | <code>\gls@level</code> .....                          | 118                         |
| <code>\glo linkprefix</code> .....                               | 51, 71, 114, 117         | <code>\gls@noidxglossary</code> .....                  | 98                          |
| <code>glossaries</code> package 12, 19, 20, 33, 35, 37, 100, 282 |                          | <code>\gls@org@glossaryentryfield</code> .....         | 101                         |
| <code>glossaries-accsupp</code> package .....                    | 18, 19, 119              | <code>\gls@org@glossarysubentryfield</code> ....       | 101                         |
| <code>glossaries-extra</code> package .....                      | 2                        | <code>\gls@save@numberlist</code> .....                | 45, 47                      |
| <code>glossaries-extra-stylemods</code> package .                | 18, 149, 281             | <code>\gls@set@xr@key</code> .....                     | 35, 36                      |
| <code>glossaries.sty</code> package .....                        | 28                       | <code>\gls@tmplen</code> .....                         | 290–296, 298                |
| <code>\GlossariesExtraWarning</code> .....                       |                          | <code>\gls@type</code> .....                           | 98                          |
| ..... 6, 14, 27, 40, 42, 96, 98, 106,                            |                          | <code>\glsabbrvdefaultfont</code> ...                  | 159, 175, 177,              |
| 110, 112, 116, 136, 137, 139, 140, 147, 174                      |                          | 180, 182, 183, 185, 187, 238, 247, 250, 261            |                             |
| <code>\GlossariesExtraWarningNoLine</code> .                     | 14, 84, 93               | <code>\glsabbrvfont</code> .                           | 217–229, 231, 232, 234, 236 |
| <code>\GlossariesWarning</code> 46, 98, 100, 102, 103, 172       |                          | <code>\glsabbrvfont</code> ..                          | 73, 74, 95, 159, 163, 164,  |
| <code>\GlossariesWarningNoLine</code> .....                      | 98, 108                  | 166, 167, 171, 175–183, 185, 187, 189,                 |                             |
| <code>glossary</code> styles:                                    |                          | 191, 192, 194, 195, 197, 198, 200, 202,                |                             |
| <code>alttree</code> .....                                       | 288, 289, 297            | 204, 206, 207, 209, 210, 212, 214, 216,                |                             |
| <code>inline</code> .....  | 288                      | 217, 219, 221, 223, 224, 226, 227, 229,                |                             |
| <code>listdotted</code> .....                                    | 283                      | 231, 232, 234, 236, 239, 240, 242, 244–                |                             |
| <code>listdottedstyle</code> .....                               | 283                      | 246, 248, 250, 251, 253, 254, 256, 259, 261            |                             |
| <code>sublistdotted</code> .....                                 | 283                      | <code>\glsabbrvhyphenfont</code> 247–249, 253, 255–260 |                             |
| <code>glossary-long</code> package .....                         | 284                      | <code>\glsabbrvonlyfont</code> .....                   | 261, 263                    |
| <code>glossary-longbooktabs</code> package .....                 | 284                      | <code>\glsabbrvscfont</code> .....                     | 190–198, 200–202            |
| <code>glossary-mcols</code> package .....                        | 282                      | <code>\glsabbrvsmfont</code> .                         | 203–207, 209, 210, 212–216  |
| <code>glossary-tree</code> package .....                         | 282                      | <code>\glsabbrvuserfont</code> .....                   | 238–245                     |
| <code>\glossaryentrynumbers</code> 47, 100, 101, 118, 119        |                          | <code>\GLSaccessdesc</code> .....                      | 56                          |
| <code>\glossaryheader</code> ....                                | 105, 116, 283–287, 298   | <code>\Glsaccessdesc</code> .....                      | 56, 136, 146                |
| <code>\glossaryname</code> .....                                 | 100                      | <code>\glsaccessdesc</code> .....                      | 56, 136, 150                |
| <code>\glossarypostamble</code> .....                            | 105, 116, 117            | <code>\GLSaccessdescplural</code> .....                | 57                          |
| <code>\glossarypreamble</code> .....                             | 105, 116                 | <code>\Glsaccessdescplural</code> .....                | 57                          |
|  |                          | <code>\glsaccessdescplural</code> .....                | 57                          |
|  |                          | <code>\GLSaccessfirst</code> .....                     | 54                          |

|  |   |  |
|--|---|--|
| <code>\Glsaccessfirst</code> .....       | 54  | 228–231, 233, 235, 237, 239, 241, 243,             |
| <code>\glsaccessfirst</code> .....       | 54  | 246, 248, 249, 251, 254, 256, 257, 259, 262        |
| <code>\GLSaccessfirstplural</code> ..... | 56  | <code>\GLSaccesssymbol</code> .....                |
| <code>\Glsaccessfirstplural</code> ..... | 55  | <code>\Glsaccesssymbol</code> .....                |
| <code>\glsaccessfirstplural</code> ..... | 55  | <code>\glsaccesssymbol</code> .....                |
| <code>\Glsaccesslong</code> .....        | 63, 158, 165, 176,                          | <code>\GLSaccesssymbolplural</code> .....          |
|  | 184, 187, 188, 191, 194, 197, 199, 204,     | <code>\Glsaccesssymbolplural</code> .....          |
|  | 208, 210–212, 218, 220, 225, 227–229,       | <code>\glsaccesssymbolplural</code> .....          |
|  | 231, 233, 239–241, 249–251, 254, 257, 262   | <code>\GLSacesstext</code> .....                   |
| <code>\glsaccesslong</code> .....        |   | <code>\Glsacesstext</code> .....                   |
|  | .. 62, 63, 157, 164, 165, 176, 178, 180–    | <code>\glsacesstext</code> .....                   |
|  | 183, 185–188, 191, 193, 194, 196–201,       | <code>\glsacrshortcutstrue</code> .....            |
|  | 203, 204, 206, 208–216, 218, 219, 221,      | <code>\glsacspacemax</code> .....                  |
|  | 223, 224, 226–233, 235, 237, 239–241,       | <code>\glsadd</code> .....                         |
|  | 243, 245, 246, 248, 250, 251, 254, 256, 262 | <code>\glsadd options</code>                       |
| <code>\Glsaccesslongpl</code> .....      | 64, 158, 168, 176,                          | <code>theHvalue</code> .....                       |
|  | 184, 187, 188, 191, 194, 197–199, 205,      | <code>thevalue</code> .....                        |
|  | 208, 211–213, 218, 220, 225, 227–229,       | <code>\glsaddstoragekey</code> .....               |
|  | 231, 233, 239, 241, 249, 251, 254, 257, 262 | <code>\glsbackslash</code> .....                   |
| <code>\glsaccesslongpl</code> .....      | 63,   | <code>\glscapscase</code> .....                    |
|  | 64, 158, 168, 169, 176, 178, 180, 181,      | <code>\glscategory</code> .....                    |
|  | 183, 184, 186–188, 191, 193, 194, 196–      | .. 47, 53, 66, 73, 74, 131, 132, 135–137,          |
|  | 199, 201, 203, 204, 206, 208–216, 218,      | 139–141, 146, 149, 150, 159–164, 166, 167          |
|  | 220, 221, 223, 225–233, 235, 237, 239–      | <code>\glscategorylabel</code> .....               |
|  | 241, 243, 246, 248, 250, 251, 254, 256, 262 | ..... 66, 153, 155, 156, 181, 202,                 |
| <code>\GLSaccessname</code> .....        | 56  | 215, 236, 240–242, 244, 253, 255, 258, 260         |
| <code>\Glsaccessname</code> .....        | 56  | <code>\glsclsebrace</code> .....                   |
| <code>\glsaccessname</code> .....        | 56  | <code>\glscurrententrylabel</code> .....           |
| <code>\GLSaccessplural</code> .....      | 55  | ..... 45–47, 101, 107, 116, 148, 149               |
| <code>\Glsaccessplural</code> .....      | 55  | <code>\glscurrentfieldvalue</code> .....           |
| <code>\glsaccessplural</code> .....      | 54  | <code>\glscustomtext</code> ..                     |
| <code>\Glsaccessshort</code> .....       |   | 6, 14, 27, 99, 100, 109, 115, 117                  |
|  | .. 61, 163, 171, 178, 180–183, 186, 193,    | <code>\glsdescriptionaccessdisplay</code> ..       |
|  | 196, 201, 203, 206, 209, 214, 216, 221,     | <code>\glsdescriptionpluralaccessdisplay</code> .. |
|  | 223, 226, 235, 237, 243, 246, 254, 259, 260 | <code>\glsdescwidth</code> .....                   |
