glossaries-extra.sty v1.7: documented code

Nicola L.C. Talbot

Dickimaw Books
http://www.dickimaw-books.com/

2025-05-14

This is the documented code for the glossaries-extra package. See glossaries-extra-manual.pdf for the user manual.

Contents

1   Main Package Code (glossaries-extra.sty)

1.1 Package Initialisation and Options

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:
```
\DeclareRelease{v1.48}{2021-11-22}{glossaries-extra-2021-11-22.sty}
\DeclareCurrentRelease{v1.7}{2025-05-14}
```

Declare package:
```
\ProvidesPackage{glossaries-extra}[2025/05/14 v1.7 (NLCT)]
```

Requires xkeyval to define package options.
```
\RequirePackage{xkeyval}
```

Requires etoolbox package.

```
\RequirePackage{etoolbox}
```

Has glossaries already been loaded?

```
\@ifpackageloaded{glossaries}
{%
```

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.

```
    \newcommand{\glsxtr@dooption}[1]{\setupglossaries{#1}}%
    \let\@glsxtr@declareoption\@gls@declareoption
}
{%
```

Not already loaded, so pass options to glossaries.

```
    \newcommand{\glsxtr@dooption}[1]{%
      \PassOptionsToPackage{#1}{glossaries}%
    }%
```

Set the defaults.

```
    \PassOptionsToPackage{toc}{glossaries}
    \PassOptionsToPackage{nopostdot}{glossaries}
    \PassOptionsToPackage{noredefwarn}{glossaries}
    \@ifpackageloaded{polyglossia}%
    {}%
    {%
      \@ifpackageloaded{babel}%
      {\PassOptionsToPackage{translate=babel}{glossaries}}%
      {}%
    }%
    \newcommand*{\@glsxtr@declareoption}[2]{%
      \DeclareOptionX{#1}{#2}%
      \DeclareOption{#1}{#2}%
    }
  }
```

Declare package options.

\glsxtrundefaction 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.

```
\newcommand*{\glsxtrundefaction}[2]{%
  \@glsxtrundeftag\PackageError{glossaries-extra}{#1}{#2}%
}
```

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

```
\newcommand*{\glsxtr@warnonexistsordo}[1]{}
```

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

```
\newcommand*{\glsxtrundeftag}{??}
\newcommand*{\@glsxtrundeftag}{}
```

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.

This is how \glsxtrundefaction should behave if undefaction=warn is set.

```
\newcommand*{\@glsxtr@warn@undefaction}[2]{%
  \@glsxtrundeftag\GlossariesExtraWarning{#1}%
}
```

This is how \glsxtrundefaction should behave if undefaction=error is set.

```
\newcommand*{\@glsxtr@err@undefaction}[2]{%
  \@glsxtrundeftag\PackageError{glossaries-extra}{#1}{#2}%
}
```

This is how \glsxtr@warnonexistsordo should behave if undefaction=warn is set.

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

```
\newcommand*{\@glsxtr@redef@forglsentries}{}
```

```
\newcommand*{\@glsxtr@do@redef@forglsentries}{%
  \renewcommand*{\forglsentries}[3][\glsdefaulttype]{%
    \protected@edef\@@glo@list{\csname glolist@##1\endcsname}%
    \ifdefstring{\@@glo@list}{,}%
    {%
      \GlossariesExtraWarning{\string\forglsentries:
        No entries defined in glossary '##1'}%
    }%
    {%
      \@for##2:=\@@glo@list\do
      {%
        \ifdefempty{##2}{}{##3}%
      }%
    }%
  }%
}%
```

```
\define@choicekey{glossaries-extra.sty}{undefaction}%
 [\glsxtr@undefaction@val\glsxtr@undefaction@nr]%
 {warn,error}%
 {%
   \ifcase\glsxtr@undefaction@nr\relax
     \let\glsxtrundefaction\@glsxtr@warn@undefaction
     \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
     \let\@glsxtr@redef@forglsentries\@glsxtr@do@redef@forglsentries
```

```
      \or
        \let\glsxtrundefaction\@glsxtr@err@undefaction
        \let\glsxtr@warnonexistsordo\@gobble
        \let\@glsxtr@redef@forglsentries\relax
      \fi
    }
```

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

\@glsxtr@record Does nothing by default.

```
\newcommand*{\@glsxtr@record}[3]{}
```

\glsxtr@recordsee Does nothing by default.

```
\newcommand*{\glsxtr@recordsee}[2]{}
```

\@glsxtr@defaultnumberformat

```
\newcommand*{\@glsxtr@defaultnumberformat}{glsnumberformat}%
```

\GlsXtrSetDefaultNumberFormat

```
\newcommand*{\GlsXtrSetDefaultNumberFormat}[1]{%
  \renewcommand*{\@glsxtr@defaultnumberformat}{#1}%
}%
```

The record option is somewhat problematic. On the first LaTeX run the
entries aren't defined. This isn't as straight-forward as commands like \cite
since attributes associated with the entry's category may switch off the indexing
or the entry's glossary type might require a particular counter. This kind of
information can't be determined until the entry has been defined. So there are
two different commands here. One that's used if the entry hasn't been defined,
which tries to use sensible defaults, and one which is used when the entry has
been defined.

\@glsxtr@do@record@wrglossary The record=only option sets \@@do@wrglossary to this command, which means
it's done within \glsadd and \@gls@link, and so is only done if the entry exists.

```
\newcommand*{\@glsxtr@do@record@wrglossary}[1]{%
  \begingroup
    \ifKV@glslink@noindex
    \else

      \protected@edef\@gls@label{\glsdetoklabel{#1}}%
      \let\glslabel\@gls@label
      \glswriteentry{#1}%
      {%
        \ifdefempty{\@glsxtr@thevalue}%
        {%
          \ifx\@glsxtr@org@theHvalue\@glsxtr@theHvalue
          \else
            \let\theHglsentrycounter\@glsxtr@theHvalue
```

5

```
            \fi
            \glsxtr@saveentrycounter
            \let\@@do@@wrglossary\@glsxtr@dorecord
         }%
         {%
            \let\theglsentrycounter\@glsxtr@thevalue
            \let\theHglsentrycounter\@glsxtr@theHvalue
            \let\@@do@@wrglossary\@glsxtr@dorecordnodefer
         }%
         \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
            \glsxtr@@do@wrglossary{#1}%
         \else
            \@@glsxtrwrglossmark
```

Increment associated counter.

```
            \glsxtr@inc@wrglossaryctr{#1}%
            \@@do@@wrglossary
         \fi
      }%
   \fi
 \endgroup
}
```

xtr@do@alsoindex@wrglossary The record=alsoindex option needs to both record and index.

```
\newcommand*{\glsxtr@do@alsoindex@wrglossary}[1]{%
  \glsxtr@@do@wrglossary{#1}%
  \@glsxtr@do@record@wrglossary{#1}%
}
```

\@@glsxtr@record The record=only option sets \@glsxtr@record to this. This performs the record-
ing if the entry *doesn't exist* and is done at the start of \@gls@field@link and
commands like \@gls@ (before the existence test). This means that it disregards
the wrgloss key.

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
second argument is the entry's label. The third argument is the key family
(glslink in most cases, glossadd for \glsadd).

```
\newcommand*{\@@glsxtr@record}[3]{%
```

Save the label in case it's needed. This needs to be outside the existence check
to allow the post-link hook to reference it.

```
\protected@edef\@gls@label{\glsdetoklabel{#2}}%
\let\glslabel\@gls@label
\ifglsentryexists{#2}{}%
{%
  \@@glsxtrwrglossmark
  \begingroup
    \let\@glsnumberformat\@glsxtr@defaultnumberformat
    \def\@glsxtr@thevalue{}%
```

6

```
\def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
\let\@glsxtr@org@theHvalue\@glsxtr@theHvalue
```

Entry hasn't been defined, so we'll have to assume it's `\glscounter` by default.

```
\let\@gls@counter\glscounter
```

Unless the equations option is on and this is inside a numbered maths environment.

```
\if@glsxtr@equations
  \@glsxtr@use@equation@counter
\fi
```

Check for default options (which may switch off indexing).

```
\@gls@setdefault@glslink@opts
```

Implement any pre-key settings.

```
\csuse{@glsxtr@#3@prekeys}%
```

Assign keys.

```
\setkeys{#3}{#1}%
```

Implement any post-key settings. Is the auto-add on?

```
\glsxtr@do@autoadd{#3}%
```

Check post-key hook.

```
\csuse{@glsxtr@#3@postkeys}%
```

Increment associated counter.

```
\glsxtr@inc@wrglossaryctr{#2}%
```

Check if noindex option has been used.

```
\ifKV@glslink@noindex
\else
  \glswriteentry{#2}%
  {%
```

Check if thevalue has been set.

```
    \ifdefempty{\@glsxtr@thevalue}%
    {%
```

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

```
      \ifx\@glsxtr@org@theHvalue\@glsxtr@theHvalue
      \else
        \let\theHglsentrycounter\@glsxtr@theHvalue
      \fi
```

Save the entry counter.

```
      \glsxtr@saveentrycounter
```

Temporarily redefine `\@@do@@wrglossary` for use with `\glsxtr@@do@wrglossary`.

```
      \let\@@do@@wrglossary\@glsxtr@dorecord
    }%
    {%
```

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.)

```
                \let\theglsentrycounter\@glsxtr@thevalue
                \let\theHglsentrycounter\@glsxtr@theHvalue
                \let\@@do@@wrglossary\@glsxtr@dorecordnodefer
            }%
            \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
                \glsxtr@@do@wrglossary{#2}%
            \else
```

No need to escape special characters.

```
                \@@do@@wrglossary
            \fi
        }%
    \fi
    \endgroup
  }%
}
```

\@glsxtr@glslink@prekeys

```
\newcommand{\@glsxtr@glslink@prekeys}{\glslinkpresetkeys}
```

\@glsxtr@glslink@postkeys

```
\newcommand{\@glsxtr@glslink@postkeys}{\glslinkpostsetkeys}
```

\@glsxtr@glossadd@prekeys

```
\newcommand{\@glsxtr@glossadd@prekeys}{\glsaddpresetkeys}
```

\@glsxtr@glossadd@postkeys

```
\newcommand{\@glsxtr@glossadd@postkeys}{\glsaddpostsetkeys}
```

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

```
\newcommand*\@glsxtr@dorecord{%
 \@@glsxtr@dorecord\@gls@label\glsxtr@record\@glsxtr@do@nameref@record
}
```

\@@glsxtr@dorecord

```
\newcommand*\@@glsxtr@dorecord[3]{%
    \global\let\@glsrecordlocref\theglsentrycounter
    \let\@glsxtr@orgprefix\@glo@counterprefix
    \ifx\theglsentrycounter\theHglsentrycounter
      \def\@glo@counterprefix{}%
    \else
```

Protect against non-expandable commands occurring in the location.

```
    \protected@edef\@glsxtr@theentrycounter{\theglsentrycounter}%
    \protected@edef\@glsxtr@theHentrycounter{\theHglsentrycounter}%
    \@onelevel@sanitize\@glsxtr@theentrycounter
```

```
    \@onelevel@sanitize\@glsxtr@theHentrycounter
    \@xp@gls@getcounterprefix
      {\@glsxtr@theentrycounter}{\@glsxtr@theHentrycounter}%
  \fi
```

Don't protect the `\@glsrecordlocref` from premature expansion. If the counter isn't `page` then it needs expanding. If the location includes `\thepage` then `\protected@write` will automatically deal with it.

```
  \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
    #3%
    {#1}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}%
    {\@glsrecordlocref}%
  \else
    \@bibgls@write@aux{}{\string#2%
      {#1}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}%
      {\@glsrecordlocref}}%
  \fi
  \@glsxtr@counterrecordhook
  \let\@glo@counterprefix\@glsxtr@orgprefix
}
```

`\@glsxtr@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.

```
\newcommand*\@glsxtr@dorecordnodefer{%
  \ifx\theglsentrycounter\theHglsentrycounter
    \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
      \@glsxtr@do@nameref@record
        {\@gls@label}{}{\@gls@counter}{\@glsnumberformat}%
        {\theglsentrycounter}%
    \else
      \@bibgls@write@aux{}{\string\glsxtr@record
        {\@gls@label}{}{\@gls@counter}{\@glsnumberformat}%
        {\theglsentrycounter}}%
    \fi
  \else
    \@xp@gls@getcounterprefix{\theglsentrycounter}{\theHglsentrycounter}%
    \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
      \@glsxtr@do@nameref@record
        {\@gls@label}{\@glo@counterprefix}{\@gls@counter}%
        {\@glsnumberformat}{\theglsentrycounter}%
    \else
      \@bibgls@write@aux{}{\string\glsxtr@record
        {\@gls@label}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}%
        {\theglsentrycounter}}%
    \fi
  \fi
  \@glsxtr@counterrecordhook
}
```

\@glsxtr@ifnum@mmode Check if in a numbered maths environment. The amsmath package is automatically loaded by datatool-base, which is required by glossaries, so \ifst@rred and \if@display should both be defined.

```
\newcommand{\@glsxtr@ifnum@mmode}[2]{%
 \ifmmode
   \ifst@rred
    #2%
   \else
```

Non-amsmath environments and regular inline math mode isn't flagged as starred by amsmath, but we can't use \mathchoice in this case as it's not the current style that's relevant. Instead we can use amsmath's \if@display. This may not work for environments that aren't provided by amsmath.

```
    \if@display #1\else #2\fi
   \fi
 \else
   #2%
 \fi
}
```

\@glsxtr@do@nameref@record With record=nameref, the current label information is included in the record, but this may not have been defined, so \csuse will prevent an undefined control sequence error and just leave the last two arguments blank if there's no information. In the event that a record is in amsmath's align environment \@currentHref will be out. There may be other instances where \@currentHref is out, so this also saves \theHglsentrycounter, which is useful if it can't be obtained by prefixing \theglsentrycounter.

```
\newcommand*{\@glsxtr@do@nameref@record}[5]{%
 \gls@ifnotmeasuring
 {%
   \@bibgls@write@aux{}{\string\glsxtr@record@nameref
    {#1}{#2}{#3}{#4}{#5}%
    {\csuse{@currentlabelname}}{\csuse{@currentHref}}%
    {\theHglsentrycounter}}%
 }%
}
```

\@@glsxtr@recordcounter

```
\newcommand*{\@@glsxtr@recordcounter}{%
 \@glsxtr@noop@recordcounter
}
```

\@glsxtr@noop@recordcounter

```
\newcommand*{\@glsxtr@noop@recordcounter}[1]{%
 \PackageError{glossaries-extra}{\string\GlsXtrRecordCounter\space
  requires record=only or record=hybrid package option}{}%
}
```

`\@glsxtr@op@recordcounter`

```
\newcommand*{\@glsxtr@op@recordcounter}[1]{%
  \protected@eappto\@glsxtr@counterrecordhook{\noexpand\@glsxtr@docounterrecord{#1}}%
}
```

`\@glsxtr@recordsee` Deal with `\glssee` in record mode. (This doesn't increment the associated counter.)

```
\newcommand*{\@glsxtr@recordsee}[2]{%
 \@@glsxtrwrglossmark
 \def\@gls@xref{#2}%
 \@onelevel@sanitize\@gls@xref
 \@bibgls@write@aux{}{\string\glsxtr@recordsee{#1}{\@gls@xref}}%
}
```

`\printunsrtglossaryunit`

```
\newcommand{\printunsrtglossaryunit}{%
  \print@noop@unsrtglossaryunit
}
```

`\glsxtr@setup@record` Initialise.

```
\newcommand*{\glsxtr@setup@record}{\let\@@do@wrglossary\glsxtr@@do@wrglossary}
```

`\@indexonly@saveentrycounter` Only store the entry counter information if the indexing is on.

```
\newcommand*{\glsxtr@indexonly@saveentrycounter}{%
 \ifKV@glslink@noindex
 \else
   \glsxtr@saveentrycounter
 \fi
}
```

`\glsxtr@addloclistfield`

```
\newcommand*{\glsxtr@addloclistfield}{%
 \key@ifundefined{glossentry}{loclist}%
 {%
   \define@key{glossentry}{loclist}{\def\@glo@loclist{##1}}%
   \appto\@gls@keymap{,{loclist}{loclist}}%
   \appto\@newglossaryentryprehook{\def\@glo@loclist{}}%
   \appto\@newglossaryentryposthook{%
     \gls@assign@field{}{\@glo@label}{loclist}{\@glo@loclist}%
   }%
   \glssetnoexpandfield{loclist}%
 }%
 {}%
```

The loclist field is just a comma-separated list. The location field is the formatted list.

```
 \key@ifundefined{glossentry}{location}%
 {%
   \define@key{glossentry}{location}{\def\@glo@location{##1}}%
```

11

```
                            \appto\@gls@keymap{,{location}{location}}%
                            \appto\@newglossaryentryprehook{\def\@glo@location{}}%
                            \appto\@newglossaryentryposthook{%
                               \gls@assign@field{}{\@glo@label}{location}{\@glo@location}%
                            }%
                            \glssetnoexpandfield{location}%
                         }%
                         {}%
```

Add a key to store the group heading.

```
                         \key@ifundefined{glossentry}{group}%
                         {%
                            \define@key{glossentry}{group}{\def\@glo@group{##1}}%
                            \appto\@gls@keymap{,{group}{group}}%
                            \appto\@newglossaryentryprehook{\def\@glo@group{}}%
                            \appto\@newglossaryentryposthook{%
                               \gls@assign@field{}{\@glo@label}{group}{\@glo@group}%
                            }%
                            \glssetnoexpandfield{group}%
                         }%
                         {}%
                         }
```

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

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

\@glsxtr@record@setting@alsoindex As from v1.46, the record=alsoindex is renamed to record=hybrid with record=alsoindex
                         as a deprecated synonym to avoid confusion. The internal commands that in-
                         clude alsoindex in the name will remain unchanged to avoid breaking things,
                         but this command will need to be redefined by record=hybrid.

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

\@glsxtr@record@setting@only

```
                         \newcommand*{\@glsxtr@record@setting@only}{only}
```

\@glsxtr@record@setting@nameref

```
                         \newcommand*{\@glsxtr@record@setting@nameref}{nameref}
```

\@glsxtr@if@record@only

```
                         \newcommand*{\@glsxtr@if@record@only}[2]{%
                          \ifx\@glsxtr@record@setting\@glsxtr@record@setting@only
                           #1%
                          \else
                           \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
                             #1%
                           \else
                             #2%
                           \fi
                          \fi
                         }
```

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

Used by hybrid method if \printglossary isn't used.

```
\newcommand\@glsxtr@warn@hybrid@noprintgloss{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesExtraWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesExtraWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^JYou have requested the hybrid setting
      record=\@glsxtr@record@setting\space which requires a
      combination of bib2gls (to fetch entries) and makeindex/xindy
      (to sort and collate the entries). If you only want to use
      bib2gls then change the option to record=only or record=nameref}%
  }%
}
```

\@glsxtr@record@only@setup Initialisation code for record=only and record=nameref

```
\newcommand*{\@glsxtr@record@only@setup}{%
 \def\glsxtr@setup@record{%
   \@glsxtr@autoseeindexfalse
   \let\@do@seeglossary\@glsxtr@recordsee
   \let\@glsxtr@record\@@glsxtr@record
   \let\@@do@wrglossary\@glsxtr@do@record@wrglossary
   \let\@gls@saveentrycounter\relax
   \let\glsxtrundefaction\@glsxtr@warn@undefaction
   \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
   \glsxtr@addloclistfield
   \renewcommand*{\@glsxtr@autoindexcrossrefs}{}%
   \let\@@glsxtr@recordcounter\@glsxtr@op@recordcounter
   \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
```

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

```
   \def\glsxtrsetaliasnoindex{}%
```

\@gls@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.

```
   \ifdef\@gls@setupsort@none{\@gls@setupsort@none}{}%
```

Warn about using \printglossary:

```
   \def\glsxtrNoGlossaryWarning{\@glsxtr@record@noglossarywarning}%
```

Load glossaries-extra-bib2gls:

```
   \RequirePackage{glossaries-extra-bib2gls}%
 }%
}
```

record   Now define the record package option.   As from v1.46, record=alsoindex is a deprecated synonym of record=hybrid to avoid confusion.

```
\define@choicekey{glossaries-extra.sty}{record}
[\@glsxtr@record@setting\glsxtr@record@nr]%
{off,only,alsoindex,nameref,hybrid}%
[only]%
{%
   \ifcase\glsxtr@record@nr\relax
```

Don't record.

```
   \def\glsxtr@setup@record{%
      \renewcommand*{\@do@seeglossary}{\@glsxtr@doseeglossary}%
      \renewcommand*{\@glsxtr@record}[3]{}%
      \let\@@do@wrglossary\glsxtr@@do@wrglossary
      \let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
      \let\glsxtrundefaction\@glsxtr@err@undefaction
      \let\glsxtr@warnonexistsordo\@gobble
      \let\@@glsxtr@recordcounter\@glsxtr@noop@recordcounter
      \def\printunsrtglossaryunit{\print@noop@unsrtglossaryunit}%
      \undef\glsxtrsetaliasnoindex
   }%
\or
```

Only record (don't index).

```
   \@glsxtr@record@only@setup
\or
```

Record and index. This option doesn't load glossaries-extra-bib2gls as the sorting is performed by xindy or makeindex. Index in this sense refers to the indexing mechanism used with indexing applications such as makeindex and xindy, but this could be confused with recording locations so "alsoindex" is now deprecated in favour of "hybrid", which is more obvious.

```
   \def\glsxtr@setup@record{%
      \renewcommand*{\@glsxtr@record@setting@alsoindex}{alsoindex}%
      \renewcommand*{\@do@seeglossary}{\@glsxtr@dosee@alsoindex@glossary}%
      \let\@glsxtr@record\@@glsxtr@record
      \let\@@do@wrglossary\glsxtr@do@alsoindex@wrglossary
      \let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
      \let\glsxtrundefaction\@glsxtr@warn@undefaction
      \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
      \glsxtr@addloclistfield
      \let\@@glsxtr@recordcounter\@glsxtr@op@recordcounter
      \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
      \undef\glsxtrsetaliasnoindex
   }%
\or
```

Only record (don't index) but also include nameref information.

```
   \@glsxtr@record@only@setup
   \ifundef\hyperlink
   {\GlossariesExtraWarning{You have requested record=nameref but
```

14

```
                                 the document doesn't support hyperlinks}}%
                              {}%
                           \or
```
Hybrid record (use bib2gls to fetch definitions) and index (use makeindex/xindy
to sort and collate).

```
                           \def\glsxtr@setup@record{%
                              \renewcommand*{\@glsxtr@record@setting@alsoindex}{hybrid}%
                              \renewcommand*{\@do@seeglossary}{\@glsxtr@dosee@alsoindex@glossary}%
                              \let\@glsxtr@record\@@glsxtr@record
                              \let\@@do@wrglossary\glsxtr@do@alsoindex@wrglossary
                              \let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
                              \let\glsxtrundefaction\@glsxtr@warn@undefaction
                              \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
                              \glsxtr@addloclistfield
                              \let\@@glsxtr@recordcounter\@glsxtr@op@recordcounter
                              \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
                              \undef\glsxtrsetaliasnoindex
                           }%
                        \fi
                     }
```

bibglsaux Provide an option to put the records in a different aux file that will only be read
           by bib2gls and not by LaTeX. A large number of records in the aux file can slow
           down the document build as LaTeX has to parse it all. This will require an extra
           write register, so may not be so desirable for documents with small glossaries
           but a large number of temporary files.

```
                  \define@key{glossaries-extra.sty}{bibglsaux}{%
                    \glsxtrsetbibglsaux{#1}%
                  }
```

\glsxtrsetbibglsaux

```
                  \newcommand{\glsxtrsetbibglsaux}[1]{%
                   \renewcommand{\@glsxtr@setup@bibglsaux}{\@@glsxtr@setup@bibglsaux{#1}}%
                  }
```

\@glsxtr@setup@bibglsaux

```
                  \newcommand{\@glsxtr@setup@bibglsaux}{%
                   \renewcommand{\glsxtrsetbibglsaux}[1]{%
                     \@@glsxtr@setup@bibglsaux{##1}%
                   }%
                  }
                  \AtBeginDocument{\@glsxtr@setup@bibglsaux}
```

\@@glsxtr@setup@bibglsaux

```
                  \newcommand{\@@glsxtr@setup@bibglsaux}[1]{%
                   \ifstrempty{#1}%
                   {\renewcommand{\@bibgls@write@aux}{\protected@write\@auxout}}%
                   {\@set@bibgls@write@aux{#1.aux}}%
                  }
```

`\@bibgls@write@aux` Just used for writing records.

```
\newcommand{\@bibgls@write@aux}{\protected@write\@auxout}
```

`\@set@bibgls@write@aux`

```
\newcommand{\@set@bibgls@write@aux}[1]{%
  \protected@write\@auxout{}%
   {\string\providecommand{\string\@bibgls@input}[1]{}}%
  \protected@write\@auxout{}{\string\@bibgls@input{#1}}%
  \global\newwrite\@bibgls@auxout
  \openout\@bibgls@auxout=#1
  \AtEndDocument{\closeout\@bibgls@auxout}%
  \gdef\@bibgls@write@aux{\protected@write\@bibgls@auxout}%
  \gdef\@set@bibgls@write@aux##1{\GlossariesExtraWarning{repeated
    invocation of bibglsaux option ignored}}%
}
```

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.

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

```
\newcommand*{\@glsxtr@docdefval}{0}
```

Need to provide conditional commands that are backward compatible:

`\if@glsxtrdocdef`

```
\newcommand*{\if@glsxtrdocdef}{\ifnum\@glsxtr@docdefval>0 }
```

`\@glsxtrdocdeftrue`

```
\newcommand*{\@glsxtrdocdeftrue}{\def\@glsxtr@docdefval{1}}
```

`\@glsxtrdocdeffalse`

```
\newcommand*{\@glsxtrdocdeffalse}{\def\@glsxtr@docdefval{0}}
```

docdef 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.

```
\define@choicekey{glossaries-extra.sty}{docdef}
 [\@glsxtr@docdefsetting\@glsxtr@docdefval]%
 {false,true,restricted,atom}[true]%
{%
  \ifnum\@glsxtr@docdefval>1\relax
    \renewcommand*{\@glsdoifexistsorwarn}{\glsdoifexists}%
  \else
    \renewcommand*{\@glsdoifexistsorwarn}{\glsdoifexistsorwarn}%
  \fi
}
```

`\if@glsxtrdocdefrestricted`

> `\newcommand*{\if@glsxtrdocdefrestricted}{\ifnum\@glsxtr@docdefval>1 }`

`\@glsdoifexistsorwarn` 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).

> `\newcommand*{\@glsdoifexistsorwarn}{\glsdoifexistsorwarn}`

`indexcrossrefs` Automatically index cross references at the end of the document

> ```
> \define@boolkey{glossaries-extra.sty}[@glsxtr]{indexcrossrefs}[true]{%
>  \if@glsxtrindexcrossrefs
>  \else
>   \renewcommand*{\@glsxtr@autoindexcrossrefs}{}%
>  \fi
> }
> ```

Switch off since this can increase the build time.

> `\@glsxtrindexcrossrefsfalse`

But allow see and seealso keys to switch it on automatically.

`\@glsxtr@autoindexcrossrefs`

> `\newcommand*{\@glsxtr@autoindexcrossrefs}{\@glsxtrindexcrossrefstrue}`

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

> ```
> \define@boolkey{glossaries-extra.sty}[@glsxtr@]{autoseeindex}[true]{%
> }
> \@glsxtr@autoseeindextrue
> ```

`equations` Provide a boolean option to automatically switch to the equation counter when in a numbered maths environment.

> ```
> \define@boolkey{glossaries-extra.sty}[@glsxtr@]{equations}[true]{%
> }
> \@glsxtr@equationsfalse
> ```

`\glsxtr@float`

> `\let\glsxtr@float\@float`

`\glsxtr@dblfloat`

> `\let\glsxtr@dblfloat\@dblfloat`

`floats` Provide a boolean option to automatically switch to the the corresponding counter when in a float.

> ```
> \define@boolkey{glossaries-extra.sty}[@glsxtr@]{floats}[true]{%
>  \if@glsxtr@floats
>   \renewcommand*{\@float}[1]{\renewcommand{\glscounter}{##1}\glsxtr@float{##1}}%
>   \renewcommand*{\@dblfloat}[1]{\renewcommand{\glscounter}{##1}\glsxtr@dblfloat{##1}}%
>  \else
> ```

17

```
                                \let\@float\glsxtr@float
                                \let\@dblfloat\glsxtr@dblfloat
                              \fi
                           }
                           \@glsxtr@floatsfalse
```

\GlossariesExtraInfo Allow users to suppress information messages.

```
                           \newcommand*{\GlossariesExtraInfo}[1]{\PackageInfo{glossaries-extra}{#1}}
```

\GlossariesExtraWarning Allow users to suppress warnings.

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

\GlossariesExtraWarningNoLine Allow users to suppress warnings.

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

                           \@glsxtr@declareoption{nowarn}{%
                            \let\GlossariesExtraWarning\@gobble
                            \let\GlossariesExtraWarningNoLine\@gobble
                            \glsxtr@dooption{nowarn}%
                           }
```

\@glsxtr@defpostpunc Redefines \glspostdescription. The postdot and nopostdot options will have
to redefine this.

```
                           \newcommand*{\@glsxtr@defpostpunc}{}
```

postdot Shortcut for nopostdot=false

```
                           \@glsxtr@declareoption{postdot}{%
                            \glsxtr@dooption{nopostdot=false}%
                            \renewcommand*{\@glsxtr@defpostpunc}{%
                              \renewcommand*{\glspostdescription}{%
                                \ifglsnopostdot\else.\spacefactor\sfcode`\. \fi}%
                            }%
                           }
```

nopostdot Needs to redefine \@glsxtr@defpostpunc

```
                           \define@choicekey{glossaries-extra.sty}{nopostdot}{true,false}[true]{%
                            \glsxtr@dooption{nopostdot=#1}%
                            \renewcommand*{\@glsxtr@defpostpunc}{%
                              \renewcommand*{\glspostdescription}{%
                                \ifglsnopostdot\else.\spacefactor\sfcode`\. \fi}%
                            }%
                           }
```

postpunc Set the post-description punctuation. This also sets the \ifglsnopostdot con-
ditional, which now indicates if the post-description punctuation has been sup-
pressed.