| <code>\glsaccessshort</code> .....       | 60, 61, 157, 158,                           | 283, 285–287                                       |
|  | 163, 164, 171, 176, 178, 180, 182–186,      | <code>\glsdetoklabel</code> .....                  |
|  | 188, 191, 192, 194–197, 199–206, 208–       | 8, 24–26, 28,                                      |
|  | 212, 214, 216, 218–221, 223–226, 228,       | 32–34, 38–40, 51, 52, 68, 71, 83, 88–93,           |
|  | 229, 231, 233, 235, 237, 239, 241, 243,     | 97, 101–103, 118, 135, 138, 139, 291–293           |
|  | 245, 248, 249, 251, 254, 256, 257, 259, 262 | <code>\glsdisplaynumberlist</code> .....           |
| <code>\Glsaccessshortpl</code> .....     | 62,   | <code>\glsdohyperlink</code> .....                 |
|  | 166, 171, 178, 180–183, 186, 193, 196,      | <code>\glsdohypertarget</code> .....               |
|  | 201, 203, 206, 209, 214–216, 221, 223,      | <code>\glsdoifexists</code> .....                  |
|  | 226, 235, 237, 243, 246, 254, 259, 260, 262 | 13, 25, 31,  |
| <code>\glsaccessshortpl</code> .....     | 61, 62,                                     | 33, 34, 48, 49, 52, 60–64, 102, 103, 159–168       |
|  | 158, 166, 167, 171, 176, 178, 180–182,      | <code>\glsdoifexistsordo</code> .....              |
|  | 184–186, 188, 191, 193–199, 201–206,        | <code>\glsdoifexistsorwarn</code> ....             |
|  | 208–214, 216, 218, 220, 221, 223–226,       | <code>\glsdoifnoexists</code> .....                |
|  |   | <code>\glsdonohyperlink</code> .....               |
|  |   | <code>\glsdosanitizesort</code> .....              |
|  |   | <code>\glsenableentrycount</code> .....            |

|   |   |   |   |
|---|---|---|---|
| <code>\glsenableentryunitcount</code> ..... | 83, 93                                      | <code>\Glsentryuserv</code> .....               | 59  |
| <code>\glsentrycounter</code> .....         | 107   | <code>\glsentryuserv</code> .....               | 60  |
| <code>\glsentrycurrcount</code> .....       | 83, 84, 90                                  | <code>\Glsentryuservi</code> .....              | 60  |
| <code>\Glsentrydesc</code> .....            | 123, 128, 137                               | <code>\glsentryuservi</code> .....              | 60  |
| <code>\glsentrydesc</code> .....            | 123, 124, 128, 137                          | <code>\glsfieldfetch</code> .....               | 71  |
| <code>\Glsentrydescplural</code> .....      | 124, 128                                    | <code>\glsfieldxdef</code> .....                | 135   |
| <code>\glsentrydescplural</code> .....      | 124, 128                                    | <code>\glsfindwidesttoplevelname</code> .....   | 290   |
| <code>\Glsentryfirst</code> .....           | 87, 121, 127                                | <code>\GLSfirst</code> .....                    | 270   |
| <code>\glsentryfirst</code> .....           | 87, 121, 127, 277                           | <code>\Glsfirst</code> .....                    | 270   |
| <code>\Glsentryfirstplural</code> .....     | 88, 122, 127                                | <code>\glsfirst</code> .....                    | 270   |
| <code>\glsentryfirstplural</code> .....     | 87, 121, 122, 127, 278                      | <code>\glsfirstabbrvdefaultfont</code> .....    |   |
| <code>\glsentryfmt</code> .....             | 29–31                                       |   | 159, 175, 178, 180, 182, 183, 185, 187, 250 |
| <code>\Glsentryfull</code> .....            | 95  | <code>\glsfirstabbrvmfont</code> .....          | 217–237                                     |
| <code>\glsentryfull</code> .....            | 95  | <code>\glsfirstabbrvfont</code> .....           | 95, 157, 158,                               |
| <code>\Glsentryfullpl</code> .....          | 95  |   | 175–188, 191, 192, 194, 195, 197, 198,      |
| <code>\glsentryfullpl</code> .....          | 95  |   | 200, 202, 204, 206, 207, 209, 210, 212,     |
| <code>\glsentryitem</code> .....            | 283–287, 298                                |   | 214, 216, 217, 219, 221, 223, 224, 226,     |
| <code>\Glsentrylong</code> .....            | 74, 75, 87, 126, 129                        |   | 227, 229, 231, 232, 234, 236, 239, 240,     |
| <code>\glsentrylong</code> .....            | 74, 75, 87, 125, 126,                       |   | 242, 245, 248, 250, 251, 253, 256, 259, 261 |
|   | 129, 181, 202, 215, 236, 242, 244, 258, 278 | <code>\glsfirstabbrvhyphenfont</code> .....     |   |
| <code>\Glsentrylongpl</code> .....          | 74, 75, 88, 126, 129                        |   | 247–249, 252, 253, 256–260                  |
| <code>\glsentrylongpl</code> .....          | 74, 75, 87, 126, 129, 279                   | <code>\glsfirstabbrvonlyfont</code> .....       | 261, 262                                    |
| <code>\Glsentryname</code> .....            | 119, 126, 138–141                           | <code>\glsfirstabbrvscfont</code> .....         | 190–203                                     |
| <code>\glsentryname</code> .....            | 119, 120, 126, 127, 142, 290–297            | <code>\glsfirstabbrvsmfont</code> .....         | 204–216                                     |
| <code>\glsentrynumberlist</code> .....      | 99, 103, 295–297                            | <code>\glsfirstabbrvuserfont</code> .....       | 239–246                                     |
| <code>\Glsentryplural</code> .....          | 120, 127                                    | <code>\glsfirstaccessdisplay</code> .....       | 121   |
| <code>\glsentryplural</code> .....          | 120, 121, 127, 276, 277                     | <code>\glsfirstlongdefaultfont</code> .....     |   |
| <code>\glsentryprevcount</code> .....       | 83, 84, 90                                  |   | 175, 178, 183, 185, 187, 190–199, 204–      |
| <code>\glsentryprevmaxcount</code> .....    | 91  |   | 213, 217, 218, 221, 222, 224–228, 231, 232  |
| <code>\glsentryprevtotalcount</code> .....  | 90  | <code>\glsfirstlongemfont</code> .....          |   |
| <code>\Glsentryshort</code> .....           | 73, 75, 125, 129                            |   | 219, 220, 222–224, 228–230, 232, 233        |
| <code>\glsentryshort</code> .....           | 73–75,                                      | <code>\glsfirstlongfont</code> .....            | 157, 158, 175–178,                          |
|   | 96, 124, 125, 128, 129, 240, 241, 252, 275  |   | 180, 182–189, 191, 192, 194, 195, 197,      |
| <code>\Glsentryshortpl</code> .....         | 74, 75, 125, 129                            |   | 198, 200, 202, 204, 206, 207, 209, 210,     |
| <code>\glsentryshortpl</code> .....         | 74, 75, 125, 129, 275, 276                  |   | 212, 214, 216, 218, 219, 221, 223, 224,     |
| <code>\Glsentrysymbol</code> .....          | 122, 128                                    |   | 226, 227, 229, 231, 232, 234, 236, 239,     |
| <code>\glsentrysymbol</code> .....          | 122, 128, 294, 295                          |   | 240, 242, 245, 248, 250, 254, 256, 259, 261 |
| <code>\Glsentrysymbolplural</code> .....    | 123, 128                                    | <code>\glsfirstlongfootnotefont</code> .....    |   |
| <code>\glsentrysymbolplural</code> .....    | 123, 128                                    |   | 179–183, 200–203, 213–216, 234–237          |
| <code>\Glsentrytext</code> .....            | 120, 127                                    | <code>\glsfirstlonghyphenfont</code> .....      | 247–259                                     |
| <code>\glsentrytext</code> .....            | 71, 120, 127, 276                           | <code>\glsfirstlongonlyfont</code> .....        | 261–263                                     |
| <code>\Glsentryuseri</code> .....           | 58  | <code>\glsfirstlonguserfont</code> .....        | 239–246                                     |
| <code>\glsentryuseri</code> .....           | 58  | <code>\GLSfirstplural</code> .....              | 270, 271                                    |
| <code>\Glsentryuserii</code> .....          | 59  | <code>\Glsfirstplural</code> .....              | 271   |
| <code>\glsentryuserii</code> .....          | 59  | <code>\glsfirstplural</code> .....              | 271   |
| <code>\Glsentryuseriii</code> .....         | 59  | <code>\glsfirstpluralaccessdisplay</code> ..... | 121, 122                                    |
| <code>\glsentryuseriii</code> .....         | 59  | <code>\glsforeachincategory</code> .....        | 172   |
| <code>\Glsentryuseriv</code> .....          | 59  | <code>\glsgenentryfmt</code> .....              | 47, 48                                      |
| <code>\glsentryuseriv</code> .....          | 59  |   |   |

|   |                                  |  |                                   |
|---|----------------------------------|--|-----------------------------------|