```
                           \define@key{glossaries-extra.sty}{postpunc}{%
                            \glsxtr@dooption{nopostdot=false}%
```

```
                    \ifstrequal{#1}{dot}%
                    {%
                      \renewcommand*{\@glsxtr@defpostpunc}{%
                        \renewcommand*{\glspostdescription}{.\spacefactor\sfcode`\. }%
                      }%
                    }%
                    {%
                      \ifstrequal{#1}{comma}%
                      {%
                        \renewcommand*{\@glsxtr@defpostpunc}{%
                          \renewcommand*{\glspostdescription}{,}%
                        }%
                      }%
                      {%
                        \ifstrequal{#1}{none}%
                        {%
                          \glsxtr@dooption{nopostdot=true}%
                          \renewcommand*{\@glsxtr@defpostpunc}{%
                            \renewcommand*{\glspostdescription}{}%
                          }%
                        }%
                        {%
                          \renewcommand*{\@glsxtr@defpostpunc}{%
                            \renewcommand*{\glspostdescription}{#1}%
                          }%
                        }%
                      }%
                    }%
                  }
```

\glsxtrabbrvtype Glossary type for abbreviations.

```
\newcommand*{\glsxtrabbrvtype}{\glsdefaulttype}
```

\@glsxtr@abbreviationsdef Set by abbreviations option.

```
\newcommand*{\@glsxtr@abbreviationsdef}{}
```

\abbreviationsname v1.50 unconditionally provide this command, so it can be redefined by a language module.

```
\@ifpackageloaded{babel}%
{\providecommand{\abbreviationsname}{\acronymname}}%
{\providecommand{\abbreviationsname}{Abbreviations}}%
```

\@glsxtr@doabbreviationsdef

```
\newcommand*{\@glsxtr@doabbreviationsdef}{%
  \newglossary[glg-abr]{abbreviations}{gls-abr}{glo-abr}{\abbreviationsname}%
  \renewcommand*{\glsxtrabbrvtype}{abbreviations}%
  \newcommand*{\printabbreviations}[1][]{%
    \printglossary[type=\glsxtrabbrvtype,##1]%
  }%
  \disable@keys{glossaries-extra.sty}{abbreviations}%
```

19

If the acronym option hasn't been used, change `\acronymtype` to `\glsxtrabbrvtype`.

```
  \ifglsacronym
  \else
    \renewcommand*{\acronymtype}{\glsxtrabbrvtype}%
  \fi
}%
```

abbreviations If abbreviations, create a new glossary type for abbreviations.

```
\@glsxtr@declareoption{abbreviations}{%
  \let\@glsxtr@abbreviationsdef\@glsxtr@doabbreviationsdef
}
```

\shortcut@gls

```
\newcommand{\shortcut@gls}{\cgls}
```

\shortcut@glspl

```
\newcommand{\shortcut@glspl}{\cglspl}
```

\shortcut@Gls

```
\newcommand{\shortcut@Gls}{\cGls}
```

\shortcut@Glspl

```
\newcommand{\shortcut@Glspl}{\cGlspl}
```

\shortcut@GLS

```
\newcommand{\shortcut@GLS}{\cGLS}
```

\shortcut@GLSpl

```
\newcommand{\shortcut@GLSpl}{\cGLSpl}
```

\GlsXtrDefineAbbreviationShortcuts Enable shortcut commands for the abbreviations. Unlike the analogous command provided by glossaries, this uses `\newcommand` instead of `\let` as a safety feature (except for `\newabbr` which is also provided with `\GlsXtrDefineAcShortcuts`).

```
\newcommand*{\GlsXtrDefineAbbreviationShortcuts}{%
  \newcommand*{\ab}{\shortcut@gls}%
  \newcommand*{\abp}{\shortcut@glspl}%
  \newcommand*{\as}{\glsxtrshort}%
  \newcommand*{\asp}{\glsxtrshortpl}%
  \newcommand*{\al}{\glsxtrlong}%
  \newcommand*{\alp}{\glsxtrlongpl}%
  \newcommand*{\af}{\glsxtrfull}%
  \newcommand*{\afp}{\glsxtrfullpl}%
  \newcommand*{\Ab}{\shortcut@Gls}%
  \newcommand*{\Abp}{\shortcut@Glspl}%
  \newcommand*{\As}{\Glsxtrshort}%
  \newcommand*{\Asp}{\Glsxtrshortpl}%
  \newcommand*{\Al}{\Glsxtrlong}%
```

```
\newcommand*{\Alp}{\Glsxtrlongpl}%
\newcommand*{\Af}{\Glsxtrfull}%
\newcommand*{\Afp}{\Glsxtrfullpl}%
\newcommand*{\AB}{\shortcut@GLS}%
\newcommand*{\ABP}{\shortcut@GLSpl}%
\newcommand*{\AS}{\GLSxtrshort}%
\newcommand*{\ASP}{\GLSxtrshortpl}%
\newcommand*{\AL}{\GLSxtrlong}%
\newcommand*{\ALP}{\GLSxtrlongpl}%
\newcommand*{\AF}{\GLSxtrfull}%
\newcommand*{\AFP}{\GLSxtrfullpl}%
\glsmfuaddmap{\ab}{\Ab}%
\glsmfublocker{\AB}%
\glsmfuaddmap{\abp}{\Abp}%
\glsmfublocker{\ABP}%
\glsmfuaddmap{\as}{\As}%
\glsmfublocker{\AS}%
\glsmfuaddmap{\asp}{\Asp}%
\glsmfublocker{\ASP}%
\glsmfuaddmap{\al}{\Al}%
\glsmfublocker{\AL}%
\glsmfuaddmap{\alp}{\Alp}%
\glsmfublocker{\ALP}%
\glsmfuaddmap{\af}{\Af}%
\glsmfublocker{\AF}%
\glsmfuaddmap{\afp}{\Afp}%
\glsmfublocker{\AFP}%

\providecommand*{\newabbr}{\newabbreviation}%
```
Disable this command after it's been used.
```
\let\GlsXtrDefineAbbreviationShortcuts\relax
}
```

`\GlsXtrDefineAcShortcuts` Enable shortcut commands for the abbreviations, but uses the analogous commands provided by glossaries.
```
\newcommand*{\GlsXtrDefineAcShortcuts}{%
\newcommand*{\ac}{\shortcut@gls}%
\newcommand*{\acp}{\shortcut@glspl}%
\newcommand*{\acs}{\glsxtrshort}%
\newcommand*{\acsp}{\glsxtrshortpl}%
\newcommand*{\acl}{\glsxtrlong}%
\newcommand*{\aclp}{\glsxtrlongpl}%
\newcommand*{\acf}{\glsxtrfull}%
\newcommand*{\acfp}{\glsxtrfullpl}%
\newcommand*{\Ac}{\shortcut@Gls}%
\newcommand*{\Acp}{\shortcut@Glspl}%
\newcommand*{\Acs}{\Glsxtrshort}%
\newcommand*{\Acsp}{\Glsxtrshortpl}%
\newcommand*{\Acl}{\Glsxtrlong}%
```

```
\newcommand*{\Aclp}{\Glsxtrlongpl}%
\newcommand*{\Acf}{\Glsxtrfull}%
\newcommand*{\Acfp}{\Glsxtrfullpl}%
\newcommand*{\AC}{\shortcut@GLS}%
\newcommand*{\ACP}{\shortcut@GLSpl}%
\newcommand*{\ACS}{\GLSxtrshort}%
\newcommand*{\ACSP}{\GLSxtrshortpl}%
\newcommand*{\ACL}{\GLSxtrlong}%
\newcommand*{\ACLP}{\GLSxtrlongpl}%
\newcommand*{\ACF}{\GLSxtrfull}%
\newcommand*{\ACFP}{\GLSxtrfullpl}%
\glsmfuaddmap{\ac}{\Ac}%
\glsmfublocker{\AC}%
\glsmfuaddmap{\acp}{\Acp}%
\glsmfublocker{\ACP}%
\glsmfuaddmap{\acs}{\Acs}%
\glsmfublocker{\ACS}%
\glsmfuaddmap{\acsp}{\Acsp}%
\glsmfublocker{\ACSP}%
\glsmfuaddmap{\acl}{\Acl}%
\glsmfublocker{\ACL}%
\glsmfuaddmap{\aclp}{\Aclp}%
\glsmfublocker{\ACLP}%
\glsmfuaddmap{\acf}{\Acf}%
\glsmfublocker{\ACF}%
\glsmfuaddmap{\acfp}{\Acfp}%
\glsmfublocker{\ACFP}%

\providecommand*{\newabbr}{\newabbreviation}%
```

Disable this command after it's been used.

```
\let\GlsXtrDefineAcShortcuts\relax
}
```

\GlsXtrDefineOtherShortcuts Similarly provide shortcut versions for the commands provided by the symbols and numbers options.

```
\newcommand*{\GlsXtrDefineOtherShortcuts}{%
  \newcommand*{\newentry}{\newglossaryentry}%
  \ifdef\printsymbols
  {%
    \newcommand*{\newsym}{\glsxtrnewsymbol}%
  }{}%
  \ifdef\printnumbers
  {%
    \newcommand*{\newnum}{\glsxtrnewnumber}%
  }{}%
  \let\GlsXtrDefineOtherShortcuts\relax
}
```

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.)

\@glsxtr@setupshortcuts Command used to set the shortcuts option.

```
\newcommand*{\@glsxtr@setupshortcuts}{}
```

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

```
\newcommand*{\@glsxtr@shortcutsval}{\ifglsacrshortcuts acro\else none\fi}%
```

shortcuts 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).

```
\define@choicekey{glossaries-extra.sty}{shortcuts}%
  [\@glsxtr@shortcutsval\@glsxtr@shortcutsnr]%
  {acronyms,acro,abbreviations,abbr,other,all,true,ac,acother,abother,none,false}[true]{%
    \ifcase\@glsxtr@shortcutsnr\relax % acronyms
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \glsacrshortcutstrue
        \DefineAcronymSynonyms
      }%
    \or % acro
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \glsacrshortcutstrue
        \DefineAcronymSynonyms
      }%
    \or % abbreviations
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \GlsXtrDefineAbbreviationShortcuts
      }%
    \or % abbr
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \GlsXtrDefineAbbreviationShortcuts
      }%
    \or % other
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \GlsXtrDefineOtherShortcuts
      }%
    \or % all
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \glsacrshortcutstrue

        \GlsXtrDefineAcShortcuts
        \GlsXtrDefineAbbreviationShortcuts
        \GlsXtrDefineOtherShortcuts
      }%
    \or % true
```

```
              \renewcommand*{\@glsxtr@setupshortcuts}{%
                \glsacrshortcutstrue

                \GlsXtrDefineAcShortcuts
                \GlsXtrDefineAbbreviationShortcuts
                \GlsXtrDefineOtherShortcuts
              }%

          \or % ac
              \renewcommand*{\@glsxtr@setupshortcuts}{%
                \glsacrshortcutstrue
                \GlsXtrDefineAcShortcuts
              }%
          \or % acother

              \renewcommand*{\@glsxtr@setupshortcuts}{%
                \glsacrshortcutstrue
                \GlsXtrDefineAcShortcuts
                \GlsXtrDefineOtherShortcuts
              }%
          \or % abother

              \renewcommand*{\@glsxtr@setupshortcuts}{%
                \glsacrshortcutstrue
                \GlsXtrDefineAbbreviationShortcuts
                \GlsXtrDefineOtherShortcuts
              }%
```

Leave none and false as last option.

```
          \else % none, false
              \renewcommand*{\@glsxtr@setupshortcuts}{}%
          \fi
        }
```

\@glsxtr@doaccsupp

```
              \newcommand*{\@glsxtr@doaccsupp}{}
```

glossaries-accsupp can't be loaded after glossaries-extra. glossaries-accsupp
v4.29+ checks \@glsxtr@doaccsupp to determine if it's been loaded too late.

accsupp If accsupp, load glossaries-accsupp package.

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

\@glsxtr@doloadprefix

```
              \newcommand*{\@glsxtr@doloadprefix}{}
```

prefix If prefix, load glossaries-prefix package.

```
          \@glsxtr@declareoption{prefix}{%
           \renewcommand*{\@glsxtr@doloadprefix}{\RequirePackage{glossaries-prefix}}}
```

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

```
\newcommand{\glsxtrNoGlossaryWarning}[1]{%
  \GlossariesExtraWarning{Glossary '#1' is missing}%
  \@glsxtr@defaultnoglossarywarning{#1}%
}
```

`nomissingglstext` If true, suppress the text and warning produced if the external glossary file is missing.

```
\define@choicekey{glossaries-extra.sty}{nomissingglstext}
 [\@glsxtr@nomissingglstextval\@glsxtr@nomissingglstextnr]%
 {true,false}[true]{%
   \ifcase\@glsxtr@nomissingglstextnr\relax % true
     \renewcommand{\glsxtrNoGlossaryWarning}[1]{\null}%
   \else % false
     \renewcommand{\glsxtrNoGlossaryWarning}[1]{%
       \@glsxtr@defaultnoglossarywarning{#1}%
     }%
   \fi
 }
```

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

`\@glsxtr@redefstyles`

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

`stylemods`

```
\define@key{glossaries-extra.sty}{stylemods}[default]{%
 \ifstrequal{#1}{default}%
 {%
   \renewcommand*{\@glsxtr@redefstyles}{%
     \RequirePackage{glossaries-extra-stylemods}}%
 }%
 {%
   \ifstrequal{#1}{all}%
   {%
     \renewcommand*{\@glsxtr@redefstyles}{%
       \PassOptionsToPackage{all}{glossaries-extra-stylemods}%
       \RequirePackage{glossaries-extra-stylemods}%
     }%
   }%
   {%
     \renewcommand*{\@glsxtr@redefstyles}{}%
     \@for\@glsxtr@tmp:=#1\do{%
       \IfFileExists{glossary-\@glsxtr@tmp.sty}%
       {%
         \eappto\@glsxtr@redefstyles{%
           \noexpand\RequirePackage{glossary-\@glsxtr@tmp}}%
       }%
```

```
        {%
          \PackageError{glossaries-extra}%
           {Glossaries style package 'glossary-\@glsxtr@tmp.sty'
            doesn't exist (did you mean to use the 'style' key?)}%
           {The list of values (#1) in the 'stylemods' key should
            match the glossary-xxx.sty files provided with
            glossaries.sty}%
        }%
      }%
      \appto\@glsxtr@redefstyles{\RequirePackage{glossaries-extra-stylemods}}%
    }
  }%
}
```

\@glsxtr@do@style

```
\newcommand*{\@glsxtr@do@style}{}
```

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

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

Defer actual style change:

```
\renewcommand*{\@glsxtr@do@style}{%
```

Set this as the default style:

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

Set this style:

```
\setglossarystyle{#1}%
}%
}
```

\glsxtr@inc@wrglossaryctr Increments the associated counter if enabled. Does nothing by default. The
optional argument is the entry label in case it's required, but the wrglossary
counter is globally used by all entries.

```
\newcommand*{\glsxtr@inc@wrglossaryctr}[1]{}
```

> \GlsXtrInternalLocationHyperlink{⟨counter⟩}{⟨prefix⟩}
> {⟨location⟩}

XtrInternalLocationHyperlink

The first two arguments are always control sequences.

```
\newcommand*{\GlsXtrInternalLocationHyperlink}[3]{%
  \glsxtrhyperlink{#1#2#3}{#3}%
}
```

rglossary@locationhyperlink

```
\newcommand*{\@glsxtr@wrglossary@locationhyperlink}[3]{%
  \pageref{wrglossary.#3}%
}
```

26

indexcounter Define the wrglossary counter that's incremented every time an entry is indexed, except for cross-references. This is designed for use with bib2gls v1.4+. It can work with the other indexing methods but it will interfere with the number list collation. This option automatically implements counter=wrglossary.

Since glossaries automatically loads amsmath, there may be a problem if the indexing occurs in the equation environment, because only one \label is allowed in each instance of that environment. It's best to change the counter when in maths mode.

```
\@glsxtr@declareoption{indexcounter}{%
  \glsxtr@dooption{counter=wrglossary}%
  \ifundef\c@wrglossary
  {%
    \newcounter{wrglossary}%
    \renewcommand{\thewrglossary}{\arabic{wrglossary}}%
  }%
  {}%
  \renewcommand*{\glsxtr@inc@wrglossaryctr}[1]{%
```

Only increment if the current counter is wrglossary.

```
    \ifdefstring\@gls@counter{wrglossary}%
    {%
      \refstepcounter{wrglossary}%
      \label{wrglossary.\thewrglossary}%
      \@@glsxtrwrglosscountermark{\thewrglossary}%
    }%
    {}%
  }%
  \renewcommand*{\GlsXtrInternalLocationHyperlink}[3]{%
    \ifdefstring\glsentrycounter{wrglossary}%
    {%
      \@glsxtr@wrglossary@locationhyperlink{##1}{##2}{##3}%
    }%
    {\glsxtrhyperlink{##1##2##3}{##3}}%
  }%
}
```

\@glsxtrwrglossmark Marks the place where indexing occurs. Does nothing by default.

```
\newcommand*{\@glsxtrwrglossmark}{}
```

\@@glsxtrwrglossmark Since \glsadd can be used in the preamble, this action needs to be disabled until the start of the document.

```
\newcommand*{\@@glsxtrwrglossmark}{}
\AtBeginDocument{\renewcommand*{\@@glsxtrwrglossmark}{\@glsxtrwrglossmark}}
```

\glsxtrwrglossmark

```
\newcommand*{\glsxtrwrglossmark}{\ensuremath{\cdot}}
```

\@glsxtrwrglosscountermark Marks the place where wrglossary counter is incremented. Does nothing by default.

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

`\@@glsxtrwrglosscountermark`

```
\newcommand*{\@@glsxtrwrglosscountermark}[1]{}
\AtBeginDocument{\renewcommand*{\@@glsxtrwrglosscountermark}{\@glsxtrwrglosscountermark}}
```

`\glsxtrwrglosscountermark`

```
\newcommand*{\glsxtrwrglosscountermark}[1]{\glsshowtargetfonttext{[#1]}}
```

`\@glsxtr@doshowtarget`

```
\newcommand\@glsxtr@doshowtarget[2]{#2}
```

`\glsxtrundefdebug` Don't do anything until after the document environment has begun.

```
\newcommand*{\glsxtrundefdebug}[1]{}
```

`\@glsxtrundefdebug` Use the same font as the targets.

```
\newcommand*{\@glsxtrundefdebug}[1]{%
\if@gls@debug \glsshowtargetfonttext{[#1]}\fi
}
```

`debug` Provide extra debug options.

```
\define@choicekey{glossaries-extra.sty}{debug}
 [\@glsxtr@debugval\@glsxtr@debugnr]%
 {true,false,showtargets,showwrgloss,all,showaccsupp}[true]{%
   \ifcase\@glsxtr@debugnr\relax % true
    \glsxtr@dooption{debug=true}%
    \renewcommand*{\@glsxtrwrglossmark}{}%
    \renewcommand*{\@glsxtrwrglosscountermark}[1]{}%
   \or % false
    \glsxtr@dooption{debug=false}%
    \renewcommand*{\@glsxtrwrglossmark}{}%
    \renewcommand*{\@glsxtrwrglosscountermark}[1]{}%
    \let\@glsxtr@doshowtarget\@secondoftwo
   \or % showtargets
    \glsxtr@dooption{debug=showtargets}%
    \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
   \or % showwrgloss
    \glsxtr@dooption{debug=true}%
    \renewcommand*{\@glsxtrwrglossmark}{\glsxtrwrglossmark}%
    \renewcommand*{\@glsxtrwrglosscountermark}{\glsxtrwrglosscountermark}%
   \or % all
    \glsxtr@dooption{debug=true,debug=showaccsupp}%
    % debug=showwrgloss:
    \renewcommand*{\@glsxtrwrglossmark}{\glsxtrwrglossmark}%
    \renewcommand*{\@glsxtrwrglosscountermark}{\glsxtrwrglosscountermark}%
    % debug=showtargets:
    \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
   \or % showaccsupp
    \glsxtr@dooption{debug=showaccsupp}%
   \fi
}
```

28

`\glsxtrshowtargetouter`

> `\newcommand*{\glsxtrshowtargetouter}{\glsshowtargetouter}`

`\glsxtrshowtargetinner`

> `\newcommand*{\glsxtrshowtargetinner}[1]{\glsshowtargetinner{#1}}`

Debugging show targets.

`\@glsxtrshowtargetleft`

> `\newcommand{\@glsxtrshowtargetleft}[2]{\@glsshowtarget{#1}#2\@glsxtrshowtargetmark}%`

`\@glsxtrshowtargetright`

> `\newcommand{\@glsxtrshowtargetright}[2]{\@glsxtrshowtargetmark#2\@glsshowtarget{#1}}%`

`\@glsxtrshowtargetmark`

> `\newcommand{\@glsxtrshowtargetmark}{}%`

showtargets Implements debug=showtargets and provides extra adjustments.

```
\define@choicekey{glossaries-extra.sty}{showtargets}
 [\@glsxtr@showtargetsval\@glsxtr@showtargetsnr]%
 {left,right,innerleft,innerright,annoteleft,annoteright}%
 {%
   \glsxtr@dooption{debug=showtargets}%
   \ifcase\@glsxtr@showtargetsnr\relax
     \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
     \def\glsxtrshowtargetouter{\glsshowtargetouter}%
     \def\glsxtrshowtargetinner{\glsshowtargetinner}%
     \let\@glsxtrshowtargetmark\empty
   \or
     \def\@glsxtr@doshowtarget{\@glsxtrshowtargetright}%
     \def\glsxtrshowtargetouter{\glsshowtargetouter}%
     \def\glsxtrshowtargetinner{\glsshowtargetinner}%
     \let\@glsxtrshowtargetmark\empty
   \or
     \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
     \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
     \def\glsxtrshowtargetinner{\glsshowtargetinnersymleft}%
     \let\@glsxtrshowtargetmark\empty
   \or
     \def\@glsxtr@doshowtarget{\@glsxtrshowtargetright}%
     \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
     \def\glsxtrshowtargetinner{\glsshowtargetinnersymright}%
     \let\@glsxtrshowtargetmark\empty
   \or
     \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
     \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
     \def\glsxtrshowtargetinner{\glsshowtargetinnersymleft}%
     \def\@glsxtrshowtargetmark{\@glsshowtargetmarkfmt\glsxtrshowtargetsymbolright}%
   \or
```

29

```
        \def\@glsxtr@doshowtarget{\@glsxtrshowtargetright}%
        \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
        \def\glsxtrshowtargetinner{\glsshowtargetinnersymright}%
        \def\@glsxtrshowtargetmark{\@glsshowtargetmarkfmt\glsxtrshowtargetsymbolleft}%
      \fi
    }
```

Pass all other options to glossaries.

Need to compensate for the problem identified in https://www.dickimaw-books.com/bugtracker.php?key=171

```
\newcommand*{\glsxtr@processunknownoptions}{}

\@ifpackageloaded{glossaries}
{%
  \DeclareOptionX*{%
   \edef\glsxtr@processunknownoptions{%
    \noexpand\setupglossaries{\expandonce\CurrentOption}}}
}
{%
  \DeclareOptionX*{%
   \expandafter\glsxtr@dooption\expandafter{\CurrentOption}}
}
```

Process options.

```
\ProcessOptionsX
```

Load glossaries if not already loaded.

```
\RequirePackage{glossaries}
\glsxtr@processunknownoptions
```

Load the glossaries-accsupp package if required.

```
\@glsxtr@doaccsupp
```

Load the glossaries-prefix package if required.

```
\@glsxtr@doloadprefix
```

Redefine \glspostdescription if required.

```
\@glsxtr@defpostpunc
```

\glsindexsetting This command was new to glossaries v4.50 so may not be defined. Note that record=only and record=nameref implement sort=none, which will change the default definition of \glsindexingsetting.

```
\let\@glsxtr@org@indexingsetting\glsindexingsetting
\providecommand{\glsindexingsetting}{\ifglsxindy xindy\else makeindex\fi}
\ifx\@glsxtr@org@indexingsetting\glsindexingsetting
  \renewcommand{\glsindexingsetting}{%
   \@glsxtr@if@record@only{bib2gls}{\ifglsxindy xindy\else makeindex\fi}%
  }
\else
 \@glsxtr@if@record@only{\renewcommand{\glsindexingsetting}{bib2gls}}{}%
\fi
```

30

The following commands are new to glossaries v4.50, so provide them if an older version is present.

\glssentencecase

    `\providecommand{\glssentencecase}[1]{\makefirstuc{#1}}`

\glslowercase   This uses `\MakeTextLowercase` because if `\glslowercase` isn't defined then textcase has been loaded and we might have an older kernel.

    `\providecommand{\glslowercase}[1]{\MakeTextLowercase{#1}}`

\glsuppercase   Not using `\unexpanded` because ditto the above.

    `\providecommand{\glsuppercase}[1]{\mfirstucMakeUppercase{#1}}`

\glspdfsentencecase   For use in PDF strings. Ensure argument fully expanded first. This command is provided rather than defined to allow for the possibility that it may be added to glossaries at a later date.

```
\ExplSyntaxOn
\providecommand{\glspdfsentencecase}[1]{ \exp_args:Ne \MFUsentencecase { #1 } }
\ExplSyntaxOff
```

\@Glsentryfield   This command was new to glossaries v4.50 so won't be defined for older versions.

```
\def\@Glsentryfield#1#2{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\glspdfsentencecase{\@gls@entry@field{#1}{#2}}}%
}
```

\glstexorpdfstring

```
\ifdef\glstexorpdfstring
{}
{
  \ifdef\texorpdfstring
  {\newcommand{\glstexorpdfstring}{\texorpdfstring}}
  {\newcommand{\glstexorpdfstring}[2]{#1}}
}
```

\@glsxtr@org@MakeUppercase   Save the original definition of `\MakeUppercase` in case it needs to be restored.

    `\let\@glsxtr@org@MakeUppercase\MakeUppercase`

\glsmeasurewidth   `\glsmeasurewidth` was only introduced to glossaries v4.51 so may not be available. This provides a definition that simply uses `\settowidth`.

```
\providecommand{\glsmeasurewidth}[2]{%
  \settowidth{#1}{#2}%
}
```

If mfirstuc v2.08+ is installed, provide interface commands. The simplest method is to test the existence of `\MFUsentencecase`, which is provided by mfirstuc v2.08+ but also by glossaries v4.50+. So it may be defined because glossaries v4.50+ is installed, in which case `\glsmfuexcl` etc are also defined,

31

but it may be defined because mfirstuc v2.08+ is installed but an older version of glossaries may be present, in which case \glsmfuexcl etc won't be defined.

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

Automatically identify exclusions, blockers and mappings.

\glsmfuexcl

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

\glsmfublocker

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

\glsmfuaddmap

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

Don't alter \MakeUppercase

\@glsxtr@saveMakeUppercase

```
\newcommand{\@glsxtr@saveMakeUppercase}{}
```

\@glsxtr@restoreMakeUppercase

```
\newcommand{\@glsxtr@restoreMakeUppercase}{}
```

\@glsxtr@assignMakeUppercase

```
\newcommand{\@glsxtr@assignMakeUppercase}{}
```

```
}
{
```

Provide \MFUsentencecase for use where expandable contexts are required.

\MFUsentencecase

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

Provide support for exclusions with \MFUsentencecase.

\glsmfuexcl

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

Just treat blockers and mappings as exclusions.

\glsmfublocker

```
\providecommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
```

\glsmfuaddmap

```
\providecommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
```

With old versions of mfirstuc, save and restore \MakeUppercase in the heading
hooks.

```
\newcommand{\@glsxtr@saveMakeUppercase}{%
  \let\@glsxtr@org@MakeUppercase\MakeUppercase
}
\newcommand{\@glsxtr@restoreMakeUppercase}{%
  \let\MakeUppercase\@glsxtr@org@MakeUppercase
}
\newcommand{\@glsxtr@assignMakeUppercase}{%
  \let\MakeUppercase\MakeTextUppercase
}
}
```

Finished LaTeX3 code.

```
\ExplSyntaxOff
```

\glsdoshowtarget Added to glossaries v4.50 so many not be defined. Need to redefine it so use
\def.

```
\def\glsdoshowtarget{\@glsxtr@doshowtarget}
```

\glsxtrshowtargetsymbolright

```
\newcommand{\glsxtrshowtargetsymbolright}{%
 \ifmmode \mbox{\tiny$\triangleleft$}\else {\tiny$\triangleleft$}\fi
}
```

\glsxtrshowtargetsymbolleft

```
\newcommand{\glsxtrshowtargetsymbolleft}{%
 \ifmmode \mbox{\tiny$\triangleright$}\else {\tiny$\triangleright$}\fi
}
```

\glsshowtargetinner Only added to glossaries in v4.50 so may not be defined.

```
\providecommand*{\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
```

\glsshowtargetfont Only added to glossaries in v4.45 so may not be defined.

```
\providecommand*{\glsshowtargetfont}{\ttfamily\footnotesize}
```

\glsshowtargetfonttext Text-block command that checks for math-mode. Only added to glossaries in
v4.50 so may not be defined.

```
\providecommand*{\glsshowtargetfonttext}[1]{%
 \ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
```

\glsshowtargetinnersymleft

```
\newcommand*{\glsshowtargetinnersymleft}[1]{%
 \glsshowtargetinner{#1}\allowbreak\glsxtrshowtargetsymbolleft}
```

`\glsshowtargetinnersymright`

```
\newcommand*{\glsshowtargetinnersymright}[1]{%
\glsxtrshowtargetsymbolright\allowbreak\glsshowtargetinner{#1}}
```

`\glsshowtargetouter` Only added to glossaries in v4.45 so may not be defined.

```
\providecommand*{\glsshowtargetouter}[1]{%
\glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
```

`\@glsshowtarget` Only added to glossaries in v4.32 so may not be defined.

```
\providecommand*{\@glsshowtarget}[1]{}
```

`\glsshowtarget` This command was introduced to glossaries v4.32 so it may not be defined. Therefore it's defined here using `\def`. `\glsshowtargetouter` was introduced in glossaries v4.45, so that also may not be defined.

```
\def\glsshowtarget#1{%
\glsxtrtitleorpdforheading
{%
  \ifmmode
    \nfss@text{\glsxtrshowtargetinner{#1}}%
  \else
    \ifinner
      \glsxtrshowtargetinner{#1}%
    \else
      \glsxtrshowtargetouter{#1}%
    \fi
  \fi
}%
{[#1]}%
{{\protect\glsshowtargetinner{#1}}}%
}
```

`\@glsshowtargetmarkfmt`

```
\newcommand*{\@glsshowtargetmarkfmt}[1]{%
\glsxtrtitleorpdforheading
{%
  \ifmmode \nfss@text{#1}\else #1\fi
}%
{}%
{\ifmmode \nfss@text{#1}\else #1\fi}%
}
```

`\@glsxtr@org@doseeglossary` Save original definition of `\@do@seeglossary`

```
\let\@glsxtr@org@doseeglossary\@do@seeglossary
```

`\@glsxtr@doseeglossary` This doesn't increment the associated counter.