| <code>\glsgetattribute</code> .....         | 70,                              | <code>noindex</code> .....                       | 7, 67, 263                        |
| 71, 84, 88–90, 107, 136, 137, 139, 140, 142 |                                  | <code>theHvalue</code> .....                     | 51                                |
| <code>\glsgetcategoryattribute</code> ..... | 131                              | <code>thevalue</code> .....                      | 51                                |
| <code>\glsgetwidestname</code> .....        | 289                              | <code>wrgloss</code> .....                       | 50                                |
| <code>\glsgroupheading</code> .....         | 118, 283–287, 298                | <code>\glslinkcheckfirsthyperhook</code> .....   | 67                                |
| <code>\glsgroupskip</code> .....            | 118, 284–288, 299                | <code>\glslinkpostsetkeys</code> .....           | 51                                |
| <code>\glshasattribute</code> .             | 70, 84, 89, 91, 93, 107,         | <code>\glslinkvar</code> .....                   | 69, 70                            |
| 136, 137, 139, 140, 142, 175, 177, 179,     |                                  | <code>\glslistdottedwidth</code> .....           | 283                               |
| 180, 182, 185, 186, 189, 190, 192, 193,     |                                  | <code>\glslocalunset</code> .....                | 50                                |
| 200, 202, 204–207, 213, 215, 217–224,       |                                  | <code>\glslongaccessdisplay</code> .....         | 125, 126                          |
| 230, 234, 236, 239, 240, 242, 244–246,      |                                  | <code>\glslongdefaultfont</code> .....           | 159, 175,                         |
| 248–250, 252, 253, 255–258, 260, 261, 263   |                                  | 178, 179, 183, 185, 187, 191, 192, 194,          |                                   |
| <code>\glshascategoryattribute</code> ..... | 131                              | 195, 197–199, 204, 206, 207, 209–212,            |                                   |
| <code>\glshyperlink</code> .....            | 71                               | 218, 221, 224, 226, 227, 231, 238, 248, 261      |                                   |
| <code>\glshypernavsep</code> .....          | 105                              | <code>\glslongemfont</code>                      | 217, 219, 222, 223, 229, 232, 233 |
| <code>\glshypernumber</code> .....          | 108, 141                         | <code>\glslongfont</code> ..                     | 74, 75, 159, 164, 165, 168,       |
| <code>\glsifattribute</code> .....          | 50, 53, 67, 69,                  | 169, 175, 176, 178, 180, 182, 183, 185,          |                                   |
| 77, 133, 136–139, 148, 150, 151, 266–274    |                                  | 187, 189, 191, 192, 194, 195, 197, 198,          |                                   |
| <code>\glsifcategory</code> .....           | 133                              | 200, 202, 204, 206, 207, 209, 210, 212,          |                                   |
| <code>\glsifcategoryattribute</code> .....  |                                  | 214, 216, 218, 219, 221, 223, 224, 226,          |                                   |
| .....                                       | 66, 131, 132, 155, 156           | 227, 229, 231, 232, 234, 236, 239, 240,          |                                   |
| <code>\glsifnotregular</code> .....         | 53                               | 242, 245, 248, 250, 254, 256, 259, 261, 263      |                                   |
| <code>\glsifnotregularcategory</code> ..... | 132                              | <code>\glslongfootnotefont</code> .....          |                                   |
| <code>\glsifplural</code> .....             |                                  | 179, 180, 182, 200, 202, 214, 216, 234, 236      |                                   |
| 49, 52, 54, 55, 57, 58, 60–64, 151, 159–170 |                                  | <code>\glslonghyphenfont</code> .....            |                                   |
| <code>\glsifregular</code> .....            | 47, 53, 87, 88                   | .....  | 248, 250, 252–254, 256, 258, 259  |
| <code>\glsifregularcategory</code> .....    | 132                              | <code>\glslongonlyfont</code> .....              | 261                               |
| <code>\glsifusetranslator</code> .....      | 30                               | <code>\glslongpltok</code> .....                 | 157, 175–                         |
| <code>\glsignore</code> .....               | 46                               | 179, 187, 189–193, 197, 200, 204–207,            |                                   |
| <code>\glsinlinedescformat</code> .....     | 288                              | 210, 213, 217–222, 224, 227–229, 232,            |                                   |
| <code>\glsinlinesubdescformat</code> .....  | 288                              | 234, 239–246, 248–253, 255–257, 261, 263         |                                   |
| <code>\glsinsert</code> .....               | 49,                              | <code>\glslongpluralaccessdisplay</code> .....   | 126                               |
| 52, 60–64, 159–171, 247, 253, 255, 258, 260 |                                  | <code>\glslongtok</code> .....                   | 94, 155, 157,                     |
| <code>\glskeylisttok</code> .....           | 94, 95, 155, 157                 | 175–179, 181, 183, 185, 187, 189–197,            |                                   |
| <code>\glslabel</code> .....                | 32, 47, 48, 50, 51, 66–68,       | 200, 202, 204–207, 209, 210, 213, 215,           |                                   |
| 70, 71, 96, 149, 150, 169–171, 181, 202,    |                                  | 217–224, 226–229, 232, 234, 236, 239–            |                                   |
| 215, 236, 240–242, 244, 253, 255, 258, 260  |                                  | 242, 244–246, 248–253, 255–258, 261–263          |                                   |
| <code>\glslabeltok</code> .....             |                                  | <code>\glslonguserfont</code> .....              | 238–243, 245                      |
| .....                                       | 94, 155, 157, 175–180, 182, 183, | <code>\glsnameaccessdisplay</code> .....         | 119, 138, 140                     |
| 185–187, 189–195, 197, 200, 202, 204–       |                                  | <code>\glsnamefont</code> .....                  | 137–141                           |
| 207, 209, 210, 213, 215, 217–224, 226,      |                                  | <code>\glsnavhyperlink</code> .....              | 105                               |
| 227, 229, 230, 232, 234, 236, 239–246,      |                                  | <code>\glsnextpages</code> .....                 | 101                               |
| 248–250, 252, 253, 255–258, 260, 261, 263   |                                  | <code>\glsnoidxdisplayloc</code> .....           | 103                               |
| <code>\glsletentryfield</code> .....        | 143                              | <code>\glsnoidxdisplaylocclsthandler</code> .... | 102                               |
| <code>\glslink</code> .....                 | 95                               | <code>\glsnoidxloclist</code> .....              | 103, 118, 119                     |
| <code>\glslink options</code>               |                                  | <code>\glsnoidxnumberlistloophandler</code> .... | 103                               |
| counter .....                               | 8                                | <code>\glsnonextpages</code> .....               | 101                               |
| format .....                                | 141                              | <code>\glsnonumberlistfalse</code> .....         | 45                                |
| hyper .....                                 | 263                              | <code>\glsnonumberlisttrue</code> .....          | 45                                |

|  |   |  |                        |
|--|---|--|------------------------|
| <code>\glsnumberlistloop</code> .....  | 99  | <code>\glstarget</code> .....                            | 283–288, 298           |
| <code>\glsnumlistlastsep</code> .....  | 102   | <code>\GLStext</code> .....                              | 268                    |
| <code>\glsnumlistsep</code> .....  | 102   | <code>\Glstext</code> .....                              | 268, 269               |
| <code>\glsopenbrace</code> .....   | 35, 110–112   | <code>\glstext</code> .....                              | 268                    |
| <code>\glsorder</code> .....   | 97  | <code>\glstextaccessdisplay</code> .....                 | 120                    |