```
\newcommand*{\@glsxtr@doseeglossary}[2]{%
\glsdoifexists{#1}%
{%
```

```
                              \@@glsxtrwrglossmark
                              \@glsxtr@org@doseeglossary{#1}{#2}%
                            }%
                          }
```

`tr@dosee@alsoindex@glossary`

```
                          \newcommand*{\@glsxtr@dosee@alsoindex@glossary}[2]{%
                            \@glsxtr@recordsee{#1}{#2}%
                            \@glsxtr@doseeglossary{#1}{#2}%
                          }
```

`\@glsxtr@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.)

```
                          \let\@glsxtr@org@gloautosee\@glo@autosee
```

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

```
                          \if@glsxtr@autoseeindex
                          \else
                            \ifdef\@glsxtr@org@gloautosee
                            {}%
                            {\PackageError{glossaries-extra}{'autoseeindex=false' package
                             option requires at least v4.30 of glossaries.sty}%
                             {You need to update the glossaries.sty package}%
                            }
                          \fi
```

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

```
                          \ifdef\@glo@autosee
                          {%
                            \renewcommand*{\@glo@autosee}{%
                               \if@glsxtr@autoseeindex\@glsxtr@org@gloautosee\fi}%
                          }%
                          {}
```

`\gls@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.

```
                          \renewcommand*{\gls@checkseeallowed}{%
                           \if@glsxtr@autoseeindex\@gls@see@noindex\fi
                          }
```

Define abbreviations glossaries if required.

```
                          \@glsxtr@abbreviationsdef
                          \let\@glsxtr@abbreviationsdef\relax
```

Setup shortcuts if required.

```
                          \@glsxtr@setupshortcuts
```

35

Redefine `\@glsxtr@redef@forglsentries` if required.

> `\@glsxtr@redef@forglsentries`

\glossariesextrasetup Allow user to set options after the package has been loaded. First modify
`\glsxtr@dooption` so that it now uses `\setupglossaries`:

> `\renewcommand{\glsxtr@dooption}[1]{\setupglossaries{#1}}%`

Disable options that can only be used when the package is loaded:

> `\disable@keys{glossaries-extra.sty}{accsupp}`

Now define the user command:

> `\newcommand*{\glossariesextrasetup}[1]{%`
> `  \let\glsxtr@setup@record\relax`
> `  \let\@glsxtr@setupshortcuts\relax`
> `  \let\@glsxtr@redef@forglsentries\relax`
> `  \let\@glsxtr@doloadprefix\relax`
> `  \setkeys{glossaries-extra.sty}{#1}%`
> `  \@glsxtr@abbreviationsdef`
> `  \let\@glsxtr@abbreviationsdef\relax`
> `  \@glsxtr@setupshortcuts`
> `  \glsxtr@setup@record`
> `  \@glsxtr@redef@forglsentries`
> `  \@glsxtr@doloadprefix`
> `}`

\glsxtr@org@@do@wrglossary Save original definition of `\@@do@wrglossary`.

> `\let\glsxtr@org@@do@wrglossary\@@do@wrglossary`

\glsxtr@@do@wrglossary The new version adds code that can show a marker for debugging and increments
the associated counter if enabled.

> `\newcommand*{\glsxtr@@do@wrglossary}[1]{%`
> `\@@glsxtrwrglossmark`
> `\glsxtr@inc@wrglossaryctr{#1}%`
> `\glsxtr@org@@do@wrglossary{#1}%`
> `}`

\glsxtr@saveentrycounter Save original definition of `\@gls@saveentrycounter`.

> `\let\glsxtr@saveentrycounter\@gls@saveentrycounter`

\@gls@saveentrycounter Change `\@gls@saveentrycounter` so that it only stores the entry counter in-
formation if the indexing is on.

> `\let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter`

\@xp@gls@getcounterprefix This command is provided by glossaries v4.50 so may not be defined. Provide a
similar command in case the new version hasn't been installed.

> `\providecommand*\@xp@gls@getcounterprefix[2]{%`
> `\bgroup`
> `  \glswrglossdisableanchorcmds`
> `  \protected@edef\@do@gls@getcounterprefix{%`

36

```
        \noexpand\egroup
        \noexpand\@gls@getcounterprefix{#1}{#2}%
  }%
  \@do@gls@getcounterprefix
 }
```

\glswrglossdisableanchorcmds

```
        \providecommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
```

\@gls@getcounterprefix This command is provided by the base glossaries package, but is redefined here.
The standard indexing methods don't directly store the hypertarget but instead
need to split it into the counter, prefix and location parts, which can be recon-
stituted in the location list. Unfortunately, not all targets are in this form, so
the links fail. With record=nameref, the complete target name can be saved, so
this modification adjusts the warning.

The expansion should now be performed in \@xp@gls@getcounterprefix.
Any commands that were using \@gls@getcounterprefix directly need to be
use \@xp@gls@getcounterprefix instead.

```
        \renewcommand*\@gls@getcounterprefix[2]{%
          \def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
          \ifx\@gls@thisloc\@gls@thisHloc
            \def\@glo@counterprefix{}%
          \else
            \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
              \def\@glo@tmp{##2}%
              \ifx\@glo@tmp\@empty
                \def\@glo@counterprefix{}%
              \else
                \def\@glo@counterprefix{##1}%
              \fi
            }%
            \@gls@get@counterprefix#2.#1\end@getprefix
```

Warn if no prefix can be formed, unless record=nameref.

```
          \ifx\@glo@counterprefix\@empty
            \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
            \else
              \GlossariesExtraWarning{Hyper target '#2' can't be formed by
              prefixing^^Jlocation '#1'. You need to modify the
              definition of \string\theH\@gls@counter^^Jotherwise you
              will get the warning: "'name{\@gls@counter.#1}' has been^^J
              referenced but does not exist"%
              \ifx\@glsxtr@record@setting\@glsxtr@record@setting@only
              . You may want to consider using record=nameref instead%
              \fi}%
            \fi
          \fi
        \fi
        }
```

37

Provide script dialect hook (does nothing unless redefined by glossaries-extra-bib2gls).

\@glsxtrdialecthook

```
\newcommand*{\@glsxtrdialecthook}{}
```

Set up record option if required.

```
\glsxtr@setup@record
```

Disable preamble-only options and switch on the undefined tag at the start of the document.

```
\AtBeginDocument{%
  \disable@keys{glossaries-extra.sty}{abbreviations,docdef,record}%
  \def\glsxtrundefdebug{\@glsxtrundefdebug}%
  \def\@glsxtrundeftag{\glsxtrundeftag}%
}
```

1.2 Extra Utilities

$$\boxed{\texttt{\textbackslash GlsXtrIfUnusedOrUndefined\{}\langle\textit{label}\rangle\texttt{\}\{}\langle\textit{true}\rangle\texttt{\}\{}\langle\textit{false}\rangle\texttt{\}}}$$

\GlsXtrIfUnusedOrUndefined

Does ⟨*true*⟩ if the entry given by ⟨*label*⟩ is either undefined or hasn't been used (or has had the first use flag reset).

```
\newcommand*{\GlsXtrIfUnusedOrUndefined}[3]{%
  \ifglsentryexists{#1}%
  {\ifbool{glo@\glsdetoklabel{#1}@flag}{#3}{#2}}%
  {#2}%
}
```

Starred form of \ifglossaryexists was only introduced to glossaries v4.46 so provide it if it hasn't been defined.

```
\ifdef\s@ifglossaryexists
{}
{
```

\ifglossaryexists

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

\@ifglossaryexists

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

\s@ifglossaryexists

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

```
    }
```

\glsxtrifemptyglossary

$\boxed{\texttt{\textbackslash glsxtrifemptyglossary\{}\langle type \rangle\texttt{\}\{}\langle true \rangle\texttt{\}\{}\langle false \rangle\texttt{\}}}$

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.)

```
\newcommand{\glsxtrifemptyglossary}[3]{%
  \ifcsdef{glolist@#1}%
  {%
    \ifcsstring{glolist@#1}{,}{#2}{#3}%
  }%
  {%
    \glsxtrundefaction{Glossary type '#1' doesn't exist}{}%
    #2%
  }%
}
```

\GlsXtrIfInGlossary

$\boxed{\texttt{\textbackslash GlsXtrIfInGlossary\{}\langle label \rangle\texttt{\}\{}\langle type \rangle\texttt{\}\{}\langle true \rangle\texttt{\}\{}\langle false \rangle\texttt{\}}}$

Test if the given entry is in the given glossary list. This may not correspond to the type key as the entry may have been copied to the list. Does ⟨*false*⟩ and issues warning if the glossary doesn't exist.

```
\ExplSyntaxOn
\clist_new:N \__glossariesxtr_glolist_clist
\newcommand*{\GlsXtrIfInGlossary}[4]{%
 \tl_if_exist:cTF { glolist@#2 }
  {
    \exp_args:NNv \clist_set:Nn
      \__glossariesxtr_glolist_clist { glolist@#2 }
    \clist_if_in:NnTF
      \__glossariesxtr_glolist_clist { #1 }
      { #3 } { #4 }
 }
 {
  \glsxtrundefaction{Glossary ~ type ~ '#1' ~ doesn't ~ exist}{}
  #4
 }
}
\ExplSyntaxOff
```

\glsxtrifkeydefined Tests if the key given in the first argument has been defined.

```
\newcommand*{\glsxtrifkeydefined}[3]{%
  \key@ifundefined{glossentry}{#1}{#3}{#2}%
```

39

```
                            }
```

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

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

\@glsxtr@provide@storagekey Unstarred version.

```
\newcommand*{\@glsxtr@provide@storagekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{,{#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
```

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

```
    \ifblank{#3}
    {}%
    {%
      \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
    }%
  }%
  {%
```

Provide the no-link command if not already defined.

```
    \ifblank{#3}
    {}%
    {%
      \providecommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
    }%
  }%
}
```

\s@glsxtr@provide@storagekey Starred version.

```
\newcommand*{\s@glsxtr@provide@storagekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname gls@assign@#1@field\endcsname}[2]{%
        \@@gls@expand@field{##1}{#1}{##2}%
      }%
  }%
  {}%
  \@glsxtr@provide@addstoragekey{#1}%
}
```

40

The name of a text-block control sequence can be stored in a field (given by `\GlsXtrFmtField`). This command can then be used with `\glsxtrfmt` `[⟨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`

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

`\GlsXtrFmtDefaultOptions`

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

`\glsxtrfmt`

> `\glsxtrfmt[⟨options⟩]{⟨entry-label⟩}{⟨text⟩}[⟨insert⟩]`

The post-link hook isn't done. This now has a starred form that checks for a final optional argument.

```
\newrobustcmd*{\glsxtrfmt}{\@ifstar\s@glsxtrfmt\@glsxtrfmt}
```

`\@glsxtrfmt` Unstarred form.

```
\newcommand*{\@glsxtrfmt}[3][]{\@@glsxtrfmt{#1}{#2}{#3}{}}
```

`\s@glsxtrfmt` Starred form.

```
\newcommand*{\s@glsxtrfmt}[3][]{%
 \new@ifnextchar[{\s@@glsxtrfmt{#1}{#2}{#3}}%
  {\@@glsxtrfmt{#1}{#2}{#3}{}}%
 }
```

`\s@@glsxtrfmt` Pick up final optional argument.

```
\def\s@@glsxtrfmt#1#2#3[#4]{\@@glsxtrfmt{#1}{#2}{#3}{#4}}
```

`\@@glsxtrfmt` Actual inner working.

```
\newcommand*{\@@glsxtrfmt}[4]{%
```

Since there's no post-link hook to worry about, grouping can be added to provide some protection against nesting (but in general nested link text should be avoided).

```
\begingroup
  \def\glslabel{#2}%
  \glsdoifexistsordo{#2}%
  {%
    \ifglshasfield{\GlsXtrFmtField}{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\relax
      \expandafter\@gls@link\expandafter[\GlsXtrFmtDefaultOptions,#1]{#2}%
        {\glsxtrfmtdisplay{\glscurrentfieldvalue}{#3}{#4}}%
    }%
    {\glsxtrfmtdisplay{@firstofone}{#3}{#4}}%
  }%
  {%
```

Has the default `noindex` been counteracted? If so, this needs `\glsadd` in case `bib2gls` needs to pick up the record.

```
      \begingroup
        \@gls@setdefault@glslink@opts
        \setkeys{glslink}{\GlsXtrFmtDefaultOptions,#1}%
        \ifKV@glslink@noindex\else\glsadd{#2}\fi
      \endgroup
      \glsxtrfmtdisplay{@firstofone}{#3}{#4}%
    }%
  \endgroup
}
```

\Glsxtrfmt

$$\boxed{\texttt{\textbackslash Glsxtrfmt[}\langle \textit{options}\rangle\texttt{]\{}\langle \textit{entry-label}\rangle\texttt{\}\{}\langle \textit{text}\rangle\texttt{\}[}\langle \textit{insert}\rangle\texttt{]}}$$

As `\glsxtrfmt` but applies a sentence-case change to ⟨*text*⟩. This is provided to allow a mapping with mfirstuc v2.08+ in the event that an automated case-change is required.

```
\newrobustcmd*{\Glsxtrfmt}{\@ifstar\s@Glsxtrfmt\@Glsxtrfmt}
\glsmfuaddmap{\glsxtrfmt}{\Glsxtrfmt}
```

\@Glsxtrfmt Unstarred form.

```
\newcommand*{\@Glsxtrfmt}[3][]{\@@glsxtrfmt{#1}{#2}{\glssentencecase{#3}}{}}
```

\s@Glsxtrfmt Starred form.

```
\newcommand*{\s@Glsxtrfmt}[3][]{%
 \new@ifnextchar[{\s@@glsxtrfmt{#1}{#2}{\glssentencecase{#3}}}%
  {\@@glsxtrfmt{#1}{#2}{\glssentencecase{#3}}{}}%
 }
```

\glsxtrfmtdisplay The command used internally by `\glsxtrfmt` to do the actual formatting. The first argument is the control sequence name, the second is the control sequence's argument, the third is the inserted material (if starred form used).

```
\newcommand{\glsxtrfmtdisplay}[3]{\csuse{#1}{#2}#3}
```

\glsxtrentryfmt No link or indexing.

```
\newcommand*{\glsxtrentryfmt}[2]{%
  \glstexorpdfstring{\@glsxtrentryfmt{#1}{#2}}{\glsxtrpdfentryfmt{#1}{#2}}%
}
```

\glsxtrpdfentryfmt Used for the PDF bookmarks.

```
\newcommand*{\glsxtrpdfentryfmt}[2]{#2}
```

\@glsxtrentryfmt

```
\newrobustcmd*{\@glsxtrentryfmt}[2]{%
```

Locally define `\glslabel` in case the helper command needs to access the label.

```
{%
  \protected@edef\glslabel{#1}%
  \glsdoifexistsordo{#1}%
  {%
    \ifglshasfield{\GlsXtrFmtField}{#1}%
    {%
      \csuse{\glscurrentfieldvalue}{#2}%
    }%
    {#2}%
  }%
  {#2}%
}%
}
```

\Glsxtrentryfmt   Sentence-case version.

```
\newcommand*{\Glsxtrentryfmt}[2]{%
  \glstexorpdfstring
    {\@glsxtrentryfmt{#1}{\glssentencecase{#2}}}%
    {\Glsxtrpdfentryfmt{#1}{#2}}%
}
\glsmfuaddmap{\glsxtrentryfmt}{\Glsxtrentryfmt}
```

\Glsxtrpdfentryfmt   Used for the PDF bookmarks.

```
\newcommand*{\Glsxtrpdfentryfmt}[2]{\MFUsentencecase{#2}}
```

\glsxtrfieldlistadd   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.

```
\newcommand*{\glsxtrfieldlistadd}[3]{%
  \listcsadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
```

\glsxtrfieldlistgadd   Similarly but uses `\listcsgadd`.

```
\newcommand*{\glsxtrfieldlistgadd}[3]{%
  \listcsgadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
```

\glsxtrfieldlisteadd   Similarly but uses `\listcseadd`.

```
\newcommand*{\glsxtrfieldlisteadd}[3]{%
  \listcseadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
```

\glsxtrfieldlistxadd   Similarly but uses `\listcsxadd`.

```
\newcommand*{\glsxtrfieldlistxadd}[3]{%
  \listcsxadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
```

Now provide commands to iterate over these lists.

\glsxtrfielddolistloop

```
\newcommand*{\glsxtrfielddolistloop}[2]{%
  \dolistcsloop{glo@\glsdetoklabel{#1}@#2}%
}
```

\glsxtrfieldforlistloop

```
\newcommand*{\glsxtrfieldforlistloop}[3]{%
  \forlistcsloop{#3}{glo@\glsdetoklabel{#1}@#2}%
}
```

\glsxtrfieldformatlist

```
\newrobustcmd*{\glsxtrfieldformatlist}[2]{%
 \begingroup
  \def\@dtl@formatlist@itemsep{}%
  \def\@dtl@formatlist@lastitem{}%
  \def\@dtl@formatlist@prelastitem{}%
  \def\@dtl@formatlist@prelastitemsep{}%
  \forlistcsloop{\@dtl@formatlist@handler}{glo@\glsdetoklabel{#1}@#2}%
  \@dtl@formatlist@prelastitem\@dtl@formatlist@lastitem
 \endgroup
}
```

List element tests:

\glsxtrfieldifinlist First argument label, second argument field, third argument item, fourth true part and fifth false part.

```
\newcommand*{\glsxtrfieldifinlist}[5]{%
  \ifinlistcs{#3}{glo@\glsdetoklabel{#1}@#2}{#4}{#5}%
}
```

\glsxtrfieldxifinlist Expands item.

```
\newcommand*{\glsxtrfieldxifinlist}[5]{%
  \xifinlistcs{#3}{glo@\glsdetoklabel{#1}@#2}{#4}{#5}%
}
```

\glsxtrforcsvfield $\{\langle label\rangle\}\{\langle field\rangle\}\{\langle cs\ handler\rangle\}$

\glsxtrforcsvfield

```
\newcommand*{\glsxtrforcsvfield}{%
  \@ifstar\s@glsxtrforcsvfield\@glsxtrforcsvfield
}
```

\@glsxtrforcsvfield Unstarred version.

```
\newcommand*{\@glsxtrforcsvfield}[3]{%
 \@glsxtrifhasfield{#2}{#1}%
 {%
   \let\glsxtrendfor\@endfortrue
```

44

```
                         \@for\@glsxtr@label:=\glscurrentfieldvalue\do
                         {\expandafter#3\expandafter{\@glsxtr@label}}}%
                        {}%
                        }
```

\s@glsxtrforcsvfield Starred version.
```
                        \newcommand*{\s@glsxtrforcsvfield}[3]{%
                        \s@glsxtrifhasfield{#2}{#1}%
                        {%
                          \let\glsxtrendfor\@endfortrue
                          \@for\@glsxtr@label:=\glscurrentfieldvalue\do
                          {\expandafter#3\expandafter{\@glsxtr@label}}}%
                        {}%
                        }
```

\glsxtrfieldformatcsvlist
```
                        \newrobustcmd*{\glsxtrfieldformatcsvlist}[2]{%
                        \@glsxtrifhasfield{#2}{#1}%
                        {\@dtlformatlist\glscurrentfieldvalue}%
                        {}%
                        }
```

> \GlsXtrIfFieldValueInCsvList{⟨*label*⟩}{⟨*field*⟩}{⟨*list*⟩}
> {⟨*true*⟩}{⟨*false*⟩}

\GlsXtrIfFieldValueInCsvList
```
                        \newcommand*{\GlsXtrIfFieldValueInCsvList}{%
                          \@ifstar\s@GlsXtrIfFieldValueInCsvList\@GlsXtrIfFieldValueInCsvList
                        }
```

Note \DTLifinlist performs one level on the list but not the element.

\@GlsXtrIfFieldValueInCsvList Unstarred version.
```
                        \newcommand*{\@GlsXtrIfFieldValueInCsvList}[5]{%
                        \@glsxtrifhasfield{#2}{#1}%
                        {%
                          \expandafter\DTLifinlist\expandafter{\glscurrentfieldvalue}%
                          {#3}{#4}{#5}%
                        }%
                        {#5}%
                        }
```

\s@GlsXtrIfFieldValueInCsvList Starred version.
```
                        \newcommand*{\s@GlsXtrIfFieldValueInCsvList}[5]{%
                        \s@glsxtrifhasfield{#2}{#1}%
                        {%
                          \expandafter\DTLifinlist\expandafter{\glscurrentfieldvalue}%
                          {#3}{#4}{#5}%
                        }%
```

```
  {#5}%
 }
```

> \GlsXtrIfValueInFieldCsvList{⟨*label*⟩}{⟨*field*⟩}{⟨*value*⟩}
> {⟨*true*⟩}{⟨*false*⟩}

Essentially the reverse. Tests if the given value is in the given field which should contain a comma-separated list.

```
\newcommand*{\GlsXtrIfValueInFieldCsvList}{%
  \@ifstar\s@GlsXtrIfValueInFieldCsvList\@GlsXtrIfValueInFieldCsvList
}
```

Unstarred version.

```
\newcommand*{\@GlsXtrIfValueInFieldCsvList}[5]{%
 \@glsxtrifhasfield{#2}{#1}%
 {%
   \DTLifinlist{#3}{\glscurrentfieldvalue}{#4}{#5}%
 }%
 {#5}%
}
```

Unstarred version.

```
\newcommand*{\s@GlsXtrIfValueInFieldCsvList}[5]{%
 \s@glsxtrifhasfield{#2}{#1}%
 {%
   \DTLifinlist{#3}{\glscurrentfieldvalue}{#4}{#5}%
 }%
 {#5}%
}
```

> \xGlsXtrIfValueInFieldCsvList{⟨*label*⟩}{⟨*field*⟩}{⟨*value*⟩}
> {⟨*true*⟩}{⟨*false*⟩}

As above but fully expand ⟨*value*⟩.

```
\newcommand*{\xGlsXtrIfValueInFieldCsvList}{%
  \@ifstar\s@xGlsXtrIfValueInFieldCsvList\@xGlsXtrIfValueInFieldCsvList
}
```

Unstarred version.

```
\newcommand*{\@xGlsXtrIfValueInFieldCsvList}[5]{%
 \@glsxtrifhasfield{#2}{#1}%
 {%
   \protected@edef\@gls@tmp{#3}%
   \expandafter\DTLifinlist\expandafter{\@gls@tmp}{\glscurrentfieldvalue}{#4}{#5}%
 }%
 {#5}%
}
```

Unstarred version.

```
\newcommand*{\s@xGlsXtrIfValueInFieldCsvList}[5]{%
 \s@glsxtrifhasfield{#2}{#1}%
 {%
   \protected@edef\@gls@tmp{#3}%
   \expandafter\DTLifinlist\expandafter{\@gls@tmp}{\glscurrentfieldvalue}{#4}{#5}%
 }%
 {#5}%
}
```

\glsxtrifhasfield

> $\boxed{\texttt{\textbackslash glsxtrifhasfield}\{\langle field \rangle\}\{\langle label \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}}$

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.

```
\newrobustcmd{\glsxtrifhasfield}{%
  \@ifstar{\s@glsxtrifhasfield}{\@glsxtrifhasfield}%
}
```

\@glsxtrifhasfield Unstarred version adds grouping.

```
\newcommand{\@glsxtrifhasfield}[4]{%
  {\s@glsxtrifhasfield{#1}{#2}{#3}{#4}}%
}
```

\s@glsxtrifhasfield Starred version omits grouping.

```
\newcommand{\s@glsxtrifhasfield}[4]{%
  \letcs{\glscurrentfieldvalue}{glo@\glsdetoklabel{#2}@#1}%
  \ifundef\glscurrentfieldvalue
  {#4}%
  {%
   \ifdefempty\glscurrentfieldvalue{#4}{#3}%
  }%
}
```

\GlsXtrIfFieldNonZero Designed for numeric fields.

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

\@GlsXtrIfFieldNonZero

```
\newcommand{\@GlsXtrIfFieldNonZero}[4]{%
  \@GlsXtrIfFieldCmpNum{#1}{#2}{=}{0}{#4}{#3}%
}
```

\s@GlsXtrIfFieldNonZero

```
\newcommand{\s@GlsXtrIfFieldNonZero}[4]{%
  \s@GlsXtrIfFieldCmpNum{#1}{#2}{=}{0}{#4}{#3}%
}
```

47

\GlsXtrIfFieldEqNum

> \GlsXtrIfFieldEqNum{⟨*field*⟩}{⟨*label*⟩}{⟨*value*⟩}{⟨*true*⟩}
> {⟨*false*⟩}

Designed for numeric fields.

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

\@GlsXtrIfFieldEqNum

```
\newcommand{\@GlsXtrIfFieldEqNum}[5]{%
  \@GlsXtrIfFieldCmpNum{#1}{#2}{=}{#3}{#4}{#5}%
}
```

\s@GlsXtrIfFieldEqNum

```
\newcommand{\s@GlsXtrIfFieldEqNum}[5]{%
  \s@GlsXtrIfFieldCmpNum{#1}{#2}{=}{#3}{#4}{#5}%
}
```

\GlsXtrIfFieldCmpNum

> \GlsXtrIfFieldCmpNum{⟨*field*⟩}{⟨*label*⟩}{⟨*comparison*⟩}
> {⟨*value*⟩}{⟨*true*⟩}{⟨*false*⟩}

Designed for numeric fields.

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

\@GlsXtrIfFieldCmpNum

```
\newcommand{\@GlsXtrIfFieldCmpNum}[6]{%
  {%
    \letcs{\glscurrentfieldvalue}{glo@\glsdetoklabel{#2}@#1}%
    \ifundef\glscurrentfieldvalue
    {\def\glscurrentfieldvalue{0}}%
    {%
     \ifdefempty\glscurrentfieldvalue
     {\def\glscurrentfieldvalue{0}}%
     {}%
    }%
    \ifnum\glscurrentfieldvalue#3#4\relax #5\else #6\fi
  }%
}
```

\s@GlsXtrIfFieldCmpNum

```
\newcommand{\s@GlsXtrIfFieldCmpNum}[6]{%
  \letcs{\glscurrentfieldvalue}{glo@\glsdetoklabel{#2}@#1}%
  \ifundef\glscurrentfieldvalue
  {\def\glscurrentfieldvalue{0}}%
```

```
  {%
   \ifdefempty\glscurrentfieldvalue
   {\def\glscurrentfieldvalue{0}}%
   {}%
  }%
  \ifnum\glscurrentfieldvalue#3#4\relax #5\else #6\fi
 }
```

\GlsXtrIfFieldUndef

> \GlsXtrIfFieldUndef{⟨*field*⟩}{⟨*label*⟩}{⟨*true*⟩}{⟨*false*⟩}

Just uses \ifcsundef.

```
\newcommand{\GlsXtrIfFieldUndef}[2]{%
 \ifcsundef{glo@\glsdetoklabel{#2}@#1}%
}
```

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

```
\newcommand*{\glsxtrusefield}[2]{%
 \@gls@entry@field{#1}{#2}%
}
```

\Glsxtrusefield Provide a user-level alternative to \@Gls@entry@field. Now uses \MFUsentencecase in PDF bookmarks.

```
\newcommand*{\Glsxtrusefield}[2]{%
 \@Glsentryfield{#1}{#2}%
}
\glsmfuaddmap{\glsxtrusefield}{\Glsxtrusefield}
```

\GLSxtrusefield As above but convert to all caps. Note that with mfirstuc v2.08+, \mfirstucMakeUppercase is expandable, so therefore \glsuppercase should also be expandable.

```
\newcommand*{\GLSxtrusefield}[2]{%
 \glsuppercase{\csuse{glo@\glsdetoklabel{#1}@#2}}%
}
\glsmfublocker{\GLSxtrusefield}
```

\glsxtrentryparentname

```
\newcommand*{\glsxtrentryparentname}[1]{%
 \ifcsdef{glo@\glsdetoklabel{#1}@parent}%
 {\csuse{glo@\csuse{glo@\glsdetoklabel{#1}@parent}@name}}%
 {}%
}
```

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

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

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

```
\newcommand*{\glsxtredeffield}[2]{\protected@csedef{glo@\glsdetoklabel{#1}@#2}}
```

\glsxtrapptocsvfield Similar to the above but will append value with a leading comma if the field is already defined. This is used by `bib2gls`. There's no check if the entry has been defined. (Because of the way that bib2gls's save-from-alias etc options are implemented, the entry may not have yet been written to the glstex file when this command is used.)

```
\newcommand*{\glsxtrapptocsvfield}[3]{%
 \ifcsdef{glo@\glsdetoklabel{#1}@#2}%
 {\csappto{glo@\glsdetoklabel{#1}@#2}{,#3}}%
 {\csdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
```

\glsxtrsetfieldifexists

```
\newcommand*{\glsxtrsetfieldifexists}[3]{\glsdoifexists{#1}{#3}}
```

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

```
\newrobustcmd*{\GlsXtrSetField}[3]{%
 \glsxtrsetfieldifexists{#1}{#2}%
 {\csdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
```

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

```
\newrobustcmd*{\GlsXtrLetField}[3]{%
 \glsxtrsetfieldifexists{#1}{#2}%
 {\cslet{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
```

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

```
\newrobustcmd*{\csGlsXtrLetField}[3]{%
 \glsxtrsetfieldifexists{#1}{#2}%
 {\csletcs{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
```

\GlsXtrLetFieldToField 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.