| <code>\glspagelistwidth</code> .....   | 283, 285–287  | <code>\glstextformat</code> .....                        | 50, 51                 |
| <code>\glspar</code> .....   | 117   | <code>\glstextup</code> .....                            | 190                    |
| <code>\GLSpl</code> .....  | 82, 93  | <code>\glstreeindent</code> .....                        | 297, 298               |
| <code>\Glspl</code> .....  | 42, 82, 93  | <code>\glstreenamebox</code> .....                       | 298                    |
| <code>\glspl</code> .....  | 42, 82, 93  | <code>\glstreenamefmt</code> .....                       | 289–298                |
| <code>\GLSplural</code> .....  | 269   | <code>\GLstrLetField</code> .....                        | 26                     |
| <code>\Glsplural</code> .....  | 269   | <code>\glstype</code> .....                              | 49, 51, 60–64, 160–169 |
| <code>\glsplural</code> .....  | 269   | <code>\glsunset</code> .....                             | 38, 50, 85, 86         |
| <code>\glspluralaccessdisplay</code> .....   | 120, 121  | <code>\glswrite</code> .....                             | 35, 97                 |
| <code>\glspluralsuffix</code> .....  | 114, 155, 159   | <code>\glswriteentry</code> .....                        | 8                      |
| <code>\glspostdescription</code> .....   | 148, 283–289  | <code>\Glsxtr</code> .....                               | 43                     |
| <code>\glspostinline</code> .....  | 288   | <code>\glsxtr</code> .....                               | 43                     |
| <code>\glspostlinkhook</code> .  | 48, 50, 60–64, 76, 160–169  | <code>\glsxtr@do@wrglossary</code> .....                 | 8, 11, 12              |
| <code>\glsprestandardsort</code> .....   | 99  | <code>\glsxtr@addloclistfield</code> .....               | 12                     |
| <code>\glsresetentrylist</code> .....  | 105, 116  | <code>\glsxtr@addunused</code> .....                     | 38                     |
| <code>\glssee</code> .....   | 35, 37  | <code>\glsxtr@applyabbrvfmt</code> .....                 | 169                    |
| <code>\glsseeformat</code> .....   | 34, 98, 103   | <code>\glsxtr@applyabbrvstyle</code> . . .               | 153, 155, 172          |
| <code>\glsseelist</code> .....   | 34  | <code>\glsxtr@counterrecord</code> .....                 | 115                    |
| <code>\glssetabbrvfmt</code> .....   | 47, 53, 73, 74,<br>135–137, 139, 140, 146, 159–164, 166, 167  | <code>\glsxtr@doption</code> .....                       | 5, 14, 19, 20          |
| <code>\glssetattribute</code> .....  | 175, 177, 179,<br>180, 182, 183, 185–187, 189, 190, 192–<br>195, 197, 200, 202, 204–207, 209, 210,<br>213, 215, 217–224, 226, 227, 229, 230,<br>232, 234, 236, 239, 240, 242, 244–246,<br>248–250, 252, 253, 255–258, 260, 261, 263 | <code>\glsxtr@fields</code> .....                        | 113                    |
| <code>\glssetcategoryattribute</code> .....  | 82, 94, 96, 131, 132, 134, 147  | <code>\glsxtr@headentry@p</code> .....                   | 77, 78                 |
| <code>\glssetnoexpandfield</code> .....  | 10, 11  | <code>\glsxtr@ifnextpunc</code> .....                    | 152                    |
| <code>\glssettoctitle</code> .....   | 100   | <code>\glsxtr@ifpunctoken</code> .....                   | 152                    |
| <code>\glsshortaccessdisplay</code> .....  | 124, 125  | <code>\glsxtr@indexonly@saveentrycounter</code><br>..... | 11, 12, 21             |
| <code>\glsshortpltok</code> 157, 175, 177–181, 183, 185,<br>190–195, 200, 202, 204–207, 209, 213,<br>215, 217–225, 234, 236, 239–242, 244–<br>246, 248, 249, 253, 255–258, 260, 261, 263 |   | <code>\glsxtr@keylist</code> .....                       | 41, 42                 |
| <code>\glsshortpluralaccessdisplay</code> .....  | 125   | <code>\glsxtr@langtag</code> .....                       | 114                    |
| <code>\glsshorttok</code> .....  | 94, 155–157,<br>175–179, 181, 183, 185, 189–196, 200–<br>202, 204, 205, 207, 209, 210, 213, 215,<br>217–225, 227, 228, 234, 236, 238–246,<br>248, 249, 251, 253, 255–258, 260, 261, 263   | <code>\glsxtr@linkprefix</code> .....                    | 113, 114               |
| <code>\glssubentryitem</code> .....  | 283–288, 298  | <code>\glsxtr@makeglossaries</code> .....                | 97                     |
| <code>\glssymbolaccessdisplay</code> .....   | 122   | <code>\glsxtr@newabbreviation</code> .....               | 95, 155                |
| <code>\glssymbolpluralaccessdisplay</code> .....   | 123   | <code>\glsxtr@next</code> .....                          | 152                    |
|  |   | <code>\glsxtr@org@getgrouptitle</code> .....             | 104                    |
|  |   | <code>\glsxtr@org@newignoredglossary</code> .....        | 28                     |
|  |   | <code>\glsxtr@org@makenoidxglossaries</code> .....       | 39                     |
|  |   | <code>\glsxtr@pluralsuffixes</code> .....                | 113, 114               |
|  |   | <code>\glsxtr@provideignoredglossary</code> .....        | 30                     |
|  |   | <code>\glsxtr@punclist</code> .....                      | 151, 152               |
|  |   | <code>\glsxtr@record</code> .....                        | 9                      |
|  |   | <code>\glsxtr@recordsee</code> .....                     | 10                     |
|  |   | <code>\glsxtr@resource</code> .....                      | 112, 113               |
|  |   | <code>\glsxtr@s@newignoredglossary</code> .....          | 28                     |
|  |   | <code>\glsxtr@s@provideignoredglossary</code> . .        | 30                     |
|  |   | <code>\glsxtr@saveentrycounter</code> .....              | 8, 10, 68              |
|  |   | <code>\glsxtr@setup@record</code> .....                  | 11, 12, 20, 21         |

|   |                                    |   |                                    |
|---|------------------------------------|---|------------------------------------|
| <code>\glxtr@shortcutsval</code> .....            | 113, 114                           | <code>\glxtrfieldlistgadd</code> .....      | 115                                |
| <code>\glxtr@texencoding</code> .....             | 114                                | <code>\glxtrfieldtitlecase</code> .....     | 136–139                            |
| <code>\glxtr@usesee</code> .....                  | 33                                 | <code>\glxtrfieldtitlecasecs</code> .....   | 135                                |
| <code>\glxtr@warnonexistsordo</code> .            | 7, 11, 12, 32, 33                  | <code>\glxtrfieldxifinlist</code> .....     | 117                                |
| <code>\glxtr@writefields</code> .....             | 112                                | <code>\glxtrfirstscfont</code> .....        | 190                                |
| <code>\glxtrabbrvfootnote</code> .....            |                                    | <code>\glxtrfirstsmfont</code> .....        | 203                                |
| ....  | 179–181, 200–202, 213–215, 234–236 | <code>\GlsXtrFmtDefaultOptions</code> ..... | 23                                 |
| <code>\glxtrabbrvpluralsuffix</code> .....        | 114,                               | <code>\GlsXtrFmtField</code> .....          | 23                                 |
| 159, 175, 177, 180, 182, 183, 185, 187,           |                                    | <code>\GlsXtrFormatLocationList</code>      | 45, 47, 295–297                    |
| 190, 204, 217, 238, 248, 250, 253, 259, 261       |                                    | <code>\GLSxtrfull</code> .....              | 15, 16, 273, 274                   |
| <code>\glxtrabbrvtype</code> .....                | 14, 157                            | <code>\Glsxtrfull</code> .....              | 15, 16, 274                        |
| <code>\glxtraddallcrossrefs</code> .....          | 38                                 | <code>\glxtrfull</code> .....               | 15, 273                            |
| <code>\glxtralias</code> .....                    | 68                                 | <code>\Glsxtrfullformat</code> .....        |                                    |
| <code>\glxtrAltTreeIndent</code> .....            | 289                                | ....  | 158, 171, 173, 176, 178, 180, 182, |
| <code>\glxtralttreeInit</code> .....              | 297                                | 184, 186, 188, 191, 193, 195, 196, 198,     |                                    |
| <code>\glxtrAltTreePar</code> .....               | 288                                | 199, 201, 202, 204, 206, 208, 210, 211,     |                                    |
| <code>\glxtrAltTreeSetHangIndent</code> ...       | 289, 298                           | 213, 214, 216, 218, 220, 221, 223, 225,     |                                    |
| <code>\glxtrAltTreeSetSubHangIndent</code> ...    | 298                                | 226, 228, 230, 232, 233, 235, 237, 239,     |                                    |
| <code>\glxtralttreeSubSymbolDescLocation</code>   | 298                                | 240, 243, 246, 249, 251, 254, 257, 259, 262 |                                    |
| <code>\glxtralttreeSymbolDescLocation</code> ..   |                                    | <code>\glxtrfullformat</code> .....         |                                    |
| .....   | 289, 298                           | ....  | 158, 170, 171, 173, 176, 178, 180, |
| <code>\glxtrassignfieldfont</code> .....          | 53–60                              | 182, 184, 186, 188, 191, 192, 195, 196,     |                                    |
| <code>\glxtrautoindex</code> .....                | 142                                | 198–200, 202, 204, 206, 208, 209, 211,      |                                    |
| <code>\glxtrautoindexassignsort</code> ....       | 142, 143                           | 213, 214, 216, 218, 219, 221, 223, 225,     |                                    |
| <code>\glxtrautoindexentry</code> .....           | 142                                | 226, 228, 230, 231, 233, 235, 236, 239,     |                                    |
| <code>\glxtrcat</code> .....                      | 41, 42                             | 240, 243, 245, 248, 251, 254, 256, 259, 262 |                                    |
| <code>\glxtrchecknohyperfirst</code> .....        | 54–56                              | <code>\GLSxtrfullpl</code> .....            | 15, 16, 273, 274                   |
| <code>\glxtrComputeTreeIndent</code> .....        | 298                                | <code>\Glsxtrfullpl</code> .....            | 15, 16, 274                        |
| <code>\glxtrComputeTreeSubIndent</code> .....     | 298                                | <code>\glxtrfullpl</code> .....             | 15, 273, 274                       |
| <code>\Glsxtrdefaultsubsequentfmt</code> ...      | 171, 173                           | <code>\Glsxtrfullplformat</code> .....      |                                    |
| <code>\glxtrdefaultsubsequentfmt</code> ...       | 171, 173                           | ....  | 158, 170, 173, 176, 178, 180, 182, |
| <code>\Glsxtrdefaultsubsequentplfmt</code> .      | 171, 173                           | 184, 186, 188, 191, 193, 195, 196, 198,     |                                    |
| <code>\glxtrdefaultsubsequentplfmt</code> .       | 171, 173                           | 199, 201, 203, 205, 206, 208, 210, 211,     |                                    |
| <code>\GlsXtrDefineAbbreviationShortcuts</code> . | 17                                 | 213, 214, 216, 218, 220, 221, 223, 225,     |                                    |
| <code>\GlsXtrDefineAcShortcuts</code> .....       | 17                                 | 226, 228, 230, 232, 233, 235, 237, 239,     |                                    |
| <code>\GlsXtrDefineOtherShortcuts</code> .....    | 17                                 | 241, 243, 246, 249, 251, 254, 257, 259, 262 |                                    |
| <code>\glxtrdiscardperiod</code> .....            | 149                                | <code>\glxtrfullplformat</code> .....       |                                    |
| <code>\glxtrdisplayendloc</code> .....            | 106                                | .....                                       | 170, 173, 176, 178, 180,           |
| <code>\glxtrdisplayendlohook</code> .....         | 106                                | 182, 184, 186, 188, 191, 193, 195, 196,     |                                    |
| <code>\glxtrdisplaysingleloc</code> .....         | 106, 107                           | 198–200, 202, 204, 206, 208, 209, 211,      |                                    |
| <code>\glxtrdisplaystartloc</code> .....          | 106                                | 213, 214, 216, 218, 219, 221, 223, 225,     |                                    |
| <code>\glxtrdoautoindexname</code> .....          | 69, 141                            | 226, 228, 230, 232, 233, 235, 237, 239,     |                                    |
| <code>\glxtrdopostpunc</code> .....               | 181, 202, 215, 236                 | 240, 243, 245, 248, 251, 254, 256, 259, 262 |                                    |
| <code>\glxtrdowrglossaryhook</code> .....         | 69                                 | <code>\glxtrfullsep</code> .....            |                                    |
| <code>\glxxtremsuffix</code> .....                | 217, 219, 221,                     | 157, 158, 175–178, 181–186, 188, 190–       |                                    |
| 223, 224, 226, 227, 229, 230, 232, 234, 236       |                                    | 199, 201, 203–209, 211–231, 233, 235,       |                                    |
| <code>\GlsXtrEnableEntryCounting</code> .....     | 93                                 | 237, 238, 247–249, 251, 252, 256–258, 262   |                                    |
| <code>\GlsXtrEnableEntryUnitCounting</code> ....  | 82                                 | <code>\glxtrgenabbrvfmt</code> .....        | 48                                 |
| <code>\GlsXtrEnableOnTheFly</code> .....          | 40, 43                             | <code>\glxtrgetgrouptitle</code> .....      | 105                                |



|   |   |  |
|---|---|--|
| <code>\Glsxtrheadfirst</code> .....             | 265   | 198, 199, 201, 203, 208, 209, 211, 212,              |
| <code>\glxtrheadfirst</code> .....              | 265   | 215, 216, 225, 226, 228, 229, 231, 233,              |
| <code>\Glsxtrheadfirstplural</code> .....       | 265   | 235, 237, 241, 243, 251, 254, 260, 262, 280          |
| <code>\glxtrheadfirstplural</code> .....        | 265   | <code>\glxtrinlinefullplformat</code> 158, 161, 162, |
| <code>\Glsxtrheadfull</code> .....              | 265   | 173, 181, 182, 184, 185, 188, 194, 196,              |
| <code>\glxtrheadfull</code> .....               | 265   | 197, 199, 201, 203, 208, 209, 211, 212,              |
| <code>\Glsxtrheadfullpl</code> .....            | 265   | 214, 216, 224, 226, 228, 229, 231, 233,              |
| <code>\glxtrheadfullpl</code> .....             | 265   | 235, 237, 241, 243, 251, 254, 259, 262, 280          |
| <code>\Glsxtrheadlong</code> .....              | 265   | <code>\glxtrininsertinsidefalse</code> .....         |
| <code>\glxtrheadlong</code> .....               | 265   | 175  |
| <code>\Glsxtrheadlongpl</code> .....            | 265   | <code>\glxtrlocationhyperlink</code> .....           |
| <code>\glxtrheadlongpl</code> .....             | 265   | 107  |
| <code>\Glsxtrheadplural</code> .....            | 265   | <code>\glxtrlocrangefmt</code> .....                 |
| <code>\glxtrheadplural</code> .....             | 265   | 106, 107   |
| <code>\Glsxtrheadshort</code> .....             | 265   | <code>\GLSxtrlong</code> .....                       |
| <code>\glxtrheadshort</code> .....              | 265   | 15, 16, 271, 272                                     |
| <code>\Glsxtrheadshortpl</code> .....           | 265   | <code>\Glsxtrlong</code> .....                       |
| <code>\glxtrheadshortpl</code> .....            | 265   | 15, 272  |
| <code>\Glsxtrheadtext</code> .....              | 265   | <code>\glxtrlong</code> .....                        |
| <code>\glxtrheadtext</code> .....               | 265   | 15, 271  |
| <code>\glxtrhyperlink</code> .....              | 71, 107                                     | <code>\glxtrlonghyphen</code> .....                  |
| <code>\glxtrhyphensuffix</code> .....           | 248, 256                                    | 254  |
| <code>\glxtrifcounttrigger</code> .....         | 85, 86                                      | <code>\glxtrlonghyphennoshort</code> .....           |
| <code>\glxtrifemptyglossary</code> .....        | 105, 111, 116                               | 250, 251   |
| <code>\glxtrifhyphenstart</code> .....          | 247, 250, 252, 256, 258                     | <code>\glxtrlonghyphenshort</code> .....             |
| <code>\glxtrifindexing</code> .....             | 69  | 248, 249   |
| <code>\glxtrifinmark</code> .....               | 77–80, 264–266                              | <code>\GLSxtrlongpl</code> .....                     |
| <code>\glxtrifnextpunc</code> .....             | 152, 153                                    | 15, 16, 272, 273                                     |
| <code>\glxtrifperiod</code> .....               | 151   | <code>\Glsxtrlongpl</code> .....                     |
| <code>\glxtrifwasfirstuse</code> .....          | 52, 54, 55, 60–64,                          | 15, 272  |
|   | 66, 96, 150, 160, 163–168, 181, 182, 202,   | <code>\glxtrlongshortdescname</code> .....           |
|   | 215, 236, 240–242, 244, 253, 255, 258, 260  | .... 176, 191, 205, 218, 220, 244, 249, 255          |