```
\newrobustcmd*{\GlsXtrLetFieldToField}[4]{%
 \glsxtrsetfieldifexists{#1}{#2}%
 {\csletcs{glo@\glsdetoklabel{#1}@#2}{glo@\glsdetoklabel{#3}@#4}}%
}
```

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

```
\newrobustcmd*{\gGlsXtrSetField}[3]{%
 \glsxtrsetfieldifexists{#1}{#2}%
 {\csgdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
```

`\xGlsXtrSetField`

```
\newrobustcmd*{\xGlsXtrSetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\protected@csxdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
```

`\eGlsXtrSetField`

```
\newrobustcmd*{\eGlsXtrSetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\protected@csedef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
```

Version 1.55: Provide LATEX3 commands for testing field values. These don't use `\glsxtrifhasfield`.

`\ExplSyntaxOn`

Test if field is defined. Syntax: {⟨*entry-label*⟩} {⟨*field-label*⟩}

```
\prg_new_conditional:Npnn \glossaries_if_field_exists:nn #1 #2
 { p , T , F , TF }
 {
   \tl_if_exist:cTF { glo@ \glsdetoklabel { #1 } @ #2 }
    { \prg_return_true: }
    { \prg_return_false: }
 }
```

Test if field is set (defined and not empty and not `\relax`). Syntax: {⟨*entry-label*⟩} {⟨*field-label*⟩}

```
\prg_new_conditional:Npnn \glossaries_if_field_set:nn #1 #2
 { p , T , F , TF }
 {
   \tl_if_exist:cTF { glo@ \glsdetoklabel { #1 } @ #2 }
    {
      \tl_if_empty:cTF { glo@ \glsdetoklabel { #1 } @ #2 }
      { \prg_return_false: }
      {
        \tl_if_eq:cNTF
         { glo@ \glsdetoklabel {#1 } @ #2 } \@gls@default@value
         { \prg_return_false: }
         { \prg_return_true: }
      }
    }
    { \prg_return_false: }
 }
```

Test if field is defined and equal to the given token list variable. Syntax: {⟨*entry-label*⟩} {⟨*field-label*⟩} ⟨*tl-var*⟩

```
\prg_new_conditional:Npnn \glossaries_if_field_eq:nnN #1 #2 #3
 { p , T , F , TF }
 {
   \tl_if_exist:cTF { glo@ \glsdetoklabel { #1 } @ #2 }
```

51

```
    {
      \tl_if_eq:cNTF { glo@ \glsdetoklabel {#1 } @ #2 } #3
       { \prg_return_true: }
       { \prg_return_false: }
    }
    { \prg_return_false: }
  }
```

Test if field is defined and equal to the given token list. Syntax: {⟨*entry-label*⟩} {⟨*field-label*⟩} {⟨*tl*⟩}

```
 \prg_new_conditional:Npnn \glossaries_if_field_eq:nnn #1 #2 #3
 { T , F , TF }
 {
   \tl_if_exist:cTF { glo@ \glsdetoklabel { #1 } @ #2 }
    {
      \tl_if_eq:cnTF { glo@ \glsdetoklabel { #1 } @ #2 } { #3 }
       { \prg_return_true: }
       { \prg_return_false: }
    }
    { \prg_return_false: }
 }
```

Test if field is defined and equal to another field (same entry). Syntax: {⟨*entry-label*⟩} {⟨*field-label*⟩} {⟨*field2-label*⟩}

```
 \prg_new_conditional:Npnn \glossaries_if_field_eq_field:nnn #1 #2 #3
 { p , T , F , TF }
 {
   \tl_if_exist:cTF { glo@ \glsdetoklabel { #1 } @ #2 }
    {
      \tl_if_eq:ccTF
        { glo@ \glsdetoklabel { #1 } @ #2 }
        { glo@ \glsdetoklabel { #1 } @ #3 }
       { \prg_return_true: }
       { \prg_return_false: }
    }
    { \prg_return_false: }
 }
```

Test if field is defined and equal to a field in a different entry. Syntax: {⟨*entry-label*⟩} {⟨*field-label*⟩} {⟨*entry2-label*⟩} {⟨*field2-label*⟩}

```
 \prg_new_conditional:Npnn \glossaries_if_field_eq_field:nnnn #1 #2 #3 #4
 { p , T , F , TF }
 {
   \bool_lazy_and:nnTF
     { \tl_if_exist_p:c { glo@ \glsdetoklabel { #1 } @ #2 } }
     { \tl_if_exist_p:c { glo@ \glsdetoklabel { #3 } @ #4 } }
    {
      \tl_if_eq:ccTF
        { glo@ \glsdetoklabel { #1 } @ #2 }
        { glo@ \glsdetoklabel { #3 } @ #4 }
       { \prg_return_true: }
```

```
                 { \prg_return_false: }
               }
             { \prg_return_false: }
           }
```

Recover field content. An error will occur if the field or entry doesn't exist.
Syntax: {⟨*entry-label*⟩} {⟨*field-label*⟩}

```
\cs_new:Nn \glossaries_use_field:nn
  {
    \tl_use:c { glo@ \glsdetoklabel { #1 } @ #2 }
  }
```

```
\ExplSyntaxOff
```

\GlsXtrIfFieldEqStr Starred version uses starred version of \glsxtrifhasfield (that is, no group-
ing).

```
\newcommand*{\GlsXtrIfFieldEqStr}{%
  \@ifstar\s@GlsXtrIfFieldEqStr\@GlsXtrIfFieldEqStr
}
```

\@GlsXtrIfFieldEqStr

```
\newrobustcmd*{\@GlsXtrIfFieldEqStr}[5]{%
  \@glsxtrifhasfield{#1}{#2}%
  {%
    \ifdefstring{\glscurrentfieldvalue}{#3}{#4}{#5}%
  }%
  {#5}%
}
```

\s@GlsXtrIfFieldEqStr

```
\newrobustcmd*{\s@GlsXtrIfFieldEqStr}[5]{%
  \s@glsxtrifhasfield{#1}{#2}%
  {%
    \ifdefstring{\glscurrentfieldvalue}{#3}{#4}{#5}%
  }%
  {#5}%
}
```

\GlsXtrIfFieldEqXpStr Like the above but first expands the string. Starred version uses starred version
of \glsxtrifhasfield (that is, no grouping).

```
\newcommand*{\GlsXtrIfFieldEqXpStr}{%
  \@ifstar\s@GlsXtrIfFieldEqXpStr\@GlsXtrIfFieldEqXpStr
}
```

\@GlsXtrIfFieldEqXpStr

```
\newrobustcmd*{\@GlsXtrIfFieldEqXpStr}[5]{%
  \@glsxtrifhasfield{#1}{#2}%
  {%
    \protected@edef\@gls@tmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
```

```
                          }%
                          {#5}%
                        }
```

\s@GlsXtrIfFieldEqXpStr

```
                        \newrobustcmd*{\s@GlsXtrIfFieldEqXpStr}[5]{%
                        \s@glsxtrifhasfield{#1}{#2}%
                        {%
                           \protected@edef\@gls@tmp{#3}%
                           \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
                        }%
                        {#5}%
                        }
```

\GlsXtrIfXpFieldEqXpStr Like the above but also expands the field value. Starred version uses starred version of \glsxtrifhasfield (that is, no grouping).

```
                        \newcommand*{\GlsXtrIfXpFieldEqXpStr}{%
                        \@ifstar\s@GlsXtrIfXpFieldEqXpStr\@GlsXtrIfXpFieldEqXpStr
                        }
```

\@GlsXtrIfXpFieldEqXpStr

```
                        \newrobustcmd*{\@GlsXtrIfXpFieldEqXpStr}[5]{%
                        \@glsxtrifhasfield{#1}{#2}%
                        {%
                           \protected@edef\@gls@tmp{\glscurrentfieldvalue}%
                           \let\glscurrentfieldvalue\@gls@tmp
                           \protected@edef\@gls@tmp{#3}%
                           \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
                        }%
                        {#5}%
                        }
```

\s@GlsXtrIfXpFieldEqXpStr

```
                        \newrobustcmd*{\s@GlsXtrIfXpFieldEqXpStr}[5]{%
                        \s@glsxtrifhasfield{#1}{#2}%
                        {%
                           \protected@edef\@gls@tmp{\glscurrentfieldvalue}%
                           \let\glscurrentfieldvalue\@gls@tmp
                           \protected@edef\@gls@tmp{#3}%
                           \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
                        }%
                        {#5}%
                        }
```

\GlsXtrForeignText{⟨*entry label*⟩}{⟨*text*⟩}

\GlsXtrForeignText

If a field is used to store a language tag (such as `en-GB` or `de-CH-1996`) then this command uses tracklang's interface to encapsulate ⟨*text*⟩. The field identifying the locale is given by `\GlsXtrForeignTextField`.

```
\ifdef\foreignlanguage
{
  \ifdef\GetTrackedDialectFromLanguageTag
  {
    \newcommand{\GlsXtrForeignText}[2]{%
```

In case this is used inside the argument of `\glsxtrifhasfield`, save and restore `\glscurrentfieldvalue`.

```
      \let\@glsxtr@org@currentfieldvalue\glscurrentfieldvalue
      \glsxtrifhasfield{\GlsXtrForeignTextField}{#1}%
      {%
        \expandafter\GetTrackedDialectFromLanguageTag\expandafter
          {\glscurrentfieldvalue}{\@glsxtr@dialect}%
        \let\@glsxtr@locale\glscurrentfieldvalue
        \let\glscurrentfieldvalue\@glsxtr@org@currentfieldvalue
        \ifdefempty\@glsxtr@dialect
        {%
```

An exact match hasn't been found. A partial match can only be obtained with at least tracklang v1.3.6.

```
          \ifundef\TrackedDialectClosestSubMatch
          {%
            \GlossariesExtraWarning{Can't obtain dialect label
              (tracklang v1.3.6+ required)}%
          }%
          {\let\@glsxtr@dialect\TrackedDialectClosestSubMatch}%
        }%
        {}%
        \ifdefempty\@glsxtr@dialect
        {%
```

No tracked dialect found for the root language.

```
        }%
        {%
```

Check if there's a caption hook for the given dialect label.

```
          \ifcsundef{captions\@glsxtr@dialect}{}%
          {%
```

Dialect label not recognised. Check if there's a known mapping.

```
            \IfTrackedDialectHasMapping{\@glsxtr@dialect}%
            {%
              \edef\@glsxtr@dialect{%
                \GetTrackedDialectToMapping{\@glsxtr@dialect}}%
```

Does a caption hook exist for this?

```
              \ifcsundef{captions\@glsxtr@dialect}{}%
              {%
```

No mapping. Try root language label instead.

```
                      \ifcsundef{captions\@tracklang@lang}{}%
                      {%
                        \let\@glsxtr@dialect\@tracklang@lang
                      }%
                    }%
                  }%
                  {%
```

No mapping. Try root language label instead.

```
                      \ifcsundef{captions\@tracklang@lang}{}%
                      {%
                        \let\@glsxtr@dialect\@tracklang@lang
                      }%
                    }%
                  }%
                }%
                \ifdefempty\@glsxtr@dialect
                {%
                  \GlsXtrUnknownDialectWarning{\@glsxtr@locale}{\@tracklang@lang}%
                  #2%
                }%
                {\foreignlanguage{\@glsxtr@dialect}{#2}}%
              }%
              {#2}% key not set
            }
          }
          {
            \newcommand{\GlsXtrForeignText}[2]{%
              \GlossariesExtraWarning{Can't encapsulate foreign text:
                tracklang v1.3.6+ required}%
              #2%
            }
          }
        }
        {
```

\foreignlanguage isn't defined so just do ⟨*text*⟩.

```
          \newcommand{\GlsXtrForeignText}[2]{#2}
        }
```

\GlsXtrForeignTextField   This is the user2 field by default but may be redefined as required.

```
        \newcommand*{\GlsXtrForeignTextField}{userii}
```

\GlsXtrUnknownDialectWarning

```
        \newcommand*{\GlsXtrUnknownDialectWarning}[2]{%
          \GlossariesExtraWarning{Can't determine valid dialect label
            for locale '#1' (root language: #2)}%
        }
```

\glsxtrpageref Like \glsrefentry but references the page number instead (if entry counting is on). The base glossaries package only introduced \GlsEntryCounterLabelPrefix in version 4.38, so it may not be defined.

```
\ifdef\GlsEntryCounterLabelPrefix
{%
  \newcommand*{\glsxtrpageref}[1]{%
    \ifglsentrycounter
      \pageref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \ifglssubentrycounter
        \pageref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
      \else
        \gls{#1}%
      \fi
    \fi
  }
}%
{%
  \newcommand*{\glsxtrpageref}[1]{%
    \ifglsentrycounter
      \pageref{glsentry-\glsdetoklabel{#1}}%
    \else
      \ifglssubentrycounter
        \pageref{glsentry-\glsdetoklabel{#1}}%
      \else
        \gls{#1}%
      \fi
    \fi
  }
}%
```

\apptoglossarypreamble

```
\newcommand{\apptoglossarypreamble}[2][\glsdefaulttype]{%
  \ifcsdef{glolist@#1}%
  {%
   \ifcsundef{@glossarypreamble@#1}%
   {\csdef{@glossarypreamble@#1}{}}%
   {}%
   \csappto{@glossarypreamble@#1}{#2}%
  }%
  {%
    \GlossariesExtraWarning{Glossary '#1' is not defined}%
  }%
}
```

\pretoglossarypreamble

```
\newcommand{\pretoglossarypreamble}[2][\glsdefaulttype]{%
  \ifcsdef{glolist@#1}%
  {%
```

```
      \ifcsundef{@glossarypreamble@#1}%
      {\csdef{@glossarypreamble@#1}{}}%
      {}%
      \cspreto{@glossarypreamble@#1}{#2}%
    }%
    {%
      \GlossariesExtraWarning{Glossary '#1' is not defined}%
    }%
  }
```

`\preglossarypreamble` Typo in command name resulted in `\preglossarypreamble` being defined when it should have been called `\pretoglossarypreamble`. Old name retained for backward compatibility.

```
\newcommand{\preglossarypreamble}{\pretoglossarypreamble}
```

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.

`\@p@glossarysection` Phantom section only needs to be added for starred section commands.

```
\renewcommand*{\@p@glossarysection}[2]{%
  \glsclearpage
  \ifdefempty\@@glossarysecstar
  {%
    \csname\@@glossarysec\endcsname{#2}%
  }%
  {%
    \phantomsection
    \@gls@toc{#1}{\@@glossarysec}%
    \csname\@@glossarysec\endcsname*{#2}%
  }%
  \@@glossaryseclabel
}
```

The original `\@gls@entry@field` causes a problem for undefined entries when used in section headings or captions. Since entries must be defined with just the base package this isn't a significant issue, but it will cause a problem with bib2gls where no entries are defined on the first LaTeX call, so redefine `\@gls@entry@field` to use `\csuse` instead of `\csname`.

`\@gls@entry@field`
> `\@gls@entry@field{`⟨*label*⟩`}{`⟨*field*⟩`}`

This command was introduced to glossaries version 4.03 but older versions are likely to be incompatible with glossaries-extra.

```
\ifdef\@gls@entry@field
{
```

```
      \renewcommand*{\@gls@entry@field}[2]{\csuse{glo@\glsdetoklabel{#1}@#2}}
    }
    {}
```

> **\ifglsused**{⟨*label*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

\ifglsused

In the event that undefined entries should trigger a warning rather than an error, \ifglsused needs to be modified to check for existence. If the boolean variable is undefined, then its state is indeterminate and is neither true nor false, so neither ⟨*true part*⟩ nor ⟨*false*⟩ part will be performed if ⟨*label*⟩ is undefined. See also \GlsXtrIfUnusedOrUndefined.

```
  \renewcommand*{\ifglsused}[3]{%
    \glsdoifexists{#1}{\ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}}%
  }
```

\@gls@noexpand@field Add check for encapinnerfmt, encapnocase and encapnocaseinnerfmt

```
  \renewcommand{\@@gls@noexpand@field}[3]{%
  \glsifcategoryattributehasitem{\@glo@category}{encapnocaseinnerfmt}{#2}%
  {%
    \csxdef{glo@#1@#2}{\noexpand\NoCaseChange{\noexpand\glsxtrgenentrytextfmt
      {\expandonce{#3}}}}%
    \glsexclapplyinnerfmtfield{#1}{#2}%
  }%
  {%
    \glsifcategoryattributehasitem{\@glo@category}{encapnocase}{#2}%
    {%
    \glsifcategoryattributehasitem{\@glo@category}{encapinnerfmt}{#2}%
    {%
      \csxdef{glo@#1@#2}{\noexpand\NoCaseChange{\noexpand\glsxtrgenentrytextfmt
        {\expandonce{#3}}}}%
      \glsexclapplyinnerfmtfield{#1}{#2}%
    }%
    {%
      \csxdef{glo@#1@#2}{\noexpand\NoCaseChange{\expandonce{#3}}}%
    }%
  }%
  {%
    \glsifcategoryattributehasitem{\@glo@category}{encapinnerfmt}{#2}%
    {%
      \csxdef{glo@#1@#2}{\noexpand\glsxtrgenentrytextfmt{\expandonce{#3}}}%
      \glsexclapplyinnerfmtfield{#1}{#2}%
    }%
    {%
      \expandafter\global\expandafter\let\csname glo@#1@#2\endcsname#3%
    }%
  }%
  }%
  }
```

Add check for encapinnerfmt, encapnocase and encapnocaseinnerfmt

```
\renewcommand{\@@gls@expand@field}[3]{%
\glsifcategoryattributehasitem{\@glo@category}{encapnocaseinnerfmt}{#2}%
{%
  \protected@csxdef{glo@#1@#2}{\noexpand\NoCaseChange
   {\noexpand\glsxtrgenentrytextfmt{#3}}}%
  \glsexclapplyinnerfmtfield{#1}{#2}%
}%
{%
  \glsifcategoryattributehasitem{\@glo@category}{encapnocase}{#2}%
  {%
   \glsifcategoryattributehasitem{\@glo@category}{encapinnerfmt}{#2}%
   {%
     \protected@csxdef{glo@#1@#2}{\noexpand\NoCaseChange
      {\noexpand\glsxtrgenentrytextfmt{#3}}}%
     \glsexclapplyinnerfmtfield{#1}{#2}%
   }%
   {%
     \protected@csxdef{glo@#1@#2}{\noexpand\NoCaseChange{#3}}%
   }%
  }%
  {%
   \glsifcategoryattributehasitem{\@glo@category}{encapinnerfmt}{#2}%
   {%
     \protected@csxdef{glo@#1@#2}{\noexpand\glsxtrgenentrytextfmt{#3}}%
     \glsexclapplyinnerfmtfield{#1}{#2}%
   }%
   {%
     \protected@csxdef{glo@#1@#2}{#3}%
   }%
  }%
}%
}
```

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 \glsxtrpostlongdescription instead.

```
\renewcommand*{\longnewglossaryentry}{%
\@ifstar\@glsxtr@s@longnewglossaryentry\@glsxtr@longnewglossaryentry
}
```

Starred version.

```
\newcommand{\@glsxtr@s@longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
   \bgroup
```

```
                              \let\@org@newglossaryentryprehook\@newglossaryentryprehook
                              \long\def\@newglossaryentryprehook{%
                                \long\def\@glo@desc{#3}%
                                \@org@newglossaryentryprehook
                              }%
                              \renewcommand*{\gls@assign@desc}[1]{%
                                  \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
                                  \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@descplural}%
                              }
                              \gls@defglossaryentry{#1}{#2}%
                            \egroup
                        }%
                      }
```

@glsxtr@longnewglossaryentry Unstarred version.

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

The following is different from the base glossaries.sty:

```
                                \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@descplural}%
                            }
                            \gls@defglossaryentry{#1}{#2}%
                          \egroup
                      }%
                    }
```

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

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

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

\newignoredglossary Redefine to check for star.

```
                      \renewcommand{\newignoredglossary}{%
                       \@ifstar\glsxtr@s@newignoredglossary\glsxtr@org@newignoredglossary
                      }
```

lsxtr@org@newignoredglossary The original definition is patched to check for existence.

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

61

```
                                \glsxtrundefaction{Glossary type '#1' already exists}{}%
                              }%
                              {%
                                \ifdefempty\@ignored@glossaries
                                {%

                                  \protected@edef\@ignored@glossaries{#1}%
                                }%
                                {%
                                  \protected@eappto\@ignored@glossaries{,#1}%
                                }%
                                \csgdef{glolist@#1}{,}%
                                \ifcsundef{gls@#1@entryfmt}%
                                {%
                                  \defglsentryfmt[#1]{\glsentryfmt}%
                                }%
                                {}%
                                \ifdefempty\@gls@nohyperlist
                                {%
                                  \renewcommand*{\@gls@nohyperlist}{#1}%
                                }%
                                {%

                                  \protected@eappto\@gls@nohyperlist{,#1}%
                                }%
                              }%
                            }
\glsxtr@s@newignoredglossary  Starred form.
                            \newcommand*{\glsxtr@s@newignoredglossary}[1]{%
                              \ifcsdef{glolist@#1}
                              {%
                                \glsxtrundefaction{Glossary type '#1' already exists}{}%
                              }%
                              {%

                                \ifdefempty\@ignored@glossaries
                                {%
                                  \protected@edef\@ignored@glossaries{#1}%
                                }%
                                {%
                                  \protected@eappto\@ignored@glossaries{,#1}%
                                }%
                                \csgdef{glolist@#1}{,}%
                                \ifcsundef{gls@#1@entryfmt}%
                                {%
                                  \defglsentryfmt[#1]{\glsentryfmt}%
                                }%
                                {}%
                              }%
                            }
```

62

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

```
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \ifcsdef{@glotype@#1@title}%
      {\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}%
      {\def\glossarytoctitle{\glossarytitle}}%
    }%
  }%
}
{
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{@glotype@#1@title}%
    {\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}%
    {\def\glossarytoctitle{\glossarytitle}}%
  }
}
```

\provideignoredglossary As above but won't do anything if the glossary already exists.

```
\newcommand{\provideignoredglossary}{%
 \@ifstar\glsxtr@s@provideignoredglossary\glsxtr@provideignoredglossary
}
```

\glsxtr@provideignoredglossary Unstarred version.

```
\newcommand*{\glsxtr@provideignoredglossary}[1]{%
 \ifcsdef{glolist@#1}
 {}%
 {%

   \ifdefempty\@ignored@glossaries
   {%
     \protected@edef\@ignored@glossaries{#1}%
   }%
   {%
     \protected@eappto\@ignored@glossaries{,#1}%
   }%
   \csgdef{glolist@#1}{,}%
   \ifcsundef{gls@#1@entryfmt}%
   {%
     \defglsentryfmt[#1]{\glsentryfmt}%
   }%
   {}%
   \ifdefempty\@gls@nohyperlist
   {%
```

63

```
                                   \renewcommand*{\@gls@nohyperlist}{#1}%
                               }%
                               {%

                                   \protected@eappto\@gls@nohyperlist{,#1}%
                               }%
                           }%
                       }

tr@s@provideignoredglossary  Starred form.
                       \newcommand*{\glsxtr@s@provideignoredglossary}[1]{%
                         \ifcsdef{glolist@#1}
                         {}%
                         {%

                             \ifdefempty\@ignored@glossaries
                             {%
                               \protected@edef\@ignored@glossaries{#1}%
                             }%
                             {%
                               \protected@eappto\@ignored@glossaries{,#1}%
                             }%
                             \csgdef{glolist@#1}{,}%
                             \ifcsundef{gls@#1@entryfmt}%
                             {%
                               \defglsentryfmt[#1]{\glsentryfmt}%
                             }%
                             {}%
                         }%
                       }

     \glsxtrcopytoglossary  Adds an entry label to another glossary list. First argument is entry label.
                            Second argument is glossary label. The starred version globally adds the entry
                            label.
                       \newcommand*{\glsxtrcopytoglossary}{%
                         \@ifstar\s@glsxtrcopytoglossary\@glsxtrcopytoglossary
                       }

   \@glsxtrcopytoglossary

                       \newcommand*{\@glsxtrcopytoglossary}[2]{%
                         \glsdoifexists{#1}%
                         {%
                           \ifcsdef{glolist@#2}
                           {%

                               \protected@cseappto{glolist@#2}{#1,}%
                           }%
                           {%
                               \glsxtrundefaction{Glossary type '#2' doesn't exist}{}%
                           }%
```

```
        }%
      }
```

`\s@glsxtrcopytoglossary`

```
\newcommand*{\s@glsxtrcopytoglossary}[2]{%
  \glsdoifexists{#1}%
  {%
    \ifcsdef{glolist@#2}
    {%
      \protected@csxappto{glolist@#2}{#1,}%
    }%
    {%
      \glsxtrundefaction{Glossary type '#2' doesn't exist}{}%
    }%
  }%
}
```

1.3.1 Existence Checks

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

```
\renewcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{\glsxtr@doifexists{#1}}%
}
```

`\glsxtr@doifexists` Provide a robust command for the error/warning in case `\glsdoifexists` is expanded.

```
\newrobustcmd{\glsxtr@doifexists}[1]{%
```

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

```
  \protected@edef\glslabel{\glsdetoklabel{#1}}%
  \expandafter\glsxtrundefdebug\expandafter
    {\expandafter\detokenize\expandafter{\glslabel}}%
  \glsxtrundefaction{Glossary entry '\glslabel'
  has not been defined}{You need to define a glossary entry before
  you can reference it.}%
}
```

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

```
\renewcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{\glsxtr@doifnoexists{#1}}{#2}%
}
```

`\glsxtr@doifnoexists` Provide a robust command for the error/warning in case `\glsdoifnoexists` is expanded.

```
\newrobustcmd{\glsxtr@doifnoexists}[1]{%
  \glsxtrundefaction{Glossary entry '\glsdetoklabel{#1}'
    has already been defined}%
}
```

`\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.

```
\ifdef\glsdoifexistsordo
{%
  \renewcommand{\glsdoifexistsordo}[3]{%
    \ifglsentryexists{#1}{#2}%
    {%
      \glsxtrundefaction{Glossary entry '\glsdetoklabel{#1}'
      has not been defined}{You need to define a glossary entry
      before you can use it.}%
      #3%
    }%
  }%
}
{%
  \glsxtr@warnonexistsordo\glsdoifexistsordo
  \newcommand{\glsdoifexistsordo}[3]{%
    \ifglsentryexists{#1}{#2}%
    {%
      \glsxtrundefaction{Glossary entry '\glsdetoklabel{#1}'
      has not been defined}{You need to define a glossary entry
      before you can use it.}%
      #3%
    }%
  }%
}
```

`\doifglossarynoexistsordo` Similarly for `\doifglossarynoexistsordo`.

```
\ifdef\doifglossarynoexistsordo
{%
  \renewcommand{\doifglossarynoexistsordo}[3]{%
    \ifglossaryexists*{#1}%
    {%
      \glsxtrundefaction{Glossary type '#1' already exists}{}%
      #3%
    }%
    {#2}%
  }%
}
{%
  \glsxtr@warnonexistsordo\doifglossarynoexistsordo
  \newcommand{\doifglossarynoexistsordo}[3]{%
    \ifglossaryexists*{#1}%
    {%
      \glsxtrundefaction{Glossary type '#1' already exists}{}%
      #3%
    }%
    {#2}%
```

```
      }%
   }
```

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).

\@newglossaryentryposthook Hook into end of \newglossaryentry to add "see" value as a field.

```
\appto\@newglossaryentryposthook{%
  \ifdefvoid\@glo@see
  {\csxdef{glo@\@glo@label @see}{}}%
  {%
    \csxdef{glo@\@glo@label @see}{\@glo@see}%
    \if@glsxtr@autoseeindex
      \@glsxtr@autoindexcrossrefs
    \fi
  }%
}
\appto\@gls@keymap{,{see}{see}}
```

\glsxtrseelistsencap{⟨content⟩}

\glsxtrseelistsencap

Encapsulates cross-reference list.

```
\newcommand*{\glsxtrseelistsencap}[1]{\space #1}
```

\glsxtrseelistsdelim Delimiter in cross-reference list.

```
\newcommand*{\glsxtrseelistsdelim}{, }
```

\glsxtrseelists{⟨label⟩}

\glsxtrseelists

```
\newcommand*{\glsxtrseelists}[1]{%
  \glsdoifexists{#1}%
  {%
    \def\@glsxtr@seelists{}%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@see}%
    \ifdefempty\@glo@see
    {}%
    {%
      \protected@edef\@glsxtr@seelists{%
        \noexpand\glsxtr@usesee\@glo@see\noexpand\@end@glsxtr@usesee
      }%
    }%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@seealso}%
    \ifdefempty\@glo@see
```

```
            {}%
            {%
              \ifdefempty\@glsxtr@seelists{}%
              {\appto\@glsxtr@seelists{\glsxtrseelistsdelim}}%
              \protected@edef\@glsxtr@seelists{%
                \noexpand\glsxtruseseealsoformat{\@glo@see}%
              }%
            }%
            \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@alias}%
            \ifdefempty\@glo@see
            {}%
            {%
              \ifdefempty\@glsxtr@seelists{}%
              {\appto\@glsxtr@seelists{\glsxtrseelistsdelim}}%
              \protected@edef\@glsxtr@seelists{%
                \noexpand\glsxtruseseeformat{\noexpand\seename}{\@glo@see}%
              }%
            }%
            \ifdefempty\@glsxtr@seelists{}%
            {\glsxtrseelistsencap\@glsxtr@seelists}%
          }%
        }
```

\glsxtrusesee   Apply \glsseeformat to the see key if not empty.

```
                \newcommand*{\glsxtrusesee}[1]{%
                  \glsdoifexists{#1}%
                  {%
                    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@see}%
                    \ifdefempty\@glo@see
                    {}%
                    {%
                      \expandafter\glsxtr@usesee\@glo@see\@end@glsxtr@usesee
                    }%
                  }%
                }
```

\glsxtr@usesee

```
                \newcommand*{\glsxtr@usesee}[1][\seename]{%
                  \@glsxtr@usesee[#1]%
                }
```

\@glsxtr@usesee

```
                \def\@glsxtr@usesee[#1]#2\@end@glsxtr@usesee{%
                  \glsxtruseseeformat{#1}{#2}%
                }
```

\glsxtruseseeformat   The format used by \glsxtrusesee. The first argument is the tag (such
                      as \seename). The second argument is the comma-separated list of cross-
                      referenced labels.

```
\newcommand*{\glsxtruseseeformat}[2]{%
  \glsseeformat[#1]{#2}{}%
}
```

\glsseeitemformat glossaries originally defined `\glsseeitemformat` to use `\glsentryname` but in
v3.0 this was switched to use `\glsentrytext` due to problems occurring with
the name field being sanitized. Since this is no longer a problem, glossaries-extra
restored the original definition as it makes more sense to use the name in the
cross-reference list. Unfortunately this doesn't take style changes into account,
so as from v1.42, this now uses `\glsfmttext` and `\glsfmtname` instead. (The
text field is chosen rather than the short field to allow for the "noshort" styles.)

```
\renewcommand*{\glsseeitemformat}[1]{%
  \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
}
```

\glsxtrhiername

> **\glsxtrhiername{⟨label⟩}**

Displays the hierarchical name for the given entry. The cross-reference for-
mat `\glsseeitemformat` may be redefined to use this command to show the
hierarchy, if required. This now uses `\glsfmttext` and `\glsfmtname` instead of
`\glsaccessshort` and `\glsaccessname` to allow for style formatting.

```
\newcommand*{\glsxtrhiername}[1]{%
  \glstexorpdfstring
  {\@glsxtrhiername{#1}}%
  {\glsentryname{#1}}%
}
```

\@glsxtrhiername Provide robust inner command.

```
\newrobustcmd*{\@glsxtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {\expandafter\glsxtrhiername\expandafter
      {\glscurrentfieldvalue}\glsxtrhiernamesep}%
    {}%
    \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
  }%
}
```

\Glsxtrhiername

> **\Glsxtrhiername{⟨label⟩}**

As above but displays the top-level name with an initial capital.

```
\newcommand*{\Glsxtrhiername}[1]{%
  \glstexorpdfstring
  {\@Glsxtrhiername{#1}}%
```

```

```
      {\MFUsentencecase{\glsentryname{#1}}}%
    }
```

**\@Glsxtrhiername**  Provide robust inner command.

```
\newrobustcmd*{\@Glsxtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {%
      \expandafter\Glsxtrhiername\expandafter
        {\glscurrentfieldvalue}\glsxtrhiernamesep
      \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
    }%
    {\ifglshasshort{#1}{\Glsfmttext{#1}}{\Glsfmtname{#1}}}%
  }%
}
\glsmfuaddmap{\glsxtrhiername}{\Glsxtrhiername}
```

\GlsXtrhiername

$\boxed{\texttt{\textbackslash GlsXtrhiername\{}\langle\mathit{label}\rangle\texttt{\}}}$

As above but converts the first letter of each name to a capital. (Note that this isn't applying title case, just capitalising the start of each hierarchical element.)

```
\newcommand*{\GlsXtrhiername}[1]{%
  \glstexorpdfstring
  {\@GlsXtrhiername{#1}}%
  {\glsentryname{#1}}%
}
```

\@GlsXtrhiername Provide robust inner command.

```
\newrobustcmd*{\@GlsXtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {\expandafter\GlsXtrhiername\expandafter
      {\glscurrentfieldvalue}\glsxtrhiernamesep}%
    {}%
    \ifglshasshort{#1}{\Glsfmttext{#1}}{\Glsfmtname{#1}}%
  }%
}
\glsmfublocker{\GlsXtrhiername}
```

\GLSxtrhiername

$\boxed{\texttt{\textbackslash GLSxtrhiername\{}\langle\mathit{label}\rangle\texttt{\}}}$

As above but displays the top-level name in all-caps.

```
\newcommand*{\GLSxtrhiername}[1]{%
```

```
            \glstexorpdfstring
            {\@GLSxtrhiername{#1}}%
            {\GLSxtrusefield{#1}{name}}%
          }
```

\@GLSxtrhiername Provide robust inner command.

```
            \newrobustcmd*{\@GLSxtrhiername}[1]{%
              \glsdoifexists{#1}%
              {%
                \glsxtrifhasfield{parent}{#1}%
                {%
                  \expandafter\GLSxtrhiername\expandafter
                   {\glscurrentfieldvalue}\glsxtrhiernamesep
                  \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
                }%
                {\ifglshasshort{#1}{\GLSfmttext{#1}}{\GLSfmtname{#1}}}%
              }%
            }
            \glsmfublocker{\GLSxtrhiername}
```

┌───┐
│   │
│   \GLSXTRhiername{⟨label⟩}   │
│   │
└───┘

\GLSXTRhiername

As above but displays all names in all-caps.

```
            \newcommand*{\GLSXTRhiername}[1]{%
              \glstexorpdfstring
              {\@GLSXTRhiername{#1}}%
              {\GLSxtrusefield{#1}{name}}%
            }
```

\@GLSXTRhiername Provide robust inner command.

```
            \newrobustcmd*{\@GLSXTRhiername}[1]{%
             \glsdoifexists{#1}%
             {%
                \glsxtrifhasfield{parent}{#1}%
                {\expandafter\GLSXTRhiername\expandafter
                    {\glscurrentfieldvalue}\glsxtrhiernamesep}%

                {}%
                \ifglshasshort{#1}{\GLSfmttext{#1}}{\GLSfmtname{#1}}%
             }%
            }
            \glsmfublocker{\GLSXTRhiername}
```

\glsxtrhiernamesep Separator used in \glsxtrhiername and variants.

```
            \newcommand*{\glsxtrhiernamesep}{\,{\small$\triangleright$}\,}
```

\glsxtruseseealso Apply \glsseeformat to the seealso key if not empty. There's no optional tag
                   to worry about here.

```
\newcommand*{\glsxtruseseealso}[1]{%
  \glsdoifexists{#1}%
  {%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@seealso}%
    \ifdefempty\@glo@see
    {}%
    {%
      \expandafter\glsxtruseseealsoformat\expandafter{\@glo@see}%
    }%
  }%
}
```

\glsxtrusealias Apply \glsseeformat to the alias key if not empty. There's no optional tag
to worry about here. The value also isn't a comma-separated list, but use the
same interface.

```
\newcommand*{\glsxtrusealias}[1]{%
  \glsdoifexists{#1}%
  {%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@alias}%
    \ifdefempty\@glo@see
    {}%
    {%
```

Expansion isn't necessary because the value is a single label not a list.

```
      \glsxtruseseeformat{\seename}{\@glo@see}%
    }%
  }%
}
```

\glsxtruseseealsoformat The format used by \glsxtruseseealso. The argument is the comma-
separated list of cross-referenced labels.

```
\newcommand*{\glsxtruseseealsoformat}[1]{%
  \glsseeformat[\seealsoname]{#1}{}%
}
```

\glsxtrseelist Fully expands argument before passing to \glsseelist. (The argument to
\glsseelist must be a comma-separated list of entry labels.)

```
\newrobustcmd{\glsxtrseelist}[1]{%
  \protected@edef\@glo@tmp{\noexpand\glsseelist{#1}}\@glo@tmp
}
```

\glsseelist Redefine to make \glsseelist more flexible.

```
\renewrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \let\@glsseeitem\@glsxtr@seefirstitem
  \let\@glsseelastsep\glsseelastsep
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
```

```
        \@gls@dolast
      \else
        \@gls@donext
      \fi
      \expandafter\@glsseeitem\expandafter{\@gls@thislabel}%
      \let\@gls@dolast\@glsseelastsep
      \let\@gls@donext\glsseesep
      \let\@glsseeitem\@glsxtr@seeitem
      \let\@glsseelastsep\glsseelastoxfordsep
    }%
  }
```

Separator between the tag and the list in \glsxtrtaggedlist

```
    \newcommand{\glsxtrtaggedlistsep}{\space}
```

> \glsxtrtaggedlist{⟨singular tag⟩}{⟨plural
> tag⟩}{⟨label prefix⟩}{⟨label list⟩}

Similar to the above but the tag is selected depending on how many items there are in the list.

```
  \newrobustcmd*{\glsxtrtaggedlist}[4]{%
   \begingroup
   \protected@edef\@gls@taggedlist@labels{#4}%
   \let\@gls@dolast\relax
   \let\@gls@donext\relax
   \let\@glsseeitem\@glsxtr@seefirstitem
   \let\@glsseelastsep\glsseelastsep
   \def\@gls@taggedlist@content{}%
   \let\@gls@taggedlist@tag\relax
   \@for\@gls@thislabel:=\@gls@taggedlist@labels\do{%
     \ifx\@xfor@nextelement\@nnil
       \ifx\@gls@dolast\relax
       \else
         \eappto\@gls@taggedlist@content{\expandonce\@gls@dolast}%
       \fi
     \else
       \ifx\@gls@dolast\relax
       \else
         \eappto\@gls@taggedlist@content{\expandonce\@gls@donext}%
       \fi
     \fi
     \protected@eappto\@gls@taggedlist@content{\noexpand\@glsseeitem
       {#3\@gls@thislabel}}%
     \let\@gls@dolast\@glsseelastsep
     \let\@gls@donext\glsseesep
     \let\@glsseeitem\@glsxtr@seeitem
     \let\@glsseelastsep\glsseelastoxfordsep
     \ifx\@gls@taggedlist@tag\relax
```

```
                         \def\@gls@taggedlist@tag{#1\glsxtrtaggedlistsep}%
                         \else
                          \def\@gls@taggedlist@tag{#2\glsxtrtaggedlistsep}%
                         \fi
                       }%
                       \@gls@taggedlist@tag\@gls@taggedlist@content
                      \endgroup
                    }
```

\@glsxtr@seeitem

```
                    \newcommand*{\@glsxtr@seeitem}[1]{%
                     \glsxtrifmulti{#1}{\mglsseeitem{#1}}{\glsseeitem{#1}}%
                    }
```

\@glsxtr@seefirstitem

```
                    \newcommand*{\@glsxtr@seefirstitem}[1]{%
                     \glsxtrifmulti{#1}{\mglsseefirstitem{#1}}{\glsseefirstitem{#1}}%
                    }
```

\mglsseeitem   Multi-entry cross-reference

```
                    \newcommand*{\mglsseeitem}[1]{%
                     \mglsname[all={noindex},setup={hyper=allmain}]{#1}%
                    }
```

\mglsseefirstitem   Multi-entry cross-reference

```
                    \newcommand*{\mglsseefirstitem}{\mglsseeitem}
```

\glsseefirstitem

```
                    \newcommand*{\glsseefirstitem}{\glsseeitem}
```

\glsseelastoxfordsep

```
                    \newcommand*{\glsseelastoxfordsep}{\glsseelastsep}
```

\seealsoname   In case this command hasn't been defined. Languages packages actually provide
                \alsoname so use that if it's defined.

```
                    \ifdef\alsoname
                    {\providecommand{\seealsoname}{\alsoname}}
                    {\providecommand{\seealsoname}{see also}}
```

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

```
                    \ifdef\@xdycrossrefhook
                    {
```

Add the cross-reference class definition to the hook.

```
                    \appto\@xdycrossrefhook{%
                      \write\glswrite{(define-crossref-class \string"seealso\string"
                        :unverified )}%
```

```
    \write\glswrite{(markup-crossref-list
        :class \string"seealso\string"^^J\space\space\space
        :open \string"\string\glsxtruseseealsoformat\glsopenbrace\string"
        :close \string"\glsclosebrace\string")}%
    }
```

Append to class list.

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

This essentially works like `\@do@seeglossary` but uses the `seealso` class. This
doesn't increment the associated counter.

```
    \newrobustcmd*{\glsxtrindexseealso}[2]{%
      \glsxtr@wrglossary@encap{#1}
      {%
        \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
          \@glsxtr@recordsee{#1}{#2}%
        \fi
        \glsdoifexists{#1}%
        {%
          \@@glsxtrwrglossmark
          \def\@gls@xref{#2}%
          \@onelevel@sanitize\@gls@xref
          \@gls@checkmkidxchars\@gls@xref
          \gls@glossary{\csname glo@#1@type\endcsname}{%
            (indexentry
              :tkey (\csname glo@#1@index\endcsname)
              :xref (\string"\@gls@xref\string")
              :attr \string"seealso\string"
            )
          }%
        }%
      }%
    }
  }
  {
```

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

```
    \newrobustcmd*{\glsxtrindexseealso}{\glssee[\seealsoname]}
  }
```

The alias key should be set to the label of the synonymous entry. The seealso
key essentially behaves like see=[\seealsoname]{⟨*xr-list*⟩}. 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`.

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

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

```
\define@key{glossentry}{alias}{%
  \gls@set@xr@key{alias}{\@glo@alias}{#1}%
}
\define@key{glossentry}{seealso}{%
  \gls@set@xr@key{seealso}{\@glo@seealso}{#1}%
}
```

Add to the key mappings.

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

Set the default value.

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

Assign the field values.

```
\appto\@newglossaryentryposthook{%
  \ifdefvoid\@glo@seealso
  {\csxdef{glo@\@glo@label @seealso}{}}%
  {%
    \csxdef{glo@\@glo@label @seealso}{\@glo@seealso}%
    \if@glsxtr@autoseeindex
      \@glsxtr@autoindexcrossrefs
    \fi
  }%
```

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

```
  \ifdefvoid\@glo@alias
  {\csxdef{glo@\@glo@label @alias}{}}%
  {%
    \csxdef{glo@\@glo@label @alias}{\@glo@alias}%
    \glsxtraliashook{\@glo@label}%
  }%
}
```

Provide user-level commands to access the values.

\glsxtralias

```
\newcommand*{\glsxtralias}[1]{\@gls@entry@field{#1}{alias}}
```

\glsxtrseealsolabels

```
\newcommand*{\glsxtrseealsolabels}[1]{\@gls@entry@field{#1}{seealso}}
```

Add to the \@glo@autosee hook.

```
\appto\@glo@autoseehook{%
  \ifdefvoid\@glo@alias
  {%
    \ifdefvoid\@glo@seealso
    {}%
    {%
```

76

```
      \protected@edef\@do@glssee{\noexpand\glsxtrindexseealso
        {\@glo@label}{\@glo@seealso}}%
      \@do@glssee
    }%
  }%
  {%
```

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

```
    \ifdefvoid\@glo@see
    {%
      \protected@edef\@do@glssee{\noexpand\glssee{\@glo@label}{\@glo@alias}}%
      \@do@glssee
      \glsxtraliashook{\@glo@label}%
    }%
    {}%
  }%
}%
}
{
```

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

\glsxtralias

```
\glsaddstoragekey*{alias}{}{\glsxtralias}
```

\glsxtrseealsolabels

```
\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.

\@newglossaryentryposthook Append to the hook to check for the alias and seealso keys.

```
\appto\@newglossaryentryposthook{%
  \ifcsvoid{glo@\@glo@label @alias}%
  {%
    \ifcsvoid{glo@\@glo@label @seealso}%
    {}%
    {%

      \protected@edef\@do@glssee{\noexpand\glsxtrindexseealso
        {\@glo@label}{\csuse{glo@\@glo@label @seealso}}}%
      \@do@glssee
    }%
  }%
  {%
```

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

```
    \ifdefvoid\@glo@see
    {%
```

77

```
                          \protected@edef\@do@glssee{\noexpand\glssee
                            {\@glo@label}{\csuse{glo@\@glo@label @alias}}}%
                          \@do@glssee
                        }%
                        {}%
                      }%
                    }

                  }
```

\glsxtraliashook Provide a hook that's used when the alias field is provided.

```
\newcommand*{\glsxtraliashook}[1]{}
```

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

```
\AtEndDocument{\if@glsxtrindexcrossrefs\glsxtraddallcrossrefs\fi}
```

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

```
\newcommand*{\glsxtraddallcrossrefs}{%
  \forallglossaries{\@glo@type}%
  {%
    \forglsentries[\@glo@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}{\glsxtraddunusedxrefs{\@glo@label}}{}%
    }%
  }%
}
```

\glsxtraddunusedxrefs Added user-level command in case user wants to redefine \glsxtraddallcrossrefs

```
\newcommand*{\glsxtraddunusedxrefs}[1]{\expandafter\@glsxtr@addunusedxrefs\expandafter{#1}}
```

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

```
\newcommand*{\@glsxtr@addunusedxrefs}[1]{%
  \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@see}%
  \ifdefvoid\@glo@see
  {}%
  {%
    \expandafter\glsxtr@addunused\@glo@see\@end@glsxtr@addunused
  }%
  \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@seealso}%
  \ifdefvoid\@glo@see
  {}%
  {%
    \expandafter\glsxtr@addunused\@glo@see\@end@glsxtr@addunused
  }%
}
```

`\glsxtr@addunused` Adds all the entries if they haven't been used.

```
\newcommand*{\glsxtr@addunused}[1][]{%
  \@glsxtr@addunused
}
```

`\@glsxtr@addunused` Adds all the entries if they haven't been used.

```
\def\@glsxtr@addunused#1\@end@glsxtr@addunused{%
 \@for\@glsxtr@label:=#1\do
 {%
   \glsxtrifmulti\@glsxtr@label
   {%
     \letcs\@glsxtr@labellist{@gls@combined@\@glsxtr@label @list}%
     \@for\@glsxtr@multilabel:=\@glsxtr@labellist\do
     {\@glsxtr@addunused\@glsxtr@multilabel\@end@glsxtr@addunused}%
   }%
   {%
     \ifglsused{\@glsxtr@label}{}%
     {%
       \glsadd[format=glsxtrunusedformat]{\@glsxtr@label}%
       \glsunset{\@glsxtr@label}%
       \expandafter\@glsxtr@addunusedxrefs\expandafter{\@glsxtr@label}%
     }%
   }%
 }%
}
```

`\glsxtrunusedformat`

```
\newcommand*{\glsxtrunusedformat}[1]{\unskip}
```

1.3.2 Document Definitions

`\gls@begindocdefs` This command was only introduced to glossaries v4.37, so it may not be defined.
If it has been defined, redefine it to check `\@glsxtr@docdefval` so that it only
inputs the .glsdefs file if docdef=true.

```
\ifdef\gls@begindocdefs
{%
  \renewcommand*{\gls@begindocdefs}{%
    \ifnum\@glsxtr@docdefval=1\relax
      \@gls@enablesavenonumberlist
      \edef\@gls@restoreat{%
        \noexpand\catcode`\noexpand\@=\number\catcode`\@\relax}%
      \makeatletter
      \InputIfFileExists{\jobname.glsdefs}{}{}%
      \@gls@restoreat
      \undef\@gls@restoreat
      \gls@defdocnewglossaryentry
    \else
      \ifnum\@glsxtr@docdefval=3\relax
```

79

The docdef=atom package option has been set. Create the .glsdefs file for the autocomplete support but don't read it.

```
\@gls@enablesavenonumberlist
\let\gls@checkseeallowed\relax
\let\newglossaryentry\new@atom@glossaryentry
\global\newwrite\@gls@deffile
\immediate\openout\@gls@deffile=\jobname.glsdefs
```

Write all currently defined entries.

```
\forallglsentries{\@glsentry}{\@gls@writedef{\@glsentry}}%
  \fi
  \fi
}
}
{%
  \ifnum\@glsxtr@docdefval=3\relax
    \PackageError{glossaries-extra}{Package option
    'docdef=\@glsxtr@docdefsetting' requires at least version 4.37
    of the base glossaries.sty package}{}
  \fi
}
```

\new@atom@glossaryentry

```
\newrobustcmd{\new@atom@glossaryentry}[2]{%
  \gls@defglossaryentry{#1}{#2}%
  \@gls@writedef{#1}%
}
```

\makenoidxglossaries   Modify \makenoidxglossaries so that it automatically sets docdef=false (unless the restricted setting is on) and disables the docdef key. This command isn't allowed with the record option.

```
\let\glsxtr@orgmakenoidxglossaries\makenoidxglossaries
\renewcommand{\makenoidxglossaries}{%
  \def\glsindexingsetting{noidx}%
  \@domakeglossaries
  {%
    \ifdefequal\@glsxtr@record@setting\@glsxtr@record@setting@off
    {%
      \glsxtr@orgmakenoidxglossaries
```

Add marker to \@do@seeglossary but don't increment associated counter.

```
\renewcommand{\@do@seeglossary}[2]{%
  \@@glsxtrwrglossmark

  \protected@edef\@gls@label{\glsdetoklabel{##1}}%
  \protected@write\@auxout{}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {%
```

```
                 \string\glsseeformat##2{}%
             }%
          }%
        \@gls@noidx@addtorefs{\@gls@label}%
      }%
```

Check for docdefs=restricted:

```
      \if@glsxtrdocdefrestricted
```

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

```
        \renewcommand*{\@gls@reference}{%
          \@glsxtr@reference
        }%
```

Adjust hook to ensure unindexed parents are added to the referenced list, otherwise a rerun warning will keep occurring.

```
        \appto\@newglossaryentryposthook{%
          \@glsxtr@restricted@newentryhook
        }%
      \else
```

Disable document definitions.

```
        \@glsxtrdocdeffalse
      \fi
      \disable@keys{glossaries-extra.sty}{docdef}%
    }%
    {%
      \PackageError{glossaries-extra}{\string\makenoidxglossaries\space
       not permitted\MessageBreak
       with record=\@glsxtr@record@setting\space package option}%
      {You may only use \string\makenoidxglossaries\ space with the
       record=off option}%
    }%
    \let\gls@warn@noidx@incompatible\@gls@warn@noidx@incompatible
   }%
  }
```

`\@glsxtr@reference` Check for new command.

```
    \ExplSyntaxOn
    \cs_if_exist:NTF \__glossaries_add_glsref:nnn
     {
       \newcommand*\@glsxtr@reference[3]{
         \__glossaries_add_glsref:nnn { #1 } { #2 } { #3 }
       }
     }
     {
       \newcommand*\@glsxtr@reference[3]{
         \ifcsundef{@glsref@#1}{\csgdef{@glsref@#1}{}}{}%
         \ifinlistcs{#2}{@glsref@#1}%
         {}%
```

81

```
      {\listcsgadd{@glsref@#1}{#2}}%
      \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
      {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
      {}%
      \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
    }
  }
```

```
\cs_if_exist:NTF \__glossaries_noidx_add_parent_of_glsrefd:n
 {
   \newcommand \@glsxtr@restricted@newentryhook
    {
      \bool_lazy_and:nnT
        { \tl_if_exist_p:c { glo@ \glslabel @parent } }
        { \bool_not_p:n { \tl_if_empty_p:c { glo@ \glslabel @parent } } }
        {
          \exp_args:Nv \__glossaries_noidx_add_parent_of_glsrefd:n
            { glo@ \glslabel @ parent }
        }
    }
 }
 {
   \newcommand \@glsxtr@restricted@newentryhook { }
 }
\ExplSyntaxOff
```

\@gls@noidx@addtorefs This was only added to glossaries v4.59, so provide it in case an older version is installed.

```
\providecommand\@gls@noidx@addtorefs[1]{}%
```

\gls@warn@noidx@incompatible

```
\newcommand*{\gls@warn@noidx@incompatible}[2]{}
```

\@gls@warn@noidx@incompatible

```
\newcommand*{\@gls@warn@noidx@incompatible}[2]{%
  #2\GlossariesExtraWarning{#1\space is incompatible with \string\makenoidxglossaries}%
}
```

@noidxmakegloss@incompatible

```
\newcommand*{\gls@warn@noidxmakegloss@incompatible}[2]{%
  \gls@warn@noidx@incompatible{#1}{#2}%
  \gls@warn@makegloss@incompatible{#1}{#2}%
}
```

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

```
\renewcommand*{\gls@defdocnewglossaryentry}{%
  \ifcase\@glsxtr@docdefval
```

docdef=false:

```
    \renewcommand*{\newglossaryentry}[2]{%
      \PackageError{glossaries-extra}{Glossary entries must
        be \MessageBreak defined in the preamble with \MessageBreak
        package option 'docdef=false'\MessageBreak(consider using
        'docdef=restricted')}{Move your glossary definitions to
        the preamble. You can also put them in a \MessageBreak separate file
        and load them with \string\loadglsentries.}%
    }%
  \or
```

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

```
    \let\gls@checkseeallowed\relax
    \let\newglossaryentry\new@glossaryentry
  \else
```

Restricted mode just needs to allow the see value.

```
    \let\gls@checkseeallowed\relax
  \fi
 }%
```

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.

\GlsXtrEnableOnTheFly

```
\newcommand*{\GlsXtrEnableOnTheFly}{%
  \@ifstar\@sGlsXtrEnableOnTheFly\@GlsXtrEnableOnTheFly
}
```

\@sGlsXtrEnableOnTheFly 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.)

```
\newcommand*{\@sGlsXtrEnableOnTheFly}{%
  \renewcommand*{\glsdetoklabel}[1]{%
    \expandafter\@glsxtr@ifcsstart\string##1 \@glsxtr@end@
    {%
      \expandafter\detokenize\expandafter{##1}%
    }%
    {\detokenize{##1}}%
  }%
  \@GlsXtrEnableOnTheFly
}
\def\@glsxtr@ifcsstart#1#2\@glsxtr@end@#3#4{%
  \expandafter\if\glsbackslash#1%
```

```
                          #3%
                        \else
                          #4%
                        \fi
                       }
```

\glsxtrstarflywarn

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

\@GlsXtrEnableOnTheFly

```
                       \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

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

\glsxtr

```
                       \newcommand*{\glsxtr}[1][]{%
                        \def\glsxtr@keylist{##1}%
                        \@glsxtr
                       }
```

\@glsxtr

```
                       \newcommand*{\@glsxtr}[2][]{%
                        \ifglsentryexists{##2}%
                        {%
                          \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
                        }%
                        {%
                          \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
                            description={\nopostdesc},##1}%
                        }%
                        \expandafter\gls\expandafter[\glsxtr@keylist]{##2}%
                       }
```

\Glsxtr

```
                       \newcommand*{\Glsxtr}[1][]{%
                        \def\glsxtr@keylist{##1}%
                        \@Glsxtr
                       }
                       \glsmfuaddmap{\glsxtr}{\Glsxtr}
```

\@Glsxtr

```
\newcommand*{\@Glsxtr}[2][]{%
 \ifglsentryexists{##2}%
 {%
   \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
 }%
 {%
   \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
     description={\nopostdesc},##1}%
 }%
 \expandafter\Gls\expandafter[\glsxtr@keylist]{##2}%
}
```

\glsxtrpl

```
\newcommand*{\glsxtrpl}[1][]{%
 \def\glsxtr@keylist{##1}%
 \@glsxtrpl
}
```

\@glsxtrpl

```
\newcommand*{\@glsxtrpl}[2][]{%
 \ifglsentryexists{##2}%
 {%
   \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
 }%
 {%
   \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
     description={\nopostdesc},##1}%
 }%
 \expandafter\glspl\expandafter[\glsxtr@keylist]{##2}%
}
```

\Glsxtrpl

```
\newcommand*{\Glsxtrpl}[1][]{%
 \def\glsxtr@keylist{##1}%
 \@Glsxtrpl
}
\glsmfuaddmap{\glsxtrpl}{\Glsxtrpl}
```

\@Glsxtrpl

```
\newcommand*{\@Glsxtrpl}[2][]{%
 \ifglsentryexists{##2}
 {%
   \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
 }%
 {%
   \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
     description={\nopostdesc},##1}%
 }%
```

```
                              \expandafter\Glspl\expandafter[\glsxtr@keylist]{##2}%
                            }
```

\GlsXtrWarning

```
                            \newcommand*{\GlsXtrWarning}[2]{%
                              \def\@glsxtr@optlist{##1}%
                              \@onelevel@sanitize\@glsxtr@optlist
                              \GlossariesExtraWarning{The options '\@glsxtr@optlist' have
                              been ignored for entry '##2' as it has already been defined}%
                            }
```

Disable commands after the glossary:

```
                            \renewcommand\@printglossary[2]{%
                              \def\@glsxtr@printglossopts{##1}%
                              \@glsxtr@orgprintglossary{##1}{##2}%
                              \def\@glsxtr{\@glsxtr@disabledflycommand\glsxtr}%
                              \def\@glsxtrpl{\@glsxtr@disabledflycommand\glsxtrpl}%
                              \def\@Glsxtr{\@glsxtr@disabledflycommand\Glsxtr}%
                              \def\@Glsxtrpl{\@glsxtr@disabledflycommand\Glsxtrpl}%
                            }
```

\@glsxtr@disabledflycommand

```
                            \newcommand*{\@glsxtr@disabledflycommand}[1]{%
                              \PackageError{glossaries-extra}%
                              {\string##1\space can't be used after any of the \MessageBreak
                               glossaries have been displayed}%
                              {The on-the-fly commands enabled by
                               \string\GlsXtrEnableOnTheFly\space may only be used \MessageBreak
                               before the glossaries. If you want to use any entries \MessageBreak
                               after any of the glossaries, you must use the standard \MessageBreak
                               method of first defining the entry and then using the \MessageBreak
                               entry with commands like \string\gls}%
                              \@@glsxtr@disabledflycommand
                            }%
                            \newcommand*{\@@glsxtr@disabledflycommand}[2][]{##2}
```

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

```
                            \let\GlsXtrEnableOnTheFly\relax
                          }
                          \@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.

\@glsxtr@current@style Initialise the current style to the default style.

```
                            \newcommand*{\@glsxtr@current@style}{\@glossary@default@style}
```

86

\glsxtrpreglossarystyle A hook to initialise default definitions for style commands.

```
\newcommand{\glsxtrpreglossarystyle}{%
  \renewcommand*{\glssubgroupheading}[4]{\glsgroupheading{##4}}%
}
```

Modify \setglossarystyle to set \@glsxtr@current@style and reset \glssubgroupheading in case the style doesn't support it.

\setglossarystyle

```
\renewcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries-extra}{Glossary style '#1' undefined}{}%
  }%
  {%
    \glsxtrpreglossarystyle
    \csname @glsstyle@#1\endcsname
```

Only set the current style if it exists.

```
    \protected@edef\@glsxtr@current@style{#1}%
  }%
```

Set this as the default, if a default hasn't been set.

```
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
```

In case we have an old version of glossaries:

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

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

```
\ifdef\glslistdottedwidth
{%
  \ifdim\glslistdottedwidth=.5\hsize
    \setlength{\glslistdottedwidth}{-\dimexpr\maxdimen-1sp\relax}
    \AtBeginDocument{%
      \ifdim\glslistdottedwidth=-\dimexpr\maxdimen-1sp\relax
       \setlength{\glslistdottedwidth}{.5\columnwidth}%
      \fi
    }%
  \fi
}
{}%
```

Similarly for \glsdescwidth:

87

\glsdescwidth

```
\ifdef\glsdescwidth
{%
  \ifdim\glsdescwidth=.6\hsize
    \setlength{\glsdescwidth}{-\dimexpr\maxdimen-1sp\relax}
    \AtBeginDocument{%
      \ifdim\glsdescwidth=-\dimexpr\maxdimen-1sp\relax
       \setlength{\glsdescwidth}{.6\columnwidth}%
      \fi
    }%
  \fi
}
{}%
```

and for `\glspagelistwidth`:

\glspagelistwidth

```
\ifdef\glspagelistwidth
{%
  \ifdim\glspagelistwidth=.1\hsize
    \setlength{\glspagelistwidth}{-\dimexpr\maxdimen-1sp\relax}
    \AtBeginDocument{%
      \ifdim\glspagelistwidth=-\dimexpr\maxdimen-1sp\relax
       \setlength{\glspagelistwidth}{.1\columnwidth}%
      \fi
    }%
  \fi
}
{}%
```

\glossaryentrynumbers Has the nonnumberlist option been used?

```
\def\org@glossaryentrynumbers#1{#1\gls@save@numberlist{#1}}%
\ifx\org@glossaryentrynumbers\glossaryentrynumbers
  \glsnonumberlistfalse
  \renewcommand*{\glossaryentrynumbers}[1]{%
    \ifglsentryexists{\glscurrententrylabel}%
    {%
      \@glsxtrpreloctag
      \GlsXtrFormatLocationList{#1}%
      \@glsxtrpostloctag
      \gls@save@numberlist{#1}%
    }{}%
  }%
\else
  \glsnonumberlisttrue
  \renewcommand*{\glossaryentrynumbers}[1]{%
    \ifglsentryexists{\glscurrententrylabel}%
    {%
      \gls@save@numberlist{#1}%
    }{}%
```

```
        }%
      \fi
```

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

```
\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.