| <code>\glxtrindexaliased</code> .....           | 68  | <code>\glxtrlongshortdescsort</code> .....           |
| <code>\glxtrindexseealso</code> .....           | 36, 37                                      | .... 176, 191, 205, 218, 220, 244, 249, 255          |
| <code>\glxtrinitwrgloss</code> .....            | 51  | <code>\glxtrmarkhook</code> .....                    |
| <code>\glxtrinitwrglossbeforefalse</code> ..... | 50  | 264  |
| <code>\glxtrinitwrglossbeforetrue</code> .....  | 50  | <code>\glxtrnewabbrevpresetkeyhook</code> .....      |
| <code>\Glsxtrinlinefullformat</code> ....       | 158, 160,                                   | 156  |
|   | 173, 181, 183, 184, 186, 188, 194, 196,     | <code>\glxtrnewnumber</code> .....                   |
|   | 197, 199, 201, 203, 208, 209, 211, 212,     | 16   |
|   | 214, 216, 225, 226, 228, 229, 231, 233,     | <code>\glxtrnewsymbol</code> .....                   |
|   | 235, 237, 241, 243, 251, 254, 260, 262, 280 | 16   |
| <code>\glxtrinlinefullformat</code> .....       | 158, 159, 161,                              | <code>\glxtrNoGlossaryWarning</code> .....           |
|   | 173, 180, 182, 183, 185, 187, 194, 195,     | 18, 108  |
|   | 197, 199, 201, 203, 208, 209, 211, 212,     | <code>\GlsXtrNoGlsWarningAutoMake</code> .....       |
|   | 214, 216, 224, 226, 227, 229, 231, 233,     | 111  |
|   | 235, 237, 241, 243, 251, 254, 259, 262, 279 | <code>\GlsXtrNoGlsWarningBuildInfo</code> .....      |
| <code>\Glsxtrinlinefullplformat</code> ..       | 159, 162,                                   | 112  |
|   | 173, 181, 183, 184, 186, 188, 194, 196,     | <code>\GlsXtrNoGlsWarningCheckFile</code> .....      |
|   |   | 111  |
|   |   | <code>\GlsXtrNoGlsWarningEmptyMain</code> ..         |
|   |   | 111, 112   |
|   |   | <code>\GlsXtrNoGlsWarningEmptyNotMain</code> ...     |
|   |   | 111  |
|   |   | <code>\GlsXtrNoGlsWarningEmptyStart</code> .....     |
|   |   | 111  |
|   |   | <code>\GlsXtrNoGlsWarningHead</code> .....           |
|   |   | 111  |
|   |   | <code>\GlsXtrNoGlsWarningMisMatch</code> .....       |
|   |   | 112  |
|   |   | <code>\GlsXtrNoGlsWarningNoOut</code> .....          |
|   |   | 112  |
|   |   | <code>\GlsXtrNoGlsWarningTail</code> .....           |
|   |   | 112  |
|   |   | <code>\glxtronlydescname</code> .....                |
|   |   | 263  |
|   |   | <code>\glxtronlydescsort</code> .....                |
|   |   | 263  |
|   |   | <code>\glxtronlysuffix</code> .....                  |
|   |   | 261  |
|   |   | <code>\glxtrorg@ifKV@glslink@hyper</code> .....      |
|   |   | 48   |
|   |   | <code>\glxtrorglong</code> .....                     |
|   |   | 155, 176, 250  |
|   |   | <code>\glxtrorgshort</code> .....                    |
|   |   | 155, 176   |
|   |   | <code>\GLSxtrp</code> .....                          |
|   |   | 77   |
|   |   | <code>\Glsxtrp</code> .....                          |
|   |   | 77   |
|   |   | <code>\glxtrp</code> .....                           |
|   |   | 76, 78   |

|  |  |   |   |
|--|--|---|---|
| <code>\glxstrparen</code> .....                | 157, 158, 175–178,<br>181–186, 188, 190–194, 196–199, 201,<br>203–209, 211–226, 228–231, 233, 235,<br>237, 238, 247–249, 251, 252, 256–258, 262  | <code>\glxstrshorthyphenlong</code> .....   | 256, 257  |
| <code>\Glsxtrpl</code> .....                   | 43   | <code>\glxstrshortlongdescname</code> ..... | 178, 193, 207, 222, 223, 246, 257, 260                                  |
| <code>\glxstrpl</code> .....                   | 43   | <code>\glxstrshortlongdescsort</code> ..... | 178, 193, 207, 222, 223, 246, 257, 260                                  |
| <code>\glxstrpostdescription</code> .....      | 134, 148, 288  | <code>\GLSxtrshortpl</code> .....           | 15, 16, 267, 268  |
| <code>\glxstrposthyphenlong</code> .....       | 258, 260   | <code>\Glsxtrshortpl</code> .....           | 15, 268   |
| <code>\glxstrposthyphenshort</code> .....      | 253, 255   | <code>\glxstrshortpl</code> .....           | 15, 267   |
| <code>\glxstrposthyphensubsequent</code> ..... | 253, 255, 258, 260   | <code>\glxstrsmfont</code> .....            | 203   |
| <code>\glxstrpostlink</code> .....             | 149  | <code>\glxstrsmsuffix</code> .....          | 204, 206, 207, 209, 210, 212, 214, 216                                  |
| <code>\glxstrpostlinkendsentence</code> .....  | 149  | <code>\Glsxtrsubsequentfmt</code> .....     | 170, 173, 187, 197, 199,<br>210, 212, 227, 229, 231, 232, 250, 254, 259 |
| <code>\glxstrpostlinkhook</code> .....         | 149  | <code>\glxstrsubsequentfmt</code> .....     | 170, 173, 187, 197, 198,<br>210, 212, 227, 229, 231, 232, 250, 254, 259 |
| <code>\glxstrpostlocalreset</code> .....       | 81, 83, 91   | <code>\Glsxtrsubsequentplfmt</code> .....   | 169, 173, 187, 197, 199,<br>211, 212, 227, 229, 231, 233, 250, 254, 259 |
| <code>\glxstrpostlocalunset</code> .....       | 81, 83, 91   | <code>\glxstrsubsequentplfmt</code> .....   | 169, 173, 187, 197, 198,<br>210, 212, 227, 229, 231, 232, 250, 254, 259 |
| <code>\glxstrpostlongdescription</code> .....  | 28   | <code>\glxstrsupplocationurl</code> .....   | 107   |
| <code>\GlsXtrPostNewAbbreviation</code> .....  | 157,<br>173, 175, 177, 179–181, 183, 185–190,<br>192–195, 197, 200, 202, 204–207, 209,<br>210, 213, 215, 217–224, 226, 227, 229,<br>230, 232, 234, 236, 239–242, 244–246,<br>248–250, 252, 253, 255–258, 260, 261, 263 | <code>\glxstrtagfont</code> .....           | 148   |
| <code>\glxstrpostreset</code> .....            | 81, 83, 91   | <code>\Glsxtrtitlefirst</code> .....        | 265, 266, 277   |
| <code>\glxstrpostunset</code> .....            | 81, 83, 91   | <code>\glxstrtitlefirst</code> .....        | 265, 266, 277   |
| <code>\glxstrprotectlinks</code> .....         | 70–72  | <code>\Glsxtrtitlefirstplural</code> ....   | 265, 266, 278   |
| <code>\GlsXtrRecordCounter</code> .....        | 9  | <code>\glxstrtitlefirstplural</code> ....   | 265, 266, 278   |
| <code>\glxstrregularfont</code> .....          | 47, 48, 53   | <code>\Glsxtrtitlefull</code> .....         | 265, 266, 280   |
| <code>\glxstrresourcecount</code> .....        | 113  | <code>\glxstrtitlefull</code> .....         | 265, 266, 279   |
| <code>\glxstrresourcefile</code> .....         | 113  | <code>\Glsxtrtitlefullpl</code> .....       | 265, 266, 280   |
| <code>\glxstrrestoremarkhook</code> .....      | 264  | <code>\glxstrtitlefullpl</code> .....       | 265, 266, 280   |
| <code>\glxstrscfont</code> .....               | 190  | <code>\Glsxtrtitlelong</code> .....         | 265, 266, 278, 279  |
| <code>\glxstrscsuffix</code> .....             | 191, 192, 194, 195, 197, 198, 200, 202   | <code>\glxstrtitlelong</code> .....         | 265, 266, 278   |
| <code>\GlsXtrSetActualChar</code> .....        | 145  | <code>\Glsxtrtitlelongpl</code> .....       | 265, 266, 279   |
| <code>\glxstrsetaliasnoindex</code> .....      | 11, 12, 68   | <code>\glxstrtitlelongpl</code> .....       | 265, 266, 279   |
| <code>\GlsXtrSetEncapChar</code> .....         | 145  | <code>\Glsxtrtitleplural</code> .....       | 265, 266, 277   |
| <code>\GlsXtrSetEscChar</code> .....           | 145  | <code>\glxstrtitleplural</code> .....       | 265, 266, 276, 277  |
| <code>\glxstrsetfieldifexists</code> .....     | 25, 26   | <code>\Glsxtrtitleshort</code> .....        | 265, 266, 275   |
| <code>\GlsXtrSetLevelChar</code> .....         | 145  | <code>\glxstrtitleshort</code> .....        | 265, 266, 275   |
| <code>\glxstrsetpopts</code> .....             | 76   | <code>\Glsxtrtitleshortpl</code> .....      | 265, 266, 276   |
| <code>\glxstrsetupfulldefs</code> .....        | 160–162, 182, 202, 215, 236  | <code>\glxstrtitleshortpl</code> .....      | 265, 266, 275   |
| <code>\GLSxtrshort</code> .....                | 15, 16, 80, 266, 267   | <code>\Glsxtrtitletext</code> .....         | 265, 266, 276   |
| <code>\Glsxtrshort</code> .....                | 15, 267  | <code>\glxstrtitletext</code> .....         | 265, 266, 276   |
| <code>\glxstrshort</code> .....                | 15, 266, 267   | <code>\glxstrtreetopindent</code> .....     | 289, 297  |
| <code>\glxstrshortdescname</code> ...          | 185, 195, 209, 225   | <code>\glxstrundefaction</code> ..          | 7, 11, 12, 21, 29, 31–33  |
| <code>\glxstrshorthyphen</code> .....          | 259  | <code>\glxstrundeftag</code> .....          | 21, 102, 103  |
|  |  | <code>\glxstrunsrtdo</code> .....           | 117   |



|   |   |
|---|---|
| <code>\GlsXtrUseAbbrStyleFmts</code> .....      | 177,<br>179, 185, 187, 189, 190, 192, 193, 205,<br>207, 219, 220, 222, 224, 230, 234, 242,<br>244–246, 249, 250, 252, 255, 257, 260, 263        |
| <code>\GlsXtrUseAbbrStyleSetup</code> .....     | .... 184, 186, 188, 189, 198, 212, 230, 234   |
| <code>\glxtruserfield</code> .....              | 238   |
| <code>\glxtruserparen</code> .....              | 239–246   |