\GlsXtrEnablePreLocationTag

```
\newcommand*{\GlsXtrEnablePreLocationTag}[2]{%
  \let\@glsxtrpreloctag\@@glsxtrpreloctag
  \let\@glsxtrpostloctag\@@glsxtrpostloctag
  \renewcommand*{\@glsxtr@pagetag}{#1}%
  \renewcommand*{\@glsxtr@pagestag}{#2}%
  \renewcommand*{\@glsxtr@savepreloctag}[2]{%
    \csgdef{@glsxtr@preloctag@##1}{##2}%
  }%
  \renewcommand*{\@glsxtr@doloctag}{%
    \ifcsundef{@glsxtr@preloctag@\glscurrententrylabel}%
    {%
      \GlossariesWarning{Missing pre-location tag for '\glscurrententrylabel'.
        Rerun required}%
    }%
    {%
      \csuse{@glsxtr@preloctag@\glscurrententrylabel}%
    }%
  }%
}
\@onlypreamble\GlsXtrEnablePreLocationTag
```

\@glsxtrpreloctag

```
\newcommand*{\@@glsxtrpreloctag}{%
  \let\@glsxtr@org@delimN\delimN
  \let\@glsxtr@org@delimR\delimR
  \let\@glsxtr@org@glsignore\glsignore
```

\gdef is required as the delimiters may occur inside a scope.

```
  \gdef\@glsxtr@thisloctag{\@glsxtr@pagetag}%
  \renewcommand*{\delimN}{%
    \gdef\@glsxtr@thisloctag{\@glsxtr@pagestag}%
    \@glsxtr@org@delimN}%
  \renewcommand*{\delimR}{%
    \gdef\@glsxtr@thisloctag{\@glsxtr@pagestag}%
    \@glsxtr@org@delimR}%
```

```
                      \renewcommand*{\glsignore}[1]{%
                        \gdef\@glsxtr@thisloctag{\relax}%
                        \@glsxtr@org@glsignore{##1}}%
                      \@glsxtr@doloctag
                    }

\@glsxtrpreloctag
                    \newcommand*{\@glsxtrpreloctag}{}

\@glsxtr@pagetag
                    \newcommand*{\@glsxtr@pagetag}{}%

\@glsxtr@pagestag
                    \newcommand*{\@glsxtr@pagestag}{}%

\@@glsxtrpostloctag
                    \newcommand*{\@@glsxtrpostloctag}{%
                       \let\delimN\@glsxtr@org@delimN
                       \let\delimR\@glsxtr@org@delimR
                       \let\glsignore\@glsxtr@org@glsignore
                       \protected@write\@auxout{}%
                        {\string\@glsxtr@savepreloctag{\glscurrententrylabel}{\@glsxtr@thisloctag}}%
                    }

\@glsxtrpostloctag
                    \newcommand*{\@glsxtrpostloctag}{}

\@glsxtr@preloctag
                    \newcommand*{\@glsxtr@savepreloctag}[2]{}
                    \protected@write\@auxout{}{%
                      \string\providecommand\string\@glsxtr@savepreloctag[2]{}}

\@glsxtr@doloctag
                    \newcommand*{\@glsxtr@doloctag}{}
```

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

```
\renewcommand*{\KV@printgloss@nonumberlist}[1]{%
 \XKV@plfalse
 \XKV@sttrue
 \XKV@checkchoice[\XKV@resa]{#1}{true,false}%
 {%
   \csname glsnonumberlist\XKV@resa\endcsname
   \ifglsnonumberlist
     \def\glossaryentrynumbers##1{\gls@save@numberlist{##1}}%
   \else
     \def\glossaryentrynumbers##1{%
       \@glsxtrpreloctag
       \GlsXtrFormatLocationList{##1}%
```

```
            \@glsxtrpostloctag
            \gls@save@numberlist{##1}}%
        \fi
    }%
  }
```

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.

```
\renewcommand*{\glsentryfmt}{%
  \ifglshasshort{\glslabel}{\glssetabbrvfmt{\glscategory{\glslabel}}}{}%
  \glsifregular{\glslabel}%
  {\glsxtrregularfont{\glsgenentryfmt}}%
  {%
    \ifglshasshort{\glslabel}%
    {\glsxtrabbreviationfont{\glsxtrgenabbrvfmt}}%
    {\glsxtrregularfont{\glsgenentryfmt}}%
  }%
}
```

\glsxtrregularfont Font used for regular entries.

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

\glsxtrabbreviationfont Font used for abbreviation entries.

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

Some formatting commands (such as highlighting or letter spacing) may require expandable content in the argument, so also provide a formatting command for use within \glsgenentryfmt for those instances.

\glsxtrdefaultentrytextfmt This is the default command that \glsxtrgenentrytextfmt is initialised to within \@gls@link.

```
\newcommand{\glsxtrdefaultentrytextfmt}[1]{#1}
```

\glsxtrattrentrytextfmt Provide a convenient command that applies the formatting according to the category attribute. This isn't used by default as this inner formatting should rarely be needed and increases complexity.

```
\newcommand{\glsxtrattrentrytextfmt}[1]{%
  \glshasattribute{\glslabel}{innertextformat}%
  {%
    \csuse{\glsgetattribute{\glslabel}{innertextformat}}{#1}%
  }%
```

```
      {#1}%
   }
```

\glsxtrgenentrytextfmt This command is a user-level command to allow it to be included in custom formats or styles but it should not be redefined at the user level as it's redefined within `\@gls@link` (similar to other style commands, such as `\glscapscase`. Redefine `\glsxtrdefaultentrytextfmt` to change the default definition for this command.

```
\newcommand*{\glsxtrgenentrytextfmt}{\glsxtrdefaultentrytextfmt}
```

\glsfmtfield

> \glsfmtfield{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}{⟨*field*⟩}

Provide a convenient way of applying a formatting command to the actual field contents. No check for existence.

Note this command intentionally isn't robust, as it's possible that a user may want to redefine an abbreviation command to use `\MakeLowercase`, for example, to use smallcaps when abbreviations have been defined with the short version in capitals. Using `\newrobustcmd` will break that case.

```
\newcommand*{\glsfmtfield}[4]{%
  \expandafter\expandafter\expandafter
  #2\expandafter\expandafter\expandafter
   {\csname glo@\glsdetoklabel{#3}@#4\endcsname #1}%
}
```

\Glsfmtfield

> \Glsfmtfield{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}{⟨*field*⟩}

As above but convert first letter to uppercase. Note that if the formatting command can go outside of `\makefirstuc` then it can simply be applied around the appropriate command that expands to the field value. For example,

```
%\emph{\Glsentrytext{label}}
%
```

instead of

```
%\Glsfmtfield{}{\emph}{sample}{text}
%
```

Note this command intentionally isn't robust for the same reason as above. The expansion allows `\makefirstuc` to pick up any mappings or blockers before the content is passed to `\MFUsentencecase`.

```
\newcommand*{\Glsfmtfield}[4]{%
  \ifx#2\@firstofone
   \expandafter\expandafter\expandafter
   \glssentencecase\expandafter\expandafter\expandafter
    {%
```

```
            \csname glo@\glsdetoklabel{#3}@#4\endcsname #1%
          }%
        \else
         \expandafter\expandafter\expandafter
         \glssentencecase\expandafter\expandafter\expandafter
         {%
           \expandafter\expandafter\expandafter
           #2\expandafter\expandafter\expandafter
             {\csname glo@\glsdetoklabel{#3}@#4\endcsname #1}%
         }%
        \fi
      }
      \glsmfuaddmap{\glsfmtfield}{\Glsfmtfield}
```

<div style="border:1px solid; background:#fbf8cc; border-radius:8px; padding:6px">

$\GLSfmtfield\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}\{\langle field\rangle\}$

</div>

\GLSfmtfield

As above but convert all to uppercase. The expansion is in case we have an older kernel.

```
      \newcommand*{\GLSfmtfield}[4]{%
        \ifx#2\@firstofone
         \expandafter\expandafter\expandafter
         \glsuppercase\expandafter\expandafter\expandafter
         {%
           \csname glo@\glsdetoklabel{#3}@#4\endcsname #1%
         }%
        \else
         \expandafter\expandafter\expandafter
         \glsuppercase\expandafter\expandafter\expandafter
         {%
           \expandafter\expandafter\expandafter
           #2\expandafter\expandafter\expandafter
             {\csname glo@\glsdetoklabel{#3}@#4\endcsname #1}%
         }%
        \fi
      }
      \glsmfublocker{\GLSfmtfield}
```

\glsfmtinsert Formats \glsinsert.

```
      \newcommand*{\glsfmtinsert}{%
        \ifdefempty\glsinsert{}%
        {\expandafter\glsxtrgenentrytextfmt\expandafter{\glsinsert}}%
      }
```

\GLSfmtinsert As above but all caps.

```
      \newcommand*{\GLSfmtinsert}{%
        \ifdefempty\glsinsert{}%
        {%
          \expandafter\glsuppercase\expandafter
```

```
                {\expandafter\glsxtrgenentrytextfmt\expandafter{\glsinsert}}%
        }%
    }
```

\glsifapplyinnerfmtfield

> $\glsifapplyinnerfmtfield\{\langle label\rangle\}\{\langle field\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}$

Does $\langle true\rangle$ if \glsgenentryfmt should encapsulate the given field with the inner format otherwise does $\langle false\rangle$.

```
\newcommand*{\glsifapplyinnerfmtfield}[4]{%
 \ifcsundef{@glo@\glsdetoklabel{#1}@innerfmt@fields}%
 {#3}%
 {\xifinlistcs{#2}{@glo@\glsdetoklabel{#1}@innerfmt@fields}{#4}{#3}}%
}
```

\glsexclapplyinnerfmtfield Adds the field to the exclusion list. This typically means that the field value already contains the inner formatting.

```
\newcommand*{\glsexclapplyinnerfmtfield}[2]{%
 \listcseadd{@glo@\glsdetoklabel{#1}@innerfmt@fields}{#2}%
}
```

\glsgenentryfmt Redefine to use \glsxtrgenentrytextfmt

```
\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
```

Plural form

```
      \glscapscase
      {%
```

Don't adjust case

```
        \ifglsused\glslabel
        {%
```

Subsequent use

```
          \glsifapplyinnerfmtfield{\glslabel}{plural}%
          {%
            \expandafter\glsaccessfmtplural\expandafter{\glsinsert}%
              {\glsxtrgenentrytextfmt}{\glslabel}%
          }%
          {\glsaccessplural{\glslabel}\glsfmtinsert}%
        }%
        {%
```

First use

```
          \glsifapplyinnerfmtfield{\glslabel}{firstpl}%
          {%
            \expandafter\glsaccessfmtfirstplural\expandafter{\glsinsert}%
```

```
              {\glsxtrgenentrytextfmt}{\glslabel}%
          }%
          {\glsaccessfirstplural{\glslabel}\glsfmtinsert}%
        }%
      }%
      {%
```

Make first letter upper case

```
        \ifglsused\glslabel
        {%
```

Subsequent use.

```
          \glsifapplyinnerfmtfield{\glslabel}{plural}%
          {%
            \expandafter\Glsaccessfmtplural\expandafter
             {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
          }%
          {\Glsaccessplural{\glslabel}\glsfmtinsert}%
        }%
        {%
```

First use

```
          \glsifapplyinnerfmtfield{\glslabel}{firstpl}%
          {%
            \expandafter\Glsaccessfmtfirstplural\expandafter
             {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
          }%
          {\Glsaccessfirstplural{\glslabel}\glsfmtinsert}%
        }%
      }%
      {%
```

Make all upper case

```
        \ifglsused\glslabel
        {%
```

Subsequent use

```
          \glsifapplyinnerfmtfield{\glslabel}{plural}%
          {%
            \expandafter\GLSaccessfmtplural\expandafter
              {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
          }%
          {\GLSaccessplural{\glslabel}\GLSfmtinsert}%
        }%
        {%
```

First use

```
          \glsifapplyinnerfmtfield{\glslabel}{firstpl}%
          {%
            \expandafter\GLSaccessfmtfirstplural\expandafter
              {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
          }%
```

```
          {\GLSaccessfirstplural{\glslabel}\GLSfmtinsert}%
        }%
      }%
    }%
    {%
```

Singular form

```
    \glscapscase
    {%
```

Don't adjust case

```
      \ifglsused\glslabel
      {%
```

Subsequent use

```
        \glsifapplyinnerfmtfield{\glslabel}{text}%
        {%
          \expandafter\glsaccessfmttext\expandafter
            {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
        }%
        {\glsaccesstext{\glslabel}\glsfmtinsert}%
      }%
      {%
```

First use

```
        \glsifapplyinnerfmtfield{\glslabel}{first}%
        {%
          \expandafter\glsaccessfmtfirst\expandafter
           {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
        }%
        {\glsaccessfirst{\glslabel}\glsfmtinsert}%
      }%
    }%
    {%
```

Make first letter upper case

```
      \ifglsused\glslabel
      {%
```

Subsequent use

```
        \glsifapplyinnerfmtfield{\glslabel}{text}%
        {%
          \expandafter\Glsaccessfmttext\expandafter
            {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
        }%
        {\Glsaccesstext{\glslabel}\glsfmtinsert}%
      }%
      {%
```

First use

```
        \glsifapplyinnerfmtfield{\glslabel}{first}%
        {%
```

```
        \expandafter\Glsaccessfmtfirst\expandafter
          {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
      }%
      {\Glsaccessfirst{\glslabel}\glsfmtinsert}%
    }%
  }%
  {%
```
Make all upper case
```
      \ifglsused\glslabel
      {%
```
Subsequent use
```
      \glsifapplyinnerfmtfield{\glslabel}{text}%
      {%
        \expandafter\GLSaccessfmttext\expandafter
          {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
      }%
      {\GLSaccesstext{\glslabel}\GLSfmtinsert}%
    }%
    {%
```
First use
```
      \glsifapplyinnerfmtfield{\glslabel}{first}%
      {%
        \expandafter\GLSaccessfmtfirst\expandafter
          {\glsinsert}{\glsxtrgenentrytextfmt}{\glslabel}%
      }%
      {\GLSaccessfirst{\glslabel}\GLSfmtinsert}%
    }%
  }%
 }%
}%
{%
```
Custom text provided in `\glsdisp`, in which case the formatting should already
be applied.
```
    \glscustomtext
  }%
 }
```

Commands like `\glsifplural` are only used by the `\gls`-like commands in
the glossaries package, but it might be useful for the post-link 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`.

`\glsxtrifwasglslike` For use in the post-link hook, this indicates whether or not the hook follows a
`\gls`-like command.

```
\newcommand*{\glsxtrifwasglslike}[2]{#2}
```

```
\newcommand*{\glsxtrifwasglslikeandfirstuse}[2]{%
  \glsxtrifwasglslike
  {%
    \glsxtrifwasfirstuse{#1}{#2}%
  }{#2}%
}
```

\glsxtrifwassubsequentuse

```
\newcommand*{\glsxtrifwassubsequentuse}[2]{%
  \glsxtrifwasglslike
  {%
    \glsxtrifwasfirstuse{#2}{#1}%
  }{#2}%
}
```

\glsxtrifallcaps Shortcut.

```
\newcommand*{\glsxtrifallcaps}[2]{\glscapscase{#2}{#1}{#1}}
```

\glsxtrcurrentfield Another placeholder to find out information about the calling command. This will be empty for the \gls and \glsxtrfull set of commands and will be the singular field otherwise.

```
\newcommand*{\glsxtrcurrentfield}{}
```

\glsxtr@shortfieldname

```
\newcommand*{\glsxtr@shortfieldname}{short}
```

glsxtrifwassubsequentorshort

```
\newcommand*{\glsxtrifwassubsequentorshort}[2]{%
  \glsxtrifwasglslike
  {%
    \glsxtrifwasfirstuse{#2}{#1}%
  }%
  {\ifdefequal\glsxtrcurrentfield\glsxtr@shortfieldname{#1}{#2}}%
}
```

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

```
\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 (unless indexing has been switched off).

```
\@glsxtr@record{#2}{#3}{glslink}%
\glsdoifexists{#3}%
{%
```

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).

```
\let\glsxtrorg@ifKV@glslink@hyper\ifKV@glslink@hyper
```

Save local setting.

```
\@gls@save@glslocal
```

Initialise preunset, prereset and postunset

```
      \glsinitreunsets
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \def\glscustomtext{#4}%
      \@glsxtr@field@linkdefs
      #1%
      \@gls@link[#2]{#3}{#4}%
      \let\ifKV@glslink@hyper\glsxtrorg@ifKV@glslink@hyper
      \@gls@restore@glslocal
    }%
    \glspostlinkhook
  }
```

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

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

```
      \let\@glsxtr@org@gls@\@gls@
      \def\@gls@#1#2{%
        \def\glsxtrcurrentfield{}%
        \@glsxtr@record{#1}{#2}{glslink}%
        \@glsxtr@org@gls@{#1}{#2}%
      }%
```

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

```
      \let\@glsxtr@org@glspl@\@glspl@
      \def\@glspl@#1#2{%
        \def\glsxtrcurrentfield{}%
        \@glsxtr@record{#1}{#2}{glslink}%
        \@glsxtr@org@glspl@{#1}{#2}%
      }%
```

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

```
      \let\@glsxtr@org@Gls@\@Gls@
      \def\@Gls@#1#2{%
        \def\glsxtrcurrentfield{}%
        \@glsxtr@record{#1}{#2}{glslink}%
        \@glsxtr@org@Gls@{#1}{#2}%
      }%
```

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

```
      \let\@glsxtr@org@Glspl@\@Glspl@
      \def\@Glspl@#1#2{%
```

```
                \def\glsxtrcurrentfield{}%
                \@glsxtr@record{#1}{#2}{glslink}%
                \@glsxtr@org@Glspl@{#1}{#2}%
              }%
```

\@GLS@ Save the original definition and redefine.

```
              \let\@glsxtr@org@GLS@\@GLS@
              \def\@GLS@#1#2{%
                \def\glsxtrcurrentfield{}%
                \@glsxtr@record{#1}{#2}{glslink}%
                \@glsxtr@org@GLS@{#1}{#2}%
              }%
```

\@GLSpl@ Save the original definition and redefine.

```
              \let\@glsxtr@org@GLSpl@\@GLSpl@
              \def\@GLSpl@#1#2{%
                \def\glsxtrcurrentfield{}%
                \@glsxtr@record{#1}{#2}{glslink}%
                \@glsxtr@org@GLSpl@{#1}{#2}%
              }%
```

\@glsdisp This is redefined to allow the recording on the first run. Can't save and restore
          \@glsdisp since it has an optional argument.

```
              \renewcommand*{\@glsdisp}[3][]{%
                \def\glsxtrcurrentfield{}%
                \@glsxtr@record{#1}{#2}{glslink}%
                \glsdoifexists{#2}{%
                  \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
                  \let\glsifplural\@secondoftwo
                  \let\glscapscase\@firstofthree
                  \def\glscustomtext{\glsxtrgenentrytextfmt{#3}}%
                  \def\glsinsert{}%
                  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
                  \@gls@link[#1]{#2}{\@glo@text}%
                  \@gls@do@glsunset{#2}%
                }%
                \glspostlinkhook
              }
```

\@gls@@link Redefine to include \@glsxtr@record

```
              \renewcommand*{\@gls@@link}[3][]{%
                \def\glsxtrcurrentfield{}%
                \@glsxtr@record{#1}{#2}{glslink}%
                \glsdoifexistsordo{#2}%
                {%
                  \let\do@gls@link@checkfirsthyper\relax
```

Post-link hook commands need initialising.

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

```
                        \@glsxtr@field@linkdefs
                        \@gls@link[#1]{#2}{\glsxtrgenentrytextfmt{#3}}%
                     }%
                     {%
                        \glstextformat{#3}%
                     }%
                     \glspostlinkhook
                   }
```

\glsxtrinitwrgloss Set the default if the wrgloss is omitted.

```
                   \newcommand*{\glsxtrinitwrgloss}{%
                    \glsifattribute{\glslabel}{wrgloss}{after}%
                    {%
                      \glsxtrinitwrglossbeforefalse
                    }%
                    {%
                      \glsxtrinitwrglossbeforetrue
                    }%
                   }
```

\ifglsxtrwrglossbefore Conditional to determine if the indexing should be done before the link text.

```
                   \newif\ifglsxtrinitwrglossbefore
                   \glsxtrinitwrglossbeforetrue
```

\setupglslink Shortcut command to set glslink options.

```
                   \newcommand*{\setupglslink}[1]{\setkeys{glslink}{#1}}
```

\setupglsadd Shortcut command to set glsadd options.

```
                   \newcommand*{\setupglsadd}[1]{\setkeys{glsadd}{#1}}
```

\@gls@do@glsprereset

```
                   \newcommand*{\@gls@do@glsprereset}[1]{}

                   \define@choicekey{glslink}{prereset}%
                   [\@glsxtr@preresetval\@glsxtr@preresetnr]%
                   {none,local,global}[local]%
                   {%
                     \ifcase\@glsxtr@preresetnr
                      \let\@gls@do@glsprereset\@gobble
                     \or
                      \def\@gls@do@glsprereset{%
                        \let\@gls@link@postkeys@checkfirsthyper\do@gls@link@checkfirsthyper
                        \let\glsxtrifwasfirstuse\@firstoftwo\glslocalreset}%
                     \or
                      \def\@gls@do@glsprereset{%
                        \let\@gls@link@postkeys@checkfirsthyper\do@gls@link@checkfirsthyper
                        \let\glsxtrifwasfirstuse\@firstoftwo\glsreset}%
                     \fi
                   }
```

```
\newcommand*{\@gls@do@glspreunset}[1]{}
```

This hook was only introduced to glossaries v4.50, so if isn't defined, need to patch the hyper key.

```
\ifdef\@gls@glslink@hyper@update@hook
{%
  \renewcommand*{\@gls@glslink@hyper@update@hook}{%
    \let\@gls@if@glslink@hyper@updated\@firstoftwo
  }
}
{
  \newcommand*{\@gls@glslink@hyper@update@hook}{%
    \let\@gls@if@glslink@hyper@updated\@firstoftwo
  }
  \renewcommand*{\KV@glslink@hyper}[1]{%
    \XKV@plfalse\XKV@sttrue
    \XKV@checkchoice[\XKV@resa ]{#1}{true,false}%
     {\csname KV@glslink@hyper\XKV@resa\endcsname\@gls@glslink@hyper@update@hook}%
  }
}

\define@choicekey{glslink}{preunset}%
[\@glsxtr@preunsetval\@glsxtr@preunsetnr]%
{none,local,global}[local]%
{%
  \ifcase\@glsxtr@preunsetnr
   \let\@gls@do@glspreunset\@gobble
  \or
   \def\@gls@do@glspreunset{%
     \let\@gls@link@postkeys@checkfirsthyper\do@gls@link@checkfirsthyper
     \let\glsxtrifwasfirstuse\@secondoftwo\glslocalunset}%
  \or
   \def\@gls@do@glspreunset{%
     \let\@gls@link@postkeys@checkfirsthyper\do@gls@link@checkfirsthyper
     \let\glsxtrifwasfirstuse\@secondoftwo\glsunset}%
  \fi
}

\define@choicekey{glslink}{postunset}%
[\@glsxtr@postunsetval\@glsxtr@postunsetnr]%
{none,local,global}[global]%
{%
  \ifcase\@glsxtr@postunsetnr
    \let\@gls@restore@glslocal\@gls@ignore@restore@glslocal
  \or
    \let\@gls@restore@glslocal\@gls@default@restore@glslocal
    \KV@glslink@localtrue
  \or
    \let\@gls@restore@glslocal\@gls@default@restore@glslocal
```

```
                              \KV@glslink@localfalse
                           \fi
                         }

       \glsinitreunsets

                         \newcommand*{\glsinitreunsets}{%
                           \let\@gls@do@glspreunset\@gobble
                           \let\@gls@do@glsprereset\@gobble
                           \let\@gls@restore@glslocal\@gls@default@restore@glslocal
                           \@glsxtrbuffer@check@repeats
                         }
```

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

```
                         \define@choicekey{glslink}{wrgloss}%
                         [\@glsxtr@wrglossval\@glsxtr@wrglossnr]%
                         {before,after}%
                         {%
                           \ifcase\@glsxtr@wrglossnr\relax
                             \glsxtrinitwrglossbeforetrue
                           \or
                             \glsxtrinitwrglossbeforefalse
                           \fi
                         }

                         \define@key{glslink}{thevalue}{\def\@glsxtr@thevalue{#1}}

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

\ifglsxtr@hyperoutside Define a hyperoutside key to determine whether \hyperlink should be outside \glstextformat.

```
                         \define@boolkey{glslink}[glsxtr@]{hyperoutside}[true]{}
                         \glsxtr@hyperoutsidetrue
```

r@current@textformat@csname

```
                         \newcommand*{\@glsxtr@current@textformat@csname}{glstextformat}
```

rent@innertextformat@csname

```
                         \newcommand*{\@glsxtr@current@innertextformat@csname}{glsxtrdefaultentrytextfmt}
```

\glsxtrassignlinktextfmt Used to assign \glstextformat and \glsxtrgenentrytextfmt in the post-link hook for "postfootnote" abbreviation styles.

```
                         \newcommand*{\glsxtrassignlinktextfmt}{}
```

\@glsxtr@local@textformat Provide a key to locally change the text format.

```
                         \define@key{glslink}{textformat}{%
                           \ifcsdef{#1}
                           {%
                             \letcs{\@glsxtr@local@textformat}{#1}%
```

103

```
                          \def\@glsxtr@current@textformat@csname{#1}%
                        }%
                        {%
                          \PackageError{glossaries-extra}{Unknown control sequence name '#1'}{}%
                        }%
                      }
```

glsxtr@local@innertextformat Provide a key to locally change the inner text format.

```
                      \define@key{glslink}{innertextformat}{%
                        \ifcsdef{#1}
                        {%
                          \letcs{\@glsxtr@local@innertextformat}{#1}%
                          \def\@glsxtr@current@innertextformat@csname{#1}%
                        }%
                        {%
                          \PackageError{glossaries-extra}{Unknown control sequence name '#1'}{}%
                        }%
                      }

                      \define@key{glslink}{prefix}{\def\glolinkprefix{#1}}
```

\glsxtrinithyperoutside Set the default if the hyperoutside is omitted.

```
                      \newcommand*{\glsxtrinithyperoutside}{%
                       \glsifattribute{\glslabel}{hyperoutside}{false}%
                       {%
                         \glsxtr@hyperoutsidefalse
                       }%
                       {%
                         \glsxtr@hyperoutsidetrue
                       }%
                      }
```

\glsxtr@inc@linkcount Does nothing by default.

```
                      \newcommand*{\glsxtr@inc@linkcount}{}
```

\glslinkpresetkeys User hook performed immediately before options are set. Does nothing by
                    default.

```
                      \newcommand*{\glslinkpresetkeys}{}
```

\GlsXtrExpandedFmt Helper command that (protected) fully expands second argument and then ap-
                    plies it to the first, which must be a command that takes a single argument.

```
                      \newrobustcmd*{\GlsXtrExpandedFmt}[2]{%
                        \protected@edef\@glsxtr@tmp{#2}%
                        \expandafter#1\expandafter{\@glsxtr@tmp}%
                      }
```

sxtr@use@equation@counter@or If in a numbered equation, change the counter to equation. This can be over-
                              ridden by explicitly setting the counter in the optional argument of commands
                              like \gls and \glslink.

```
\newcommand*{\@glsxtr@use@equation@counter}{%
  \@glsxtr@ifnum@mmode{\def\@gls@counter{equation}}{}%
}
```

\glsxtr@do@autoadd If \GlsXtrAutoAddOnFormat is used, this will automatically use \glsadd. It's
therefore only used with \@gls@link not with \glsadd otherwise it could trigger
an infinite loop. The argument indicates the key family (glslink or glossadd).

```
\newcommand*{\glsxtr@do@autoadd}[1]{}
```

\GlsXtrAutoAddOnFormat[⟨*label*⟩]{⟨*format list*⟩}{⟨*glsadd
options*⟩}

\GlsXtrAutoAddOnFormat

If an entry is indexed with the format set to one identified in the comma-
separated list, then automatically index it using \glsadd with the given options,
which may override the current options. Scoping is needed to prevent leakage.

```
\newcommand*{\GlsXtrAutoAddOnFormat}[3][\glslabel]{%
  \renewcommand*{\glsxtr@do@autoadd}[1]{%
    \begingroup
      \protected@edef\@glsxtr@do@autoadd{%
        \noexpand\ifstrequal{##1}{glslink}%
        {%
          \noexpand\DTLifinlist
            {\@glsnumberformat}{#2}%
            {\noexpand\glsadd[format={\@glsnumberformat},#3]{#1}}{}%
        }%
        {}%
      }%
      \@glsxtr@do@autoadd
    \endgroup
  }%
}
```

\GlsXtrClearAutoAddOnFormat Reset.

```
\newcommand\GlsXtrClearAutoAddOnFormat{%
  \renewcommand*{\glsxtr@do@autoadd}[1]{}%
}
```

\glslinkwrcontent This was defined to add grouping to resolve issue #189 but had unexpected
consequences (issue #194) so the grouping has been removed and transferred
to \glsencapwrcontent.

```
\providecommand*{\glslinkwrcontent}[1]{#1}
```

\@glslink@prefix@label Hyperlink using current prefix and label.

```
\newcommand*{\@glslink@prefix@label}[1]{%
  \@glslink{\glolinkprefix\glslabel}{#1}}
```

\@noglslink@prefix@label

```
\newcommand*{\@noglslink@prefix@label}[1]{%
  \glsdonohyperlink{\glolinkprefix\glslabel}{#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.

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

  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \let\@glsnumberformat\@glsxtr@defaultnumberformat
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
```

Save local setting.

```
  \@gls@save@glslocal
```

Initialise preunset, prereset and postunset

```
  \glsinitreunsets
```

Save current value of `\glolinkprefix`:

```
  \let\@glsxtr@org@glolinkprefix\glolinkprefix
```

Initialise `\@glsxtr@local@textformat`

```
  \let\@glsxtr@local@textformat\relax
  \def\@glsxtr@current@textformat@csname{glstextformat}%
```

Initialise inner text format (1.49):

```
  \let\@glsxtr@local@innertextformat\glsxtrdefaultentrytextfmt
  \def\@glsxtr@current@innertextformat@csname{glsxtrdefaultentrytextfmt}%
```

Initialise thevalue and theHvalue (v1.19).

```
  \def\@glsxtr@thevalue{}%
  \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
```

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

```
  \glsxtrinitwrgloss
```

Initialise whether `\hyperlink` should be outside `\glstextformat` (new to v1.21).

```
  \glsxtrinithyperoutside
```

Note that the default link options may override `\glsxtrinitwrgloss`.

```
  \@gls@setdefault@glslink@opts
```

Increment link counter if enabled (new to v1.26).

```
  \glsxtr@inc@linkcount
```

Check if the equations option has been set (new to v1.37).

```
\if@glsxtr@equations
  \@glsxtr@use@equation@counter
\fi
```

As the original definition.

```
\do@glsdisablehyperinlist
\do@gls@link@checkfirsthyper
```

Provide way of finding if hyper key has been explicitly set.

```
\let\@gls@if@glslink@hyper@updated\@secondoftwo
\let\@gls@link@postkeys@checkfirsthyper\relax
```

User hook before options are set (new to v1.26):

```
\glslinkpresetkeys
```

Set options.

```
\setkeys{glslink}{#1}%
```

Perform auto add if set (new to v1.37)

```
\glsxtr@do@autoadd{glslink}%
```

User hook after options are set:

```
\glslinkpostsetkeys
```

Reset/unset if required:

```
\@gls@do@glsprereset{#2}%
\@gls@do@glspreunset{#2}%
```

If the hyper setting hasn't changed, and reset/unset option has been used, need to perform another check.

```
\@gls@if@glslink@hyper@updated{}{\@gls@link@postkeys@checkfirsthyper}%
```

Set inner text format (1.49):

```
\let\glsxtrgenentrytextfmt\@glsxtr@local@innertextformat
```

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

```
\ifdefempty{\@glsxtr@thevalue}%
{%
  \@gls@saveentrycounter
}%
{%
  \let\theglsentrycounter\@glsxtr@thevalue
  \def\theHglsentrycounter{\@glsxtr@theHvalue}%
}%
\@gls@setsort{\glslabel}%
```

Check if the textformat key has been used.

```
\ifx\@glsxtr@local@textformat\relax
```

Check textformat attribute (new to v1.21).

```
\glshasattribute{\glslabel}{textformat}%
{%
  \protected@edef\@glsxtr@attrval{\glsgetattribute{\glslabel}{textformat}}%
```

107

```
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@textformat}{\@glsxtr@attrval}%
          \let\@glsxtr@current@textformat@csname\@glsxtr@attrval
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          '\@glsxtr@attrval' supplied in textformat attribute
          for entry '\glslabel'. Reverting to default \string\glstextformat}%
          \let\@glsxtr@textformat\glstextformat
        }%
      }%
      {%
        \let\@glsxtr@textformat\glstextformat
      }%
    \else
      \let\@glsxtr@textformat\@glsxtr@local@textformat
    \fi
```

Setup formatting assignments for use in post-link hook.

```
    \edef\glsxtrassignlinktextfmt{%
      \noexpand\def\noexpand\glslabel{\expandonce\glslabel}%
      \noexpand\letcs\noexpand\glstextformat{\@glsxtr@current@textformat@csname}%
      \noexpand\letcs\noexpand\glsxtrgenentrytextfmt
        {\@glsxtr@current@innertextformat@csname}%
    }%
```

Encapsulate link text and indexing.

```
    \glslinkwrcontent
    {%
```

Do write if it should occur before the link text:

```
      \ifglsxtrinitwrglossbefore
        \glsxtr@wrglossary@encap{#2}{\@do@wrglossary{#2}}%
      \fi
```

Do the link text:

```
      \ifKV@glslink@hyper
        \ifglsxtr@hyperoutside
         \@glslink@prefix@label{\@glsxtr@textformat{#3}}%
        \else
         \@glsxtr@textformat{\@glslink@prefix@label{#3}}%
        \fi
      \else
        \ifglsxtr@hyperoutside
         \@noglslink@prefix@label{\@glsxtr@textformat{#3}}%
        \else
         \@glsxtr@textformat{\@noglslink@prefix@label{#3}}%
        \fi
      \fi
```

Do write if it should occur after the link text:

```
    \ifglsxtrinitwrglossbefore
    \else
      \glsxtr@wrglossary@encap{#2}{\@do@wrglossary{#2}}%
    \fi
  }%
```

Restore original value of `\glolinkprefix`:

```
    \let\glolinkprefix\@glsxtr@org@glolinkprefix
```

As the original definition:

```
    \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
    \@gls@restore@glslocal
  }
```

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

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

`\glsaddpresetkeys`

```
    \newcommand*{\glsaddpresetkeys}{}
```

`\glsaddpostsetkeys`

```
    \newcommand*{\glsaddpostsetkeys}{}
```

`\glsadd` Redefine to include `\@glsxtr@record` and suppress in headings

```
    \renewrobustcmd*{\glsadd}[2][]{%
      \glsxtrifinmark
      {}%
      {%
        \@gls@adjustmode
        \begingroup
        \@glsadd{#1}{#2}%
        \endgroup
      }%
    }
```

`\@glsadd`

```
    \newcommand{\@glsadd}[2]{%
      \@glsxtr@record{#1}{#2}{glossadd}%
      \glsdoifexists{#2}%
      {%
        \let\@glsnumberformat\@glsxtr@defaultnumberformat

        \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
        \def\@glsxtr@thevalue{}%
        \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
```

Implement any default settings (before options are set)

```
        \glsaddpresetkeys
        \setkeys{glossadd}{#1}%
```

109

Implement any default settings (after options are set)

```
\glsaddpostsetkeys
\ifdefempty{\@glsxtr@thevalue}%
{%
  \@gls@saveentrycounter
}%
{%
  \let\theglsentrycounter\@glsxtr@thevalue
  \def\theHglsentrycounter{\@glsxtr@theHvalue}%
}%
```

Define sort key if necessary (in case of sort=use):

```
\@gls@setsort{#2}%
```

Ensure that indexing occurs (since that's the point of `\glsadd`). If indexing has been switched off by default, don't want the setting to affect `\glsadd`. The ignored format `\glsignore` can be used for selection without location, but the indexing still needs to be performed.

```
  \KV@glslink@noindexfalse
  \glsxtr@wrglossary@encap{#2}{\@@do@wrglossary{#2}}%
 }%
}
```

`\glsaddeach` Performs `\glsadd` for each entry listed in the mandatory argument.

```
\newrobustcmd{\glsaddeach}[2][]{%
  \glsxtrifinmark
  {}%
  {%
    \@gls@adjustmode
    \@for\@gls@thislabel:=#2\do{{\@glsadd{#1}{\@gls@thislabel}}}%
  }%
}
```

`\glsxtr@rangeformat`

```
\newcommand{\glsxtr@rangeformat}{\@glsxtr@defaultnumberformat}
```

`\GlsXtrSetDefaultRangeFormat`

```
\newcommand*{\GlsXtrSetDefaultRangeFormat}[1]{%
  \renewcommand*{\glsxtr@rangeformat}{#1}%
}%
```

`\glsstartrange` Essentially does `\glsadd[format=(]{`$\langle label\rangle$`}`

```
\newrobustcmd{\glsstartrange}[2][]{%
  \glsxtrifinmark
  {}%
  {%
    \@gls@adjustmode
    \begingroup
    \appto\glsaddpresetkeys{\protected@edef\@glsnumberformat{\glsxtr@rangeformat}}%
```

```
            \appto\glsaddpostsetkeys{\protected@edef\@glsnumberformat{(\@glsnumberformat}}%
            \@for\@gls@thislabel:=#2\do{\@glsadd{#1}{\@gls@thislabel}}%
            \endgroup
          }%
        }
```

\glsendrange   Essentially does \glsadd[format=)]{⟨label⟩}

```
          \newrobustcmd{\glsendrange}[2][]{%
            \glsxtrifinmark
            {}%
            {%
              \@gls@adjustmode
              \begingroup
              \appto\glsaddpresetkeys{\protected@edef\@glsnumberformat{\glsxtr@rangeformat}}%
              \appto\glsaddpostsetkeys{\protected@edef\@glsnumberformat{)\@glsnumberformat}}%
              \@for\@gls@thislabel:=#2\do{\@glsadd{#1}{\@gls@thislabel}}%
              \endgroup
            }%
          }
```

\@glsxtr@field@linkdefs   Default settings for \@gls@field@link. Note that from v1.49, \glsinsert is
                           set with \glsxtrsaveinsert.

```
          \newcommand*{\@glsxtr@field@linkdefs}{%
            \let\glsxtrifwasglslike\@secondoftwo
            \let\glsxtrifwasfirstuse\@secondoftwo
            \let\glsifplural\@secondoftwo
            \let\glscapscase\@firstofthree
          }
```

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

\glsxtrassignfieldfont

```
          \newcommand*{\glsxtrassignfieldfont}[1]{%
            \ifglsentryexists{#1}%
            {%
              \ifglshasshort{#1}%
              {%
                \glssetabbrvfmt{\glscategory{#1}}%
                \glsifregular{#1}%
                {\let\@gls@field@font\glsxtrregularfont}%
                {\let\@gls@field@font\@firstofone}%
              }%
              {%
                \glsifnotregular{#1}%
                {\let\@gls@field@font\@firstofone}%
                {\let\@gls@field@font\glsxtrregularfont}%
              }%
            }%
            {%
```

```
      \let\@gls@field@font\@gobble
    }%
  }
```

<div style="border:1px solid; background:#ffffcc; border-radius:8px; padding:8px">

$\glsxtrsaveinsert\{\langle entry\text{-}label\rangle\}\{\langle insert\rangle\}$

</div>

\glsxtrsaveinsert

The insert argument isn't saved in `\glsinsert` for the `\glslike` commands, but provide a way to save it if it is required for the post-link hook. The default is to set `\glsinsert` to empty. This means that the insert won't appear in the post-link hook with commands like `\glsxtrfull` for the hyphen abbreviation styles. The entry label is provided in case the insert should only be saved for certain entries, such as those with a particular category.

```
\newcommand*{\glsxtrsaveinsert}[2]{\def\glsinsert{}}
```

\glsxtrfullsaveinsert As above but specifically for commands like `\glsxtrfull`

```
\newcommand*{\glsxtrfullsaveinsert}{\glsxtrsaveinsert}
```

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

```
\def\@glstext@#1#2[#3]{%
  \def\glsxtrcurrentfield{text}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link{#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{text}%
    {%
      \@gls@field@font{\glsaccessfmttext{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\glsaccesstext{#2}\glsxtrgenentrytextfmt{#3}}%
    }%
  }%
}
```

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

```
\def\@GLStext@#1#2[#3]{%
  \def\glsxtrcurrentfield{text}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]{#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{text}%
    {%
      \@gls@field@font{\GLSaccessfmttext{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
```

```
            \ifx\glscapscase\@thirdofthree
              \@gls@field@font{\GLSaccesstext{#2}%
                \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
            \else
              \@gls@field@font{\glsaccesstext{#2}\glsxtrgenentrytextfmt{#3}}%
            \fi
          }%
        }%
      }
```

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

```
\def\@Glstext@#1#2[#3]{%
  \def\glsxtrcurrentfield{text}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@secondofthree]{#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{text}%
    {%
      \@gls@field@font{\Glsaccessfmttext{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccesstext{#2}\glsxtrgenentrytextfmt{#3}}%
    }%
  }%
}
```

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.

\glsxtrchecknohyperfirst

```
\newcommand*{\glsxtrchecknohyperfirst}[1]{%
  \glsifattribute{#1}{nohyperfirst}{true}{\KV@glslink@hyperfalse}{}%
}
```

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

```
\def\@glsfirst@#1#2[#3]{%
  \def\glsxtrcurrentfield{first}%
  \glsxtrassignfieldfont{#2}%
```

Ensure that `\glsfirst` honours the nohyperfirst attribute.

```
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glsxtrifwasfirstuse\@firstoftwo
   \glsxtrchecknohyperfirst{#2}%
   \glsxtr@check@complexstyle{#2}{#3}%
  ]{#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{first}%
```

```
            {%
              \@gls@field@font{\glsaccessfmtfirst{#3}{\glsxtrgenentrytextfmt}{#2}}%
            }%
            {%
              \@gls@field@font{\glsaccessfirst{#2}\glsxtrgenentrytextfmt{#3}}%
            }%
         }%
       }
```

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

```
       \def\@Glsfirst@#1#2[#3]{%
         \def\glsxtrcurrentfield{first}%
         \glsxtrassignfieldfont{#2}%
```

Ensure that `\Glsfirst` honours the nohyperfirst attribute.

```
         \glsxtrsaveinsert{#2}{#3}%
         \@gls@field@link
         [\let\glsxtrifwasfirstuse\@firstoftwo
          \let\glscapscase\@secondofthree
          \glsxtrchecknohyperfirst{#2}%
          \glsxtr@check@complexstyle{#2}{#3}%
         ]%
         {#1}{#2}%
         {%
           \glsifapplyinnerfmtfield{#2}{first}%
           {%
             \@gls@field@font{\Glsaccessfmtfirst{#3}{\glsxtrgenentrytextfmt}{#2}}%
           }%
           {%
             \@gls@field@font{\Glsaccessfirst{#2}\glsxtrgenentrytextfmt{#3}}%
           }%
         }%
       }
```

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

```
       \def\@GLSfirst@#1#2[#3]{%
         \def\glsxtrcurrentfield{first}%
         \glsxtrassignfieldfont{#2}%
```

Ensure that `\GLSfirst` honours the nohyperfirst attribute.

```
         \glsxtrsaveinsert{#2}{#3}%
         \@gls@field@link
         [\let\glsxtrifwasfirstuse\@firstoftwo
          \let\glscapscase\@thirdofthree
          \glsxtrchecknohyperfirst{#2}%
          \glsxtr@check@complexstyle{#2}{#3}%
         ]%
         {#1}{#2}%
         {%
           \ifx\glscapscase\@thirdofthree
```

114

```
               \glsifapplyinnerfmtfield{#2}{first}%
               {%
                 \@gls@field@font{\GLSaccessfmtfirst{#3}{\glsxtrgenentrytextfmt}{#2}}%
               }%
               {%
                 \@gls@field@font{\GLSaccessfirst{#2}%
                   \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
               }%
              \else
              \glsifapplyinnerfmtfield{#2}{first}%
              {%
                \@gls@field@font{\glsaccessfmtfirst{#3}{\glsxtrgenentrytextfmt}{#2}}%
              }%
              {%
                \@gls@field@font{\glsaccessfirst{#2}\glsxtrgenentrytextfmt{#3}}%
              }%
             \fi
           }%
         }
```

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

```
         \def\@glsplural@#1#2[#3]{%
           \def\glsxtrcurrentfield{text}%
           \glsxtrassignfieldfont{#2}%
           \glsxtrsaveinsert{#2}{#3}%
           \@gls@field@link
            [\let\glsifplural\@firstoftwo
             \glsxtr@check@complexstyle{#2}{#3}%
            ]{#1}{#2}%
           {%
             \glsifapplyinnerfmtfield{#2}{plural}%
             {%
               \@gls@field@font{\glsaccessfmtplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
             }%
             {%
               \@gls@field@font{\glsaccessplural{#2}\glsxtrgenentrytextfmt{#3}}%
             }%
           }%
         }
```

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

```
         \def\@Glsplural@#1#2[#3]{%
           \def\glsxtrcurrentfield{text}%
           \glsxtrassignfieldfont{#2}%
           \glsxtrsaveinsert{#2}{#3}%
           \@gls@field@link
           [\let\glsifplural\@firstoftwo
            \let\glscapscase\@secondofthree
             \glsxtr@check@complexstyle{#2}{#3}%
           ]%
```

```
            {#1}{#2}%
            {%
              \glsifapplyinnerfmtfield{#2}{plural}%
              {%
                \@gls@field@font{\Glsaccessfmtplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
              }%
              {%
                \@gls@field@font{\Glsaccessplural{#2}\glsxtrgenentrytextfmt{#3}}%
              }%
            }%
          }
```

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

```
          \def\@GLSplural@#1#2[#3]{%
            \def\glsxtrcurrentfield{text}%
            \glsxtrassignfieldfont{#2}%
            \glsxtrsaveinsert{#2}{#3}%
            \@gls@field@link
            [\let\glsifplural\@firstoftwo
             \let\glscapscase\@thirdofthree
             \glsxtr@check@complexstyle{#2}{#3}%
            ]%
            {#1}{#2}%
            {%
             \ifx\glscapscase\@thirdofthree
              \glsifapplyinnerfmtfield{#2}{plural}%
              {%
                \@gls@field@font{\GLSaccessfmtplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
              }%
              {%
                \@gls@field@font{\GLSaccessplural{#2}%
                  \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
              }%
             \else
              \glsifapplyinnerfmtfield{#2}{plural}%
              {%
                \@gls@field@font{\glsaccessfmtplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
              }%
              {%
                \@gls@field@font{\glsaccessplural{#2}\glsxtrgenentrytextfmt{#3}}%
              }%
             \fi
            }%
          }
```

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

```
          \def\@glsfirstplural@#1#2[#3]{%
            \def\glsxtrcurrentfield{first}%
            \glsxtrassignfieldfont{#2}%
```

116

Ensure that `\glsfirstplural` honours the nohyperfirst attribute.

```
    \glsxtrsaveinsert{#2}{#3}%
    \@gls@field@link
    [\let\glsxtrifwasfirstuse\@firstoftwo
     \let\glsifplural\@firstoftwo
     \glsxtrchecknohyperfirst{#2}%
     \glsxtr@check@complexstyle{#2}{#3}%
    ]%
    {#1}{#2}%
    {%
      \glsifapplyinnerfmtfield{#2}{firstpl}%
      {%
        \@gls@field@font{\glsaccessfmtfirstplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
      }%
      {%
        \@gls@field@font{\glsaccessfirstplural{#2}\glsxtrgenentrytextfmt{#3}}%
      }%
    }%
  }
```

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

```
  \def\@Glsfirstplural@#1#2[#3]{%
    \def\glsxtrcurrentfield{first}%
    \glsxtrassignfieldfont{#2}%
```

Ensure that `\glsfirstplural` honours the nohyperfirst attribute.

```
    \glsxtrsaveinsert{#2}{#3}%
    \@gls@field@link
    [\let\glsxtrifwasfirstuse\@firstoftwo
     \let\glsifplural\@firstoftwo
     \let\glscapscase\@secondofthree
     \glsxtrchecknohyperfirst{#2}%
     \glsxtr@check@complexstyle{#2}{#3}%
    ]%
    {#1}{#2}%
    {%
      \glsifapplyinnerfmtfield{#2}{firstpl}%
      {%
        \@gls@field@font{\Glsaccessfmtfirstplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
      }%
      {%
        \@gls@field@font{\Glsaccessfirstplural{#2}\glsxtrgenentrytextfmt{#3}}%
      }%
    }%
  }
```

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

```
  \def\@GLSfirstplural@#1#2[#3]{%
    \def\glsxtrcurrentfield{first}%
    \glsxtrassignfieldfont{#2}%
```

Ensure that `\glsfirstplural` honours the nohyperfirst attribute.

```
    \glsxtrsaveinsert{#2}{#3}%
    \@gls@field@link
    [\let\glsxtrifwasfirstuse\@firstoftwo
     \let\glsifplural\@firstoftwo
     \let\glscapscase\@thirdofthree
     \glsxtrchecknohyperfirst{#2}%
     \glsxtr@check@complexstyle{#2}{#3}%
    ]%
    {#1}{#2}%
    {%
     \ifx\glscapscase\@thirdofthree
     \glsifapplyinnerfmtfield{#2}{firstpl}%
     {%
       \@gls@field@font{\GLSaccessfmtfirstplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
     }%
     {%
       \@gls@field@font{\GLSaccessfirstplural{#2}%
        \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
     }%
     \else
     \glsifapplyinnerfmtfield{#2}{firstpl}%
     {%
       \@gls@field@font{\glsaccessfmtfirstplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
     }%
     {%
       \@gls@field@font{\glsaccessfirstplural{#2}\glsxtrgenentrytextfmt{#3}}%
     }%
     \fi
    }%
  }
```

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

```
  \def\@glsname@#1#2[#3]{%
    \def\glsxtrcurrentfield{name}%
    \glsxtrassignfieldfont{#2}%
    \glsxtrsaveinsert{#2}{#3}%
    \@gls@field@link{#1}{#2}%
    {%
     \glsifapplyinnerfmtfield{#2}{name}%
     {%
       \@gls@field@font{\glsaccessfmtname{#3}{\glsxtrgenentrytextfmt}{#2}}%
     }%
     {%
       \@gls@field@font{\glsaccessname{#2}\glsxtrgenentrytextfmt{#3}}%
     }%
    }%
  }
```

118

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

```
\def\@Glsname@#1#2[#3]{%
  \def\glsxtrcurrentfield{name}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]{#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{name}%
    {%
      \@gls@field@font{\Glsaccessfmtname{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccessname{#2}\glsxtrgenentrytextfmt{#3}}%
    }%
  }%
}
```

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

```
\def\@GLSname@#1#2[#3]{%
  \def\glsxtrcurrentfield{name}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{name}%
    {%
      \@gls@field@font{\GLSaccessfmtname{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\GLSaccessname{#2}%
       \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
    }%
  }%
}
```

\@glsdesc@

```
\def\@glsdesc@#1#2[#3]{%
  \def\glsxtrcurrentfield{description}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link{#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{desc}%
    {%
      \@gls@field@font{\glsaccessfmtdesc{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
```

```
                  \@gls@field@font{\glsaccessdesc{#2}\glsxtrgenentrytextfmt{#3}}%
                }%
              }%
            }
```

\@Glsdesc@ First letter uppercase version.

```
            \def\@Glsdesc@#1#2[#3]{%
              \def\glsxtrcurrentfield{description}%
              \glsxtrassignfieldfont{#2}%
              \glsxtrsaveinsert{#2}{#3}%
              \@gls@field@link
              [\let\glscapscase\@secondofthree]{#1}{#2}%
              {%
                \glsifapplyinnerfmtfield{#2}{desc}%
                {%
                  \@gls@field@font{\Glsaccessfmtdesc{#3}{\glsxtrgenentrytextfmt}{#2}}%
                }%
                {%
                  \@gls@field@font{\Glsaccessdesc{#2}\glsxtrgenentrytextfmt{#3}}%
                }%
              }%
            }
```

\@GLSdesc@ All uppercase version.

```
            \def\@GLSdesc@#1#2[#3]{%
              \def\glsxtrcurrentfield{description}%
              \glsxtrassignfieldfont{#2}%
              \glsxtrsaveinsert{#2}{#3}%
              \@gls@field@link[\let\glscapscase\@thirdofthree]%
              {#1}{#2}%
              {%
                \glsifapplyinnerfmtfield{#2}{desc}%
                {%
                  \@gls@field@font{\GLSaccessfmtdesc{#3}{\glsxtrgenentrytextfmt}{#2}}%
                }%
                {%
                  \@gls@field@font{\GLSaccessdesc{#2}%
                   \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
                }%
              }%
            }
```

\@glsdescplural@ No case-changing version.

```
            \def\@glsdescplural@#1#2[#3]{%
              \def\glsxtrcurrentfield{description}%
              \glsxtrassignfieldfont{#2}%
              \glsxtrsaveinsert{#2}{#3}%
              \@gls@field@link
              [\let\glscapscase\@secondofthree
```

```
       \let\glsifplural\@firstoftwo
     ]{#1}{#2}%
     {%
       \glsifapplyinnerfmtfield{#2}{descplural}%
       {%
         \@gls@field@font{\glsaccessfmtdescpl{#3}{\glsxtrgenentrytextfmt}{#2}}%
       }%
       {%
         \@gls@field@font{\glsaccessdescplural{#2}\glsxtrgenentrytextfmt{#3}}%
       }%
     }%
   }
```

\@Glsdescplural@ First letter uppercase version.

```
   \def\@Glsdescplural@#1#2[#3]{%
     \def\glsxtrcurrentfield{description}%
     \glsxtrassignfieldfont{#2}%
     \glsxtrsaveinsert{#2}{#3}%
     \@gls@field@link
     [\let\glscapscase\@secondofthree
      \let\glsifplural\@firstoftwo
     ]{#1}{#2}%
     {%
       \glsifapplyinnerfmtfield{#2}{descplural}%
       {%
         \@gls@field@font{\Glsaccessfmtdescpl{#3}{\glsxtrgenentrytextfmt}{#2}}%
       }%
       {%
         \@gls@field@font{\Glsaccessdescplural{#2}#3}%
       }%
     }%
   }
```

\@GLSdescplural@ All uppercase version.

```
   \def\@GLSdescplural@#1#2[#3]{%
     \def\glsxtrcurrentfield{description}%
     \glsxtrassignfieldfont{#2}%
     \glsxtrsaveinsert{#2}{#3}%
     \@gls@field@link
     [\let\glscapscase\@thirdofthree
      \let\glsifplural\@firstoftwo
     ]%
     {#1}{#2}%
     {%
       \glsifapplyinnerfmtfield{#2}{descplural}%
       {%
         \@gls@field@font{\GLSaccessfmtdescplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
       }%
       {%
         \@gls@field@font{\GLSaccessdescplural{#2}%
```

```
              \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
          }%
        }%
      }
```

\@glssymbol@

```
        \def\@glssymbol@#1#2[#3]{%
          \def\glsxtrcurrentfield{symbol}%
          \glsxtrassignfieldfont{#2}%
          \glsxtrsaveinsert{#2}{#3}%
          \@gls@field@link{#1}{#2}%
          {%
            \glsifapplyinnerfmtfield{#2}{symbol}%
            {%
              \@gls@field@font{\glsaccessfmtsymbol{#3}{\glsxtrgenentrytextfmt}{#2}}%
            }%
            {%
              \@gls@field@font{\glsaccesssymbol{#2}\glsxtrgenentrytextfmt{#3}}%
            }%
          }%
        }
```

\@Glssymbol@ First letter uppercase version.

```
        \def\@Glssymbol@#1#2[#3]{%
          \def\glsxtrcurrentfield{symbol}%
          \glsxtrassignfieldfont{#2}%
          \glsxtrsaveinsert{#2}{#3}%
          \@gls@field@link
          [\let\glscapscase\@secondofthree]%
          {#1}{#2}%
          {%
            \glsifapplyinnerfmtfield{#2}{symbol}%
            {%
              \@gls@field@font{\Glsaccessfmtsymbol{#3}{\glsxtrgenentrytextfmt}{#2}}%
            }%
            {%
              \@gls@field@font{\Glsaccesssymbol{#2}\glsxtrgenentrytextfmt{#3}}%
            }%
          }%
        }
```

\@GLSsymbol@ All uppercase version.

```
        \def\@GLSsymbol@#1#2[#3]{%
          \def\glsxtrcurrentfield{symbol}%
          \glsxtrassignfieldfont{#2}%
          \glsxtrsaveinsert{#2}{#3}%
          \@gls@field@link[\let\glscapscase\@thirdofthree]%
          {#1}{#2}%
          {%
```

122

```
                     \glsifapplyinnerfmtfield{#2}{symbol}%
                     {%
                       \@gls@field@font{\GLSaccessfmtsymbol{#3}{\glsxtrgenentrytextfmt}{#2}}%
                     }%
                     {%
                       \@gls@field@font{\GLSaccesssymbol{#2}%
                        \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
                     }%
                   }%
                 }
```

\@glssymbolplural@ No case-changing version.

```
                   \def\@glssymbolplural@#1#2[#3]{%
                     \def\glsxtrcurrentfield{symbol}%
                     \glsxtrassignfieldfont{#2}%
                     \glsxtrsaveinsert{#2}{#3}%
                     \@gls@field@link
                     [\let\glscapscase\@secondofthree
                      \let\glsifplural\@firstoftwo
                     ]{#1}{#2}%
                     {%
                       \glsifapplyinnerfmtfield{#2}{symbolplural}%
                       {%
                         \@gls@field@font{\glsaccessfmtsymbolplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
                       }%
                       {%
                         \@gls@field@font{\glsaccesssymbolplural{#2}\glsxtrgenentrytextfmt{#3}}%
                       }%
                     }%
                   }
```

\@Glssymbolplural@ First letter uppercase version.

```
                   \def\@Glssymbolplural@#1#2[#3]{%
                     \def\glsxtrcurrentfield{symbol}%
                     \glsxtrassignfieldfont{#2}%
                     \glsxtrsaveinsert{#2}{#3}%
                     \@gls@field@link
                     [\let\glscapscase\@secondofthree
                      \let\glsifplural\@firstoftwo
                     ]{#1}{#2}%
                     {%
                       \glsifapplyinnerfmtfield{#2}{symbolplural}%
                       {%
                         \@gls@field@font{\Glsaccessfmtsymbolplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
                       }%
                       {%
                         \@gls@field@font{\Glsaccesssymbolplural{#2}\glsxtrgenentrytextfmt{#3}}%
                       }%
                     }%
                   }
```

\@GLSsymbolplural@ All uppercase version.

```
\def\@GLSsymbolplural@#1#2[#3]{%
  \def\glsxtrcurrentfield{symbol}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@thirdofthree
   \let\glsifplural\@firstoftwo
  ]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{symbolplural}%
    {%
      \@gls@field@font{\GLSaccessfmtsymbolplural{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\GLSaccesssymbolplural{#2}%
        \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
    }%
  }%
}
```

\@glsuseri@ User 1 field.

```
\def\@glsuseri@#1#2[#3]{%
  \def\glsxtrcurrentfield{user1}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{useri}%
    {%
      \@gls@field@font{\glsaccessfmtuseri{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\glsaccessuseri{#2}\glsxtrgenentrytextfmt{#3}}%
    }%
  }%
}
```

\@Glsuseri@ First letter uppercase version.

```
\def\@Glsuseri@#1#2[#3]{%
  \def\glsxtrcurrentfield{user1}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]{#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{useri}%
```

```
          {%
            \@gls@field@font{\Glsaccessfmtuseri{#3}{\glsxtrgenentrytextfmt}{#2}}%
          }%
          {%
            \@gls@field@font{\Glsaccessuseri{#2}\glsxtrgenentrytextfmt{#3}}%
          }%
        }%
      }
```

\@GLSuseri@ All uppercase version.

```
      \def\@GLSuseri@#1#2[#3]{%
        \def\glsxtrcurrentfield{user1}%
        \glsxtrassignfieldfont{#2}%
        \glsxtrsaveinsert{#2}{#3}%
        \@gls@field@link[\let\glscapscase\@thirdofthree]%
        {#1}{#2}%
        {%
          \glsifapplyinnerfmtfield{#2}{useri}%
          {%
            \@gls@field@font{\GLSaccessfmtuseri{#3}{\glsxtrgenentrytextfmt}{#2}}%
          }%
          {%
            \ifx\glscapscase\@thirdofthree
              \@gls@field@font{\GLSaccessuseri{#2}%
                \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
            \else
              \@gls@field@font{\glsaccessuseri{#2}\glsxtrgenentrytextfmt{#3}}%
            \fi
          }%
        }%
      }
```

\@glsuserii@ User 2 field.

```
      \def\@glsuserii@#1#2[#3]{%
        \def\glsxtrcurrentfield{user2}%
        \glsxtrassignfieldfont{#2}%
        \glsxtrsaveinsert{#2}{#3}%
        \@gls@field@link
        {#1}{#2}%
        {%
          \glsifapplyinnerfmtfield{#2}{userii}%
          {%
            \@gls@field@font{\glsaccessfmtuserii{#3}{\glsxtrgenentrytextfmt}{#2}}%
          }%
          {%
            \@gls@field@font{\glsaccessuserii{#2}\glsxtrgenentrytextfmt{#3}}%
          }%
        }%
      }
```

`\@Glsuserii@` First letter uppercase version.

```
\def\@Glsuserii@#1#2[#3]{%
  \def\glsxtrcurrentfield{user2}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{userii}%
    {%
      \@gls@field@font{\Glsaccessfmtuserii{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccessuserii{#2}\glsxtrgenentrytextfmt{#3}}%
    }%
  }%
}
```