| <code>\glxtrusersuffix</code> .....             | 239, 240, 242, 245  |
| <code>\glxtruseseealsoformat</code> .....       | 34, 35  |
| <code>\glxtruseseeeformat</code> .....          | 34  |
| <code>\GlsXtrWarnDeprecatedAbbrStyle</code>     | 153, 174  |
| <code>\GlsXtrWarning</code> .....               | 41, 42  |
| <code>\glxtrword</code> .....                   | 154   |
| <code>\glxtrwordsep</code> ..                   | 154, 247, 250, 252, 256, 258  |
| <b>H</b>  |   |
| <code>\hangindent</code> .....                  | 289, 297  |
| <code>\hbox</code> .....                        | 283   |
| <code>\hfill</code> .....                       | 283   |
| <code>\href</code> .....                        | 71  |
| <code>\hsize</code> .....                       | 44  |
| <code>\hss</code> .....                         | 283   |
| <code>\hyperlink</code> .....                   | 71  |
| <code>\hyperpage</code> .....                   | 141   |
| <code>\hyperref</code> .....                    | 70, 107   |
| <code>hyperref package</code> .....             | 72, 141, 263, 275   |
| <b>I</b>  |   |
| <code>\if</code> .....                          | 40  |
| <code>\if@glxtr@autoseeindex</code> ....        | 19, 20, 33, 36  |
| <code>\if@glxtr@format@override</code> .....    | 142   |
| <code>\if@glxtr@docdefrestricted</code> .....   | 39  |
| <code>\if@glxtr@indexcrossrefs</code> .....     | 13, 38  |
| <code>\ifblank</code> .....                     | 22, 41, 42, 97  |
| <code>\ifcase</code> .....                      | 7, 11, 17, 18, 39, 50, 101  |
| <code>\ifcsdef</code> .....                     | 7, 21, 27–31,<br>65, 76–80, 88, 100, 104, 105, 117, 136,<br>137, 139, 140, 150, 153, 169, 173, 283–287  |
| <code>\ifcsstring</code> .....                  | 21, 131, 172  |
| <code>\ifcsundef</code> .....                   | 27,<br>29–31, 39, 43, 46, 72, 83, 88–92, 104,<br>105, 108, 117, 131, 172–174, 282, 290, 297   |
| <code>\ifcsvoid</code> .....                    | 37, 130   |
| <code>\ifdef</code> .....                       | 12, 16, 20, 23, 25, 32,<br>35, 44, 67, 71, 77–79, 100, 102, 103, 114,<br>133, 134, 145, 148, 238, 275–280, 283, 288                             |
| <code>\ifdefempty</code> .                      | 7, 8, 25, 29–31, 33, 34, 51, 52,<br>82, 94, 97, 100, 106, 116, 118, 147, 154, 169   |
| <code>\ifdefequal</code> .....                  | 112, 118  |
| <code>\ifdefstring</code> .....                 | 6, 142, 147   |
| <code>\ifdefvoid</code> ...                     | 33, 36–38, 71, 88, 104, 107, 118  |
| <code>\ifdim</code> .....                       | 44, 45, 96, 290–297   |
| <code>\IfFileExists</code> ....                 | 18, 108, 111, 112, 114, 282   |
| <code>\ifglossaryexists</code> .....            | 33  |
| <code>\ifglssacronym</code> .....               | 14, 111   |
| <code>\ifglssacrshortcuts</code> .....          | 16  |
| <code>\ifglssautomake</code> .....              | 100, 111, 114   |
| <code>\ifglssentrycounter</code> .....          | 26  |
| <code>\ifglssentryexists</code> .....           | .... 31, 32, 41, 42, 45, 53, 131, 148, 149  |
| <code>\ifglssfieldeq</code> .....               | 130   |
| <code>\ifglsshasfield</code> .....              | 23, 68, 238   |
| <code>\ifglsshaslong</code> .....               | 87, 88  |
| <code>\ifglsshasparent</code> .....             | 118, 290–293  |
| <code>\ifglsshasshort</code> .....              | 47, 48, 53  |
| <code>\ifglsshasymbol</code> .....              | 150, 289  |
| <code>\ifglssindexonlyfirst</code> .....        | 69  |
| <code>\ifglssnogroupskip</code> .....           | 284–288, 299  |
| <code>\ifglssnonumberlist</code> .....          | 47  |
| <code>\ifglsssanitizesort</code> .....          | 99  |
| <code>\ifglsssubentrycounter</code> .....       | 26  |
| <code>\ifglssused</code> .....                  | 38, 66,<br>69, 84, 93, 96, 169, 290, 291, 293, 295, 296   |
| <code>\ifglssxindy</code> .....                 | 108, 110  |
| <code>\ifglssxtrinitwrglossbefore</code> .....  | 50–52   |
| <code>\ifglssxtrininsertinside</code>           | 163–169, 171, 176,<br>178, 180–188, 191–206, 208–216, 218–<br>221, 223–233, 235, 237, 239–241, 243,<br>245–247, 250, 252–254, 256, 258–260, 262 |
| <code>\ifHy@hyperindex</code> .....             | 141   |
| <code>\ifinlistcs</code> .....                  | 24, 39  |
| <code>\ifKV@glsslink@hyper</code> .....         | 48, 51, 52  |
| <code>\ifKV@glsslink@local</code> .....         | 49  |
| <code>\ifKV@glsslink@noindex</code> .....       | 8, 10, 68   |
| <code>\ifnum</code> .....                       | 12, 13, 84, 92, 93, 104, 113, 298   |
| <code>\ifstrequal</code> .....                  | 18  |
| <code>\ifthenelse</code> .....                  | 111   |
| <code>\IfTrackedLanguageFileExists</code> ..... | 281   |
| <code>\ifundef</code> .....                     | 72, 97, 147, 148  |
| <code>\ifx</code> .....                         | 8, 9, 44, 45, 100, 106, 107,<br>114, 142, 143, 145, 152, 154, 156, 247, 299   |
| <code>\immediate</code> .....                   | 84, 92, 108, 114  |
| <code>\index</code> .....                       | 142   |
| <code>\indexspace</code> .....                  | 299   |
| <code>\input</code> .....                       | 281   |
| <code>\inputencodingname</code> .....           | 114   |
| <code>\istfilename</code> .....                 | 97  |
| <code>\item</code> .....                        | 110, 283  |

|   |   |
|---|---|
| <b>J</b>                                      |   |
| <code>\jobname</code> .....                   | 108, 110–114  |
| <b>K</b>                                      |   |
| <code>\key@ifundefined</code> .               | 10, 11, 21, 22, 64, 116, 118  |
| <code>\KV@glslink@hyperfalse</code> .....     | 53, 66, 67, 72  |
| <code>\KV@glslink@noindexfalse</code> .....   | 67  |
| <code>\KV@glslink@noindextrue</code> .....    | 67, 72  |
| <b>L</b>                                      |   |
| <code>\LaTeX</code> .....                     | 109–111   |
| <code>\leaders</code> .....                   | 283   |
| <code>\leavevmode</code> .....                | 28, 51  |
| <code>\let</code> .....                       | 5, 7–9, 11, 12, 14–17, 19–21, 23,<br>27, 28, 39, 40, 43–46, 48–70, 72, 73, 76,<br>82, 83, 91–105, 112, 114, 116, 118, 136–<br>144, 147, 148, 152–156, 159–169, 171,<br>173, 182, 202, 215, 236, 264–266, 288, 290                               |
| <code>\letabbreviationstyle</code> ..         | 181, 183, 184,<br>186, 188, 189, 195, 196, 208, 210, 225, 227   |
| <code>\letcs</code> .....                     | 22, 25,<br>33, 34, 38, 64, 102–104, 117, 118, 136–140   |
| <code>\levelchar</code> .....                 | 145   |
| <code>\listadd</code> .....                   | 88  |
| <code>\listbreak</code> .....                 | 147   |
| <code>\listcsadd</code> .....                 | 24  |
| <code>\listcseadd</code> .....                | 24, 89  |
| <code>\listcsgadd</code> .....                | 24, 39  |
| <code>\listcsxadd</code> .....                | 24, 89  |
| <code>\loadglsentries</code> .....            | 39, 109   |
| <code>\long</code> .....                      | 28  |
| <b>M</b>                                      |   |
| <code>\MakeAcronymsAbbreviations</code> ..... | 96  |
| <code>\makeatletter</code> .....              | 108, 112, 144   |
| <code>\makeatother</code> .....               | 144   |
| <code>\makebox</code> .....                   | 283, 298  |
| <code>\makefirstuc</code> .....               | 148   |
| <code>makeglossaries</code> .....             | 101   |
| <code>\makeglossaries</code> .....            | 97, 108, 110–112, 115   |
| <code>\makeglossary</code> .....              | 97  |
| <code>makeindex</code> .....                  | 300   |
| <code>makeindex</code> .....                  | 96  |
| <code>\makenoidxglossaries</code> .....       | 110   |
| <code>\MakeTextUppercase</code> .....         | 265   |
| <code>\MakeUppercase</code> .....             | 265, 266  |
| <code>\markboth</code> .....                  | 264   |
| <code>\markright</code> .....                 | 264   |
| <code>\maxdimen</code> .....                  | 44, 45  |
| <code>\mbox</code> .....                      | 298   |
| <code>\medskip</code> .....                   | 111, 112, 117   |
| <code>\MessageBreak</code> ....               | 39, 43, 84, 93, 99, 100, 172  |
| <code>mfistuc package</code> .....            | 147   |
| <code>\mfistucMakeUppercase</code> .....      | 53–<br>64, 66, 74, 75, 77, 80, 87, 95, 120–129,<br>138, 139, 161, 162, 164, 165, 167, 169–171   |
| <code>\mfu@checkword@arg</code> .....         | 147, 148  |
| <code>\mfu@checkword@do</code> .....          | 148   |
| <b>N</b>                                      |   |
| <code>\NeedsTeXFormat</code> .....            | 5, 282  |
| <code>\new@glossaryentry</code> .....         | 40, 99  |
| <code>\new@ifnextchar</code> .                | 65, 66, 86, 87, 151, 159–168  |
| <code>\newabbr</code> .....                   | 15, 16  |
| <code>\newabbreviation</code> .....           | 15, 16  |
| <code>\newabbreviationhook</code> .....       | 157   |
| <code>\newabbreviationstyle</code> ..         | 175–179, 181,<br>183–196, 198, 200, 201, 204–207, 209,<br>210, 212, 213, 215, 217–220, 222–225,<br>227, 228, 230, 232, 234, 236, 238, 240–<br>246, 248–251, 253, 255–258, 260, 261, 263   |
| <code>\newacronym</code> .....                | 94, 95  |
| <code>\newacronymhook</code> .....            | 94  |
| <code>\newacronymstyle</code> .....           | 95, 96  |
| <code>\newcommand</code> ..                   | 5–7, 9–34, 36, 38–43, 45–48,<br>50, 52, 53, 64–72, 75–84, 86–96, 100–<br>104, 106, 107, 109–135, 141–155, 157–<br>169, 171–174, 176, 178, 179, 185, 190,<br>203, 204, 217, 237, 238, 247–249, 252,<br>253, 255, 257, 258, 261–282, 288–290, 297 |
| <code>\newcount</code> .....                  | 12, 113   |
| <code>\newentry</code> .....                  | 16  |
| <code>\newglossary</code> .....               | 14, 97  |
| <code>\newglossaryentry</code> .....          | ..... 16, 39, 40, 82, 90, 94, 133, 134, 157   |
| <code>\newglossaryentry options</code>        |   |
| alias .....                                   | 13, 33, 35–38   |
| desc .....                                    | 123, 128  |
| descplural .....                              | 124, 128  |
| first ....                                    | 70, 121, 127, 175, 270, 271, 277, 300   |
| firstplural                                   | 121, 122, 127, 175, 270, 271, 277, 300  |
| loclist .....                                 | 24  |
| long .....                                    | 126, 129, 278   |
| longplural .....                              | 126, 129, 279   |
| name .....                                    | 119, 126, 127, 142  |
| plural .....                                  | 120, 121, 127, 175, 269, 276  |
| see .....                                     | 13, 20, 33, 35, 38, 40, 97  |
| seealso .....                                 | 13, 33–35, 37, 38, 311  |
| short .....                                   | 125, 129, 154   |

|                               |   |  |  |
|-------------------------------|---|--|--|
| shortplural                   | 125, 129, 154   |  |  |
| symbol                        | 122, 128  |  |  |
| symbolplural                  | 122, 123, 128   |  |  |
| text                          | 70, 120, 127, 175, 177, 268, 269, 276   |  |  |
| \newif                        | 50, 141, 175  |  |  |
| \newlength                    | 289   |  |  |
| \newnum                       | 16  |  |  |
| \newrobustcmd                 | 23, 25, 26, 34, 35, 65, 66, 76, 77, 86, 87, 104, 117, 147, 148, 159–168, 247, 264, 267–274, 290–296   |  |  |
| \newsym                       | 16  |  |  |
| \newterm                      | 133   |  |  |
| \newtoks                      | 153   |  |  |
| \newwrite                     | 97  |  |  |
| \NoCaseChange                 | 77–80, 266–274  |  |  |
| \noexpand                     | 9, 10, 18, 34, 36, 37, 94, 108, 112, 116, 143, 144, 157, 282  |  |  |
| \nofiles                      | 111   |  |  |
| \noindent                     | 111, 112  |  |  |
| \nopostdesc                   | 28, 41, 42, 101, 134  |  |  |
| \nr                           | 7, 11, 13, 16–18, 50, 101   |  |  |
| \ns@GLSxtrfull                | 160   |  |  |
| \ns@GLsxtrfull                | 160   |  |  |
| \ns@glsxtrfull                | 159   |  |  |
| \ns@GLSxtrfullpl              | 162   |  |  |
| \ns@GLsxtrfullpl              | 161   |  |  |
| \ns@glsxtrfullpl              | 161   |  |  |
| \ns@GLSxtrlong                | 165   |  |  |
| \ns@GLsxtrlong                | 164   |  |  |
| \ns@glsxtrlong                | 164   |  |  |
| \ns@GLSxtrlongpl              | 168   |  |  |
| \ns@GLsxtrlongpl              | 168   |  |  |
| \ns@glsxtrlongpl              | 167   |  |  |
| \ns@GLSxtrshort               | 163   |  |  |
| \ns@GLsxtrshort               | 163   |  |  |
| \ns@glsxtrshort               | 162   |  |  |
| \ns@GLSxtrshortpl             | 167   |  |  |
| \ns@GLsxtrshortpl             | 166   |  |  |
| \ns@glsxtrshortpl             | 166   |  |  |
| \null                         | 18  |  |  |
| \number                       | 89–92, 112  |  |  |
| \numexpr                      | 89, 92  |  |  |
| <b>O</b>                      |   |  |  |
| \or                           | 7, 11, 12, 17, 39, 40, 50   |  |  |
| \org@glossaryentrynumbers     | 45, 101   |  |  |
| \org@glossarytitle            | 100   |  |  |
| \org@ifKV@glslink@hyper       | 51, 52  |  |  |
| <b>P</b>                      |   |  |  |
| \p@gl@s@hyp@opt               | 70  |  |  |
| package options:              |   |  |  |
| abbreviations                 | 14  |  |  |
| accsupp                       | 18, 119   |  |  |
| acronym                       | 14  |  |  |
| automake                      | 100, 109, 114   |  |  |
| true                          | 114   |  |  |
| autoseeindex                  | 20  |  |  |
| false                         | 19  |  |  |
| debug                         |   |  |  |
| showtargets                   | 71  |  |  |
| docdef                        | 12, 39, 82, 90  |  |  |
| false                         | 39  |  |  |
| restricted                    | 13  |  |  |
| true                          | 40  |  |  |
| nonumberlist                  | 45  |  |  |
| nopostdot                     |   |  |  |
| false                         | 14  |  |  |
| numbers                       | 16  |  |  |
| record                        | 7, 11, 48, 112, 309   |  |  |
| alsoindex                     | 9   |  |  |
| shortcuts                     | 16  |  |  |
| ac                            | 16  |  |  |
| all                           | 16  |  |  |
| false                         | 16  |  |  |
| none                          | 16  |  |  |
| true                          | 16  |  |  |
| style                         | 19  |  |  |
| stylemods                     | 19  |  |  |
| symbols                       | 16, 134   |  |  |
| undefaction                   | 31, 32  |  |  |
| error                         | 6   |  |  |
| warn                          | 6   |  |  |
| xindy                         | 35  |  |  |
| \PackageError                 | 6, 9, 18, 20, 39, 43, 65, 66, 68, 76, 77, 82, 83, 90, 92, 93, 95, 97, 99, 105, 114, 117, 172–174, 282 |  |  |
| \PackageWarning               | 14  |  |  |
| \PackageWarningNoLine         | 14  |  |  |
| \pageref                      | 26  |  |  |
| \par                          | 111, 112, 288, 289, 298   |  |  |
| \parindent                    | 289, 298  |  |  |
| \PassOptionsToPackage         | 5   |  |  |
| \preglossarypreamble          | 27  |  |  |
| \preto                        | 67  |  |  |
| \print@noop@unsrtglossaryunit | 10, 11  |  |  |
| \print@op@unsrtglossaryunit   | 12  |  |  |
| \printabbreviations           | 14  |  |  |



|                              |                           |                         |                          |
|------------------------------|---------------------------|-------------------------|--------------------------|
| \toks@ .....                 | 107, 145, 146             | \write .....            | 35, 84, 92, 97, 108, 114 |
| tracklang package .....      | 114                       |                         |                          |
| <b>U</b>                     |                           | <b>X</b>                |                          |
| \undef .....                 | 11, 12, 147               | \x .....                | 107                      |
| \underline .....             | 148                       | \xcapitalisewords ..... | 135                      |
| \unskip .....                | 28, 39, 283               | \xdef .....             | 101, 116                 |
| \usepackage .....            | 110–112                   | \xifinlist .....        | 88                       |
| <b>V</b>                     |                           | \xifinlistcs .....      | 24                       |
| \val .....                   | 7, 11, 13, 16–18, 50, 101 | xindy .....             | 300                      |
| <b>W</b>                     |                           | xindy .....             | 96                       |
| \warn@nomakeglossaries ..... | 98                        | xkeyval package .....   | 5                        |
| \warn@noprintglossary .....  | 98, 101                   | \XKV@checkchoice .....  | 47                       |
|                              |                           | \XKV@plfalse .....      | 47                       |
|                              |                           | \XKV@resa .....         | 47                       |
|                              |                           | \XKV@sttrue .....       | 47                       |