`\@GLSuserii@` All uppercase version.

```
\def\@GLSuserii@#1#2[#3]{%
  \def\glsxtrcurrentfield{user2}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{userii}%
    {%
      \@gls@field@font{\GLSaccessfmtuserii{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \ifx\glscapscase\@thirdofthree
        \@gls@field@font{\GLSaccessuserii{#2}%
          \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
      \else
        \@gls@field@font{\glsaccessuserii{#2}\glsxtrgenentrytextfmt{#3}}%
      \fi
    }%
  }%
}
```

`\@glsuseriii@` User 3 field.

```
\def\@glsuseriii@#1#2[#3]{%
  \def\glsxtrcurrentfield{user3}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  {#1}{#2}%
```

126

```
                {%
                  \glsifapplyinnerfmtfield{#2}{useriii}%
                  {%
                    \@gls@field@font{\glsaccessfmtuseriii{#3}{\glsxtrgenentrytextfmt}{#2}}%
                  }%
                  {%
                    \@gls@field@font{\glsaccessuseriii{#2}\glsxtrgenentrytextfmt{#3}}%
                  }%
                }%
              }
```

\@Glsuseriii@ First letter uppercase version.

```
              \def\@Glsuseriii@#1#2[#3]{%
                \def\glsxtrcurrentfield{user3}%
                \glsxtrassignfieldfont{#2}%
                \glsxtrsaveinsert{#2}{#3}%
                \@gls@field@link
                [\let\glscapscase\@secondofthree]%
                {#1}{#2}%
                {%
                  \glsifapplyinnerfmtfield{#2}{useriii}%
                  {%
                    \@gls@field@font{\Glsaccessfmtuseriii{#3}{\glsxtrgenentrytextfmt}{#2}}%
                  }%
                  {%
                    \@gls@field@font{\Glsaccessuseriii{#2}\glsxtrgenentrytextfmt{#3}}%
                  }%
                }%
              }
```

\@GLSuseriii@ All uppercase version.

```
              \def\@GLSuseriii@#1#2[#3]{%
                \def\glsxtrcurrentfield{user3}%
                \glsxtrassignfieldfont{#2}%
                \glsxtrsaveinsert{#2}{#3}%
                \@gls@field@link[\let\glscapscase\@thirdofthree]%
                {#1}{#2}%
                {%
                  \glsifapplyinnerfmtfield{#2}{useriii}%
                  {%
                    \@gls@field@font{\GLSaccessfmtuseriii{#3}{\glsxtrgenentrytextfmt}{#2}}%
                  }%
                  {%
                    \ifx\glscapscase\@thirdofthree
                      \@gls@field@font{\GLSaccessuseriii{#2}%
                        \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
                    \else
                      \@gls@field@font{\glsaccessuseriii{#2}\glsxtrgenentrytextfmt{#3}}%
                    \fi
                  }%
```

```
                }%
              }

\@glsuseriv@  User 4 field.

              \def\@glsuseriv@#1#2[#3]{%
                \def\glsxtrcurrentfield{user4}%
                \glsxtrassignfieldfont{#2}%
                \glsxtrsaveinsert{#2}{#3}%
                \@gls@field@link
                {#1}{#2}%
                {%
                  \glsifapplyinnerfmtfield{#2}{useriv}%
                  {%
                    \@gls@field@font{\glsaccessfmtuseriv{#3}{\glsxtrgenentrytextfmt}{#2}}%
                  }%
                  {%
                    \@gls@field@font{\glsaccessuseriv{#2}\glsxtrgenentrytextfmt{#3}}%
                  }%
                }%
              }

\@Glsuseriv@  First letter uppercase version.

              \def\@Glsuseriv@#1#2[#3]{%
                \def\glsxtrcurrentfield{user4}%
                \glsxtrassignfieldfont{#2}%
                \glsxtrsaveinsert{#2}{#3}%
                \@gls@field@link
                [\let\glscapscase\@secondofthree]%
                 {#1}{#2}%
                {%
                  \glsifapplyinnerfmtfield{#2}{useriv}%
                  {%
                    \@gls@field@font{\Glsaccessfmtuseriv{#3}{\glsxtrgenentrytextfmt}{#2}}%
                  }%
                  {%
                    \@gls@field@font{\Glsaccessuseriv{#2}#3}%
                  }%
                }%
              }

\@GLSuseriv@  All uppercase version.

              \def\@GLSuseriv@#1#2[#3]{%
                \def\glsxtrcurrentfield{user4}%
                \glsxtrassignfieldfont{#2}%
                \glsxtrsaveinsert{#2}{#3}%
                \@gls@field@link[\let\glscapscase\@thirdofthree]%
                {#1}{#2}%
                {%
                  \glsifapplyinnerfmtfield{#2}{useriv}%
```

```
        {%
          \@gls@field@font{\GLSaccessfmtuseriv{#3}{\glsxtrgenentrytextfmt}{#2}}%
        }%
        {%
          \ifx\glscapscase\@thirdofthree
            \@gls@field@font{\GLSaccessuseriv{#2}%
              \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
          \else
            \@gls@field@font{\glsaccessuseriv{#2}\glsxtrgenentrytextfmt{#3}}%
          \fi
        }%
      }%
    }
```

\@glsuserv@ User 5 field.

```
    \def\@glsuserv@#1#2[#3]{%
      \def\glsxtrcurrentfield{user5}%
      \glsxtrassignfieldfont{#2}%
      \glsxtrsaveinsert{#2}{#3}%
      \@gls@field@link
      {#1}{#2}%
      {%
        \glsifapplyinnerfmtfield{#2}{userv}%
        {%
          \@gls@field@font{\glsaccessfmtuserv{#3}{\glsxtrgenentrytextfmt}{#2}}%
        }%
        {%
          \@gls@field@font{\glsaccessuserv{#2}\glsxtrgenentrytextfmt{#3}}%
        }%
      }%
    }
```

\@Glsuserv@ First letter uppercase version.

```
    \def\@Glsuserv@#1#2[#3]{%
      \def\glsxtrcurrentfield{user5}%
      \glsxtrassignfieldfont{#2}%
      \glsxtrsaveinsert{#2}{#3}%
      \@gls@field@link
      [\let\glscapscase\@secondofthree]%
      {#1}{#2}%
      {%
        \glsifapplyinnerfmtfield{#2}{userv}%
        {%
          \@gls@field@font{\Glsfmtfield{#3}{\glsxtrgenentrytextfmt}{#2}{userv}}%
        }%
        {%
          \@gls@field@font{\Glsentryuserv{#2}\glsxtrgenentrytextfmt{#3}}%
        }%
      }%
    }
```

\@GLSuserv@ All uppercase version.

```
\def\@GLSuserv@#1#2[#3]{%
  \def\glsxtrcurrentfield{user5}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{userv}%
    {%
      \@gls@field@font{\GLSaccessfmtuserv{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \ifx\glscapscase\@thirdofthree
        \@gls@field@font{\GLSaccessuserv{#2}%
          \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
      \else
        \@gls@field@font{\glsaccessuserv{#2}\glsxtrgenentrytextfmt{#3}}%
      \fi
    }%
  }%
}
```

\@glsuservi@ User 6 field.

```
\def\@glsuservi@#1#2[#3]{%
  \def\glsxtrcurrentfield{user6}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{uservi}%
    {%
      \@gls@field@font{\glsaccessfmtuservi{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\glsaccessuservi{#2}\glsxtrgenentrytextfmt{#3}}%
    }%
  }%
}
```

\@Glsuservi@ First letter uppercase version.

```
\def\@Glsuservi@#1#2[#3]{%
  \def\glsxtrcurrentfield{user6}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]%
  {#1}{#2}%
```

```
                      {%
                        \glsifapplyinnerfmtfield{#2}{uservi}%
                        {%
                          \@gls@field@font{\Glsaccessfmtuservi{#3}{\glsxtrgenentrytextfmt}{#2}}%
                        }%
                        {%
                          \@gls@field@font{\Glsaccessuservi{#2}\glsxtrgenentrytextfmt{#3}}%
                        }%
                      }%
                    }
```

\@GLSuservi@ All uppercase version.

```
\def\@GLSuservi@#1#2[#3]{%
  \def\glsxtrcurrentfield{user6}%
  \glsxtrassignfieldfont{#2}%
  \glsxtrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{uservi}%
    {%
      \@gls@field@font{\GLSaccessfmtuservi{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \ifx\glscapscase\@thirdofthree
        \@gls@field@font{\GLSaccessuservi{#2}%
          \glsuppercase{\glsxtrgenentrytextfmt{#3}}}%
      \else
        \@gls@field@font{\glsaccessuservi{#2}\glsxtrgenentrytextfmt{#3}}%
      \fi
    }%
  }%
}
```

Commands like \acrshort already set \glsifplural, but they don't set
\glsxtrifwasfirstuse so they need adjusting. These commands shouldn't
be used with \newabbreviation, but the redefinitions below allow for users
reverting \newacronym back to its base definition.

\@@glsxtr@base@acrcmd@warn Warn user that they need to use to new abbreviation commands.

```
\newcommand*{\@@glsxtr@base@acrcmd@warn}[2]{%
  \GlossariesExtraWarning{Base acronym command \string#1\space
  should not be used with new abbreviation definitions. Use
  \string#2\space instead}%
}
```

\@glsxtr@base@acrcmd Warn user that they need to use to new abbreviation commands.

```
\let\@glsxtr@base@acrcmd\@@glsxtr@base@acrcmd@warn
```

The following acr commands don't support innertextformat.

`\@acrshort` No case change.

```
\def\@acrshort#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
  \@glsxtr@base@acrcmd\acrshort\glsxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsaccessshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

`\@Acrshort` First letter uppercase.

```
\def\@Acrshort#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
  \@glsxtr@base@acrcmd\Acrshort\Glsxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccessshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

`\@ACRshort` All uppercase.

```
\def\@ACRshort#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
  \@glsxtr@base@acrcmd\ACRshort\GLSxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
```

```
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@thirdofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \glsuppercase{\acronymfont{\glsaccessshort{#2}}#3}%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
  }
```

\@acrshortpl No case change.

```
  \def\@acrshortpl#1#2[#3]{%
    \def\glsxtrcurrentfield{short}%
    \@glsxtr@base@acrcmd\acrshortpl\glsxtrshortpl
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasglslike\@secondoftwo
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@firstofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\glsaccessshortpl{#2}}#3%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
  }
```

\@Acrshortpl First letter uppercase.

```
  \def\@Acrshortpl#1#2[#3]{%
    \def\glsxtrcurrentfield{short}%
    \@glsxtr@base@acrcmd\Acrshortpl\Glsxtrshortpl
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasglslike\@secondoftwo
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@secondofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\Glsaccessshortpl{#2}}#3%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
  }
```

\@ACRshortpl All uppercase.

```
\def\@ACRshortpl#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
  \@glsxtr@base@acrcmd\ACRshortpl\GLSxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsaccessshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

\@acrlong No case change.

```
\def\@acrlong#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
  \@glsxtr@base@acrcmd\acrlong\glsxtrlong
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsaccesslong{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

\@Acrlong First letter uppercase.

```
\def\@Acrlong#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
  \@glsxtr@base@acrcmd\Acrlong\Glsxtrlong
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
```

134

```
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@secondofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
          \acronymfont{\Glsaccesslong{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%
      \glspostlinkhook
    }
```

\@ACRlong All uppercase.

```
    \def\@ACRlong#1#2[#3]{%
      \def\glsxtrcurrentfield{long}%
      \@glsxtr@base@acrcmd\ACRlong\GLSxtrlong
      \glsdoifexists{#2}%
      {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsxtrifwasglslike\@secondoftwo
        \let\glsxtrifwasfirstuse\@secondoftwo
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@thirdofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
          \glsuppercase{\acronymfont{\glsaccesslong{#2}}#3}%
        }%
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%
      \glspostlinkhook
    }
```

\@acrlongpl No case change.

```
    \def\@acrlongpl#1#2[#3]{%
      \def\glsxtrcurrentfield{long}%
      \@glsxtr@base@acrcmd\acrlongpl\glsxtrlongpl
      \glsdoifexists{#2}%
      {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsxtrifwasglslike\@secondoftwo
        \let\glsxtrifwasfirstuse\@secondoftwo
        \let\glsifplural\@firstoftwo
        \let\glscapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
          \acronymfont{\glsaccesslongpl{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%
      \glspostlinkhook
    }
```

\@Acrlongpl First letter uppercase.

```
\def\@Acrlongpl#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
  \@glsxtr@base@acrcmd\Acrlongpl\Glsxtrlongpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccesslongpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

\@ACRlongpl All uppercase.

```
\def\@ACRlongpl#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
  \@glsxtr@base@acrcmd\ACRlongpl\GLSxtrlongpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsaccesslongpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

The full formats use the internal long and short commands (such as `\@acrshort` and `\@acrlong`). Therefore they don't need adjustments, but they do need clearer warnings. This means three warnings per use (once for the full command and once each for the short and long commands), but at least this way the most important warning (replace `\acrfull` with `\glsxtrfull` etc) is present.

\@acrfull

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

```
              \def\glsxtrcurrentfield{}%
              \@glsxtr@base@acrcmd\acrfull\glsxtrfull
              \acrfullfmt{#1}{#2}{#3}%
            }
```

\@Acrfull

```
            \def\@Acrfull#1#2[#3]{%
              \def\glsxtrcurrentfield{}%
              \@glsxtr@base@acrcmd\Acrfull\Glsxtrfull
              \Acrfullfmt{#1}{#2}{#3}%
            }
```

\@ACRfull

```
            \def\@ACRfull#1#2[#3]{%
              \def\glsxtrcurrentfield{}%
              \@glsxtr@base@acrcmd\ACRfull\GLSxtrfull
              \ACRfullfmt{#1}{#2}{#3}%
            }
```

\@acrfullpl

```
            \def\@acrfullpl#1#2[#3]{%
              \def\glsxtrcurrentfield{}%
              \@glsxtr@base@acrcmd\acrfullpl\glsxtrfullpl
              \acrfullplfmt{#1}{#2}{#3}%
            }
```

\@Acrfullpl

```
            \def\@Acrfullpl#1#2[#3]{%
              \def\glsxtrcurrentfield{}%
              \@glsxtr@base@acrcmd\Acrfullpl\Glsxtrfullpl
              \Acrfullplfmt{#1}{#2}{#3}%
            }
```

\@ACRfullpl

```
            \def\@ACRfullpl#1#2[#3]{%
              \def\glsxtrcurrentfield{}%
              \@glsxtr@base@acrcmd\ACRfullpl\GLSxtrfullpl
              \ACRfullplfmt{#1}{#2}{#3}%
            }
```

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

\@glsaddkey

```
            \renewcommand*{\@glsaddkey}[7]{%
              \key@ifundefined{glossentry}{#1}%
              {%
                \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
                \appto\@gls@keymap{,{#1}{#1}}%
```

```
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
    \newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
```

Now for the commands with links. These currently don't support the inner text format. First the version with no case change:

```
    \ifcsdef{@gls@user@#1@}%
    {%
      \PackageError{glossaries}%
      {Can't define '\string#5' as helper command
       '\expandafter\string\csname @gls@user@#1@\endcsname' already
       exists}%
      {}%
    }%
    {%
      \expandafter\newcommand\expandafter*\expandafter
        {\csname @gls@user@#1\endcsname}[2][]{%
          \new@ifnextchar[%
            {\csuse{@gls@user@#1@}{##1}{##2}}%
            {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
      \csdef{@gls@user@#1@}##1##2[##3]{%
        \def\glsxtrcurrentfield{#1}%
        \glsxtrassignfieldfont{##2}%
        \glsxtrsaveinsert{##2}{##3}%
        \@gls@field@link{##1}{##2}{\@gls@field@font{#3{##2}##3}}%
      }%
      \newrobustcmd*{#5}{%
        \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
    }%
```

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

```
    \ifcsdef{@Gls@user@#1@}%
    {%
      \PackageError{glossaries}%
      {Can't define '\string#6' as helper command
       '\expandafter\string\csname @Gls@user@#1@\endcsname' already
        exists}%
      {}%
    }%
    {%
      \expandafter\newcommand\expandafter*\expandafter
        {\csname @Gls@user@#1\endcsname}[2][]{%
          \new@ifnextchar[%
            {\csuse{@Gls@user@#1@}{##1}{##2}}%
            {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
      \csdef{@Gls@user@#1@}##1##2[##3]{%
        \def\glsxtrcurrentfield{#1}%
```

```
                  \glsxtrassignfieldfont{##2}%
                  \glsxtrsaveinsert{##2}{##3}%
                  \@gls@field@link[\let\glscapscase\@secondofthree]%
                     {##1}{##2}{\@gls@field@font{#4{##2}##3}}%
               }%
               \newrobustcmd*{#6}{%
                  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
            }%
```

Finally the all caps version (modified):

```
            \ifcsdef{@GLS@user@#1@}%
            {%
               \PackageError{glossaries}%
               {Can't define '\string#7' as helper command
                '\expandafter\string\csname @GLS@user@#1@\endcsname' already
                 exists}%
               {}%
            }%
            {%
               \expandafter\newcommand\expandafter*\expandafter
                 {\csname @GLS@user@#1\endcsname}[2][]{%
                    \new@ifnextchar[%
                       {\csuse{@GLS@user@#1@}{##1}{##2}}%
                       {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
               \csdef{@GLS@user@#1@}##1##2[##3]{%
                  \def\glsxtrcurrentfield{#1}%
                  \glsxtrassignfieldfont{##2}%
                  \glsxtrsaveinsert{##2}{##3}%
                  \@gls@field@link[\let\glscapscase\@thirdofthree]%
                     {##1}{##2}{\@gls@field@font{\glsuppercase{#3{##2}##3}}}%
               }%
               \newrobustcmd*{#7}{%
                  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
            }%
```

Add mappings.

```
            \glsmfuaddmap{#3}{#4}%
            \glsmfuaddmap{#5}{#6}%
            \glsmfublocker{#7}%
         }%
         {%
            \PackageError{glossaries-extra}{Key '#1' already exists}{}%
         }%
       }
```

\@gls@link@nocheckfirsthyper Old versions of glossaries don't define this, so provide it just in case it hasn't
                              been defined.

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

ink@postkeys@checkfirsthyper Need another check after preunset and postunset options have been applied.

```
                  \newcommand*{\@gls@link@postkeys@checkfirsthyper}{}
```

139

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

```
\let\@glsxtr@org@checkfirsthyper\@gls@link@checkfirsthyper
\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 to automatically set the value for the commands that change the first use flag. The other commands should set \glsxtrifwasfirstuse to \@secondoftwo (which is done in \@glsxtr@field@linkdefs). Note that if the entry is undefined (as with bib2gls on the first LaTeX run), \ifglsused does neither true nor false parts. However, in that case, this macro won't be called anyway (since it's used in the argument of \glsdoifexistsordo).

```
\ifglsused{\glslabel}%
 {\let\glsxtrifwasfirstuse\@secondoftwo}
 {\let\glsxtrifwasfirstuse\@firstoftwo}%
```

Similarly for \glsxtrifwasglslike

```
\let\glsxtrifwasglslike\@firstoftwo
```

Store the category label for convenience.

```
\protected@edef\glscategorylabel{\glscategory{\glslabel}}%
\glsxtrifwasfirstuse
{%
  \glsifcategoryattribute{\glscategorylabel}{nohyperfirst}{true}%
    {\KV@glslink@hyperfalse}{}%
}%
{%
  \glsifcategoryattribute{\glscategorylabel}{nohypernext}{true}%
    {\KV@glslink@hyperfalse}{}%
}%
\glslinkcheckfirsthyperhook
}
```

\do@glsdisablehyperinlist 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.

```
\ifdef\do@glsdisablehyperinlist
{%
  \let\@glsxtr@do@glsdisablehyperinlist\do@glsdisablehyperinlist
  \renewcommand*{\do@glsdisablehyperinlist}{%
    \@glsxtr@do@glsdisablehyperinlist
    \glsifattribute{\glslabel}{nohyper}{true}{\KV@glslink@hyperfalse}{}%
  }
}
{}
```

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

```
\define@boolkey{glslink}{noindex}[true]{}
\KV@glslink@noindexfalse
```

\@gls@save@glslocal Defined in glossaries v4.50 so may not be defined.

```
\providecommand*{\@gls@save@glslocal}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
```

\@gls@restore@glslocal Defined in glossaries v4.50 so may not be defined.

```
\providecommand*{\@gls@restore@glslocal}{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset\glslocalunset
  \else
    \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
```

\@gls@default@restore@glslocal Save default definition of \@gls@restore@glslocal

```
\let\@gls@default@restore@glslocal\@gls@restore@glslocal
```

\@gls@ignore@restore@glslocal

```
\newcommand*{\@gls@ignore@restore@glslocal}{%
  \let\@gls@do@glsunset\@gobble
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
```

\@gls@do@glsunset Defined in glossaries v4.50 so may not be defined.

```
\providecommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
```

\@gls@default@glslink@opts The noindex setting needs to be initialised, so it's now always set to false first before applying the default options. Otherwise, if noindex is explicitly set in a command like \gls then it won't get reset if the default option list doesn't set it.

```
\newcommand*{\@gls@default@glslink@opts}{}
```

If \@gls@setdefault@glslink@opts has been defined (glossaries v4.20) use it to set the default keys in \@glslink.

\@gls@setdefault@glslink@opts

```
\ifdef\@gls@setdefault@glslink@opts
{
  \renewcommand*{\@gls@setdefault@glslink@opts}{%
    \KV@glslink@noindexfalse
    \expandafter\setupglslink\expandafter{\@gls@default@glslink@opts}%
    \@glsxtrsetaliasnoindex
  }
}
{
```

141

Not defined so prepend it to `\do@glsdisablehyperinlist` to achieve the same effect.

```
\newcommand*{\@gls@setdefault@glslink@opts}{%
  \KV@glslink@noindexfalse
  \expandafter\setupglslink\expandafter{\@gls@default@glslink@opts}%
  \@glsxtrsetaliasnoindex
}
\preto\do@glsdisablehyperinlist{\@gls@setdefault@glslink@opts}
}
```

`\glsxtrsetaliasnoindex` 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.)

```
\providecommand*{\glsxtrsetaliasnoindex}{%
  \KV@glslink@noindextrue
}
```

`\@glsxtrsetaliasnoindex` The change made in v1.46 to remove the grouping has had the knock-on effect of redefining `\glscurrentfieldvalue`, which may be a problem, so v1.47 has changed this to use `\ifcsvoid`.

```
\newcommand*{\@glsxtrsetaliasnoindex}{%
\ifcsvoid{glo@\glsdetoklabel{\glslabel}@alias}%
{}%
{%
  \let\glsxtrindexaliased\@glsxtrindexaliased
  \glsxtrsetaliasnoindex
  \let\glsxtrindexaliased\@no@glsxtrindexaliased
}%
}
```

`\@glsxtrindexaliased`

```
\newcommand{\@glsxtrindexaliased}{%
\ifKV@glslink@noindex
\else
  \begingroup
  \let\@glsnumberformat\@glsxtr@defaultnumberformat

  \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{\glslabel}@counter\endcsname}%
  \glsxtr@saveentrycounter
  \glsxtr@wrglossary@encap{\glsxtralias{\glslabel}}{\@@do@wrglossary{\glsxtralias{\glslabel}}}%
  \endgroup
\fi
}
```

`\@no@glsxtrindexaliased`

```
\newcommand{\@no@glsxtrindexaliased}{%
  \PackageError{glossaries-extra}{\string\glsxtrindexaliased\space
  not permitted outside definition of \string\glsxtrsetaliasnoindex}%
  {}%
}
```

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

```
\let\glsxtrindexaliased\@no@glsxtrindexaliased
```

\GlsXtrSetDefaultGlsOpts Set the default options for \glslink etc.

```
\newcommand*{\GlsXtrSetDefaultGlsOpts}[1]{%
  \renewcommand*{\@gls@default@glslink@opts}{#1}%
}
```

\GlsXtrAppToDefaultGlsOpts

```
\newcommand*{\GlsXtrAppToDefaultGlsOpts}[1]{%
  \appto\@gls@default@glslink@opts{,#1}%
}
```

\GlsXtrPreToDefaultGlsOpts

```
\newcommand*{\GlsXtrPreToDefaultGlsOpts}[1]{%
  \preto\@gls@default@glslink@opts{#1,}%
}
```

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

```
\newcommand*{\glsxtrifindexing}[2]{%
  \ifKV@glslink@noindex #2\else #1\fi
}
```

\glsxtr@@wrglossary@encap{⟨*label*⟩}{⟨*whatsit*⟩}

\glsxtr@wrglossary@encap

Encapsulate indexing whatsit and increment indexed count. See also \glsxtrdowrglossaryhook

```
\newcommand*{\glsxtr@wrglossary@encap}[2]{\glsencapwrcontent{#2\@glsxtr@inc@indexcount{#1}}}
```

Keep track of how many times an entry has been indexed. This doesn't test if the entry has been defined to allow for the first LaTeX run before calling bib2gls.

\@glsxtr@inc@indexcount

```
\newcommand*{\@glsxtr@inc@indexcount}[1]{%
\ifcsdef{glo@\glsdetoklabel{#1}@indexed}
{%
  \csxdef{glo@\glsdetoklabel{#1}@indexed}{%
    \expandafter\number\expandafter\numexpr\csname glo@\glsdetoklabel{#1}@indexed\endcsname+1}%
}%
{%
  \csgdef{glo@\glsdetoklabel{#1}@indexed}{1}%
}%
}
```

`\glsentryindexcount`

```
\newcommand*{\glsentryindexcount}[1]{%
  \ifcsdef{glo@\glsdetoklabel{#1}@indexed}
  {\csuse{glo@\glsdetoklabel{#1}@indexed}}%
  {0}%
}
```

`\glsifindexed`

```
\newcommand*{\glsifindexed}[3]{%
  \ifcsdef{glo@\glsdetoklabel{#1}@indexed}%
  {\expandafter\ifnum\csname glo@\glsdetoklabel{#1}@indexed\endcsname>0 #2\else#3\fi}%
  {#3}%
}
```

`\glsaddallunindexed`

```
\newcommand*{\glsaddallunindexed}[1][\@glo@types]{%
 \forallglsentries[#1]{\@glo@entry}%
 {%
    \glsifindexed{\@glo@entry}{}{\glsadd[format=glsignore]{\@glo@entry}}%
 }%
}
```

`\glsencapwrcontent` This command was added to glossaries v4.50 so may not be defined.

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

`\glswriteentry` Redefine to test for indexonlyfirst category attribute. This needs to use `\GlsXtrIfUnusedOrUndefined` instead of `\ifglsused` to allow it to work with bib2gls.

```
\renewcommand*{\glswriteentry}[2]{%
  \glsxtrifindexing
  {%
   \ifglsindexonlyfirst
     \GlsXtrIfUnusedOrUndefined{#1}
     {#2}%
     {\glsxtrdoautoindexname{#1}{dualindex}}%
   \else
     \glsifattribute{#1}{indexonlyfirst}{true}%
     {%
       \GlsXtrIfUnusedOrUndefined{#1}%
       {#2}%
       {\glsxtrdoautoindexname{#1}{dualindex}}%
     }%
     {#2}%
   \fi
  }%
  {}%
}
```

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

```
\appto\@@do@@wrglossary{\@glsxtr@do@@wrindex
  \glsxtrdowrglossaryhook{\@gls@label}%
}
```

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

Similarly for the "noidx" version:

\gls@noidxglossary

```
\appto\gls@noidxglossary{\@glsxtr@do@@wrindex
  \glsxtrdowrglossaryhook{\@gls@label}%
}
```

\@glsxtr@do@@wrindex

```
\newcommand*{\@glsxtr@do@@wrindex}{%
  \glsxtrdoautoindexname{\@gls@label}{dualindex}%
}
```

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

```
\newcommand*{\glsxtrdowrglossaryhook}[1]{}
```

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

```
\newcommand*{\@gls@alt@hyp@opt}[1]{%
 \let\glslinkvar\@firstofthree

 \def\@gls@hyp@opt@cs{#1}%
 \@ifstar{\s@gls@hyp@opt}%
 {\@ifnextchar+%
   {\@firstoftwo{\p@gls@hyp@opt}}%
   {%
     \expandafter\@ifnextchar\@gls@alt@hyp@opt@char
     {\@firstoftwo{\@alt@gls@hyp@opt}}%
     {#1}%
   }%
 }%
}
```

\@alt@gls@hyp@opt User version

```
\newcommand*{\@alt@gls@hyp@opt}[1][]{%
 \let\glslinkvar\@firstofthree
 \expandafter\@gls@hyp@opt@cs\expandafter[\@gls@alt@hyp@opt@keys,#1]}
```

\@gls@alt@hyp@opt@char Contains the character used as the command modifier.

```
\newcommand*{\@gls@alt@hyp@opt@char}{}
```

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

> `\newcommand*{\@gls@alt@hyp@opt@keys}{}`

`\GlsXtrSetAltModifier`

> `\newcommand*{\GlsXtrSetAltModifier}[2]{%`
> `  \let\@gls@hyp@opt\@gls@alt@hyp@opt`

Check that the supplied character isn't "+" or "*"

> `  \ifstrequal{#1}{+}%`
> `  {\PackageError{glossaries-extra}%`
> `   {Can't use '#1' as modifier (it's already in use)}{}}%`
> `  {%`
> `    \ifstrequal{#1}{*}%`
> `    {\PackageError{glossaries-extra}%`
> `     {Can't use '#1' as modifier (it's already in use)}{}}%`
> `    {}%`
> `  }%`
> `  \def\@gls@alt@hyp@opt@char{#1}%`
> `  \def\@gls@alt@hyp@opt@keys{#2}%`
> `  \ifdefequal\@glsxtr@record@setting\@glsxtr@record@setting@off`
> `  {}%`
> `  {%`

Let `bib2gls` know the modifier.

> `    \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@altmodifier}[1]{}}%`
> `    \protected@write\@auxout{}{\string\@glsxtr@altmodifier{#1}}%`
> `  }%`
> `}`

`\GlsXtrSetPlusModifier` Allow user to override the plus modifier.

> `\newcommand*{\GlsXtrSetPlusModifier}[1]{%`
> ` \renewcommand*{\p@gls@hyp@opt}[1][]{%`
> `   \let\glslinkvar\@thirdofthree`
> `   \@gls@hyp@opt@cs[#1,##1]%`
> ` }%`
> `}`

`\GlsXtrSetStarModifier` Allow user to override the star modifier.

> `\newcommand*{\GlsXtrSetStarModifier}[1]{%`
> ` \renewcommand*{\s@gls@hyp@opt}[1][]{%`
> `   \let\glslinkvar\@secondofthree`
> `   \@gls@hyp@opt@cs[#1,##1]%`
> ` }`
> `}`

`\glsxtr@org@dohyperlink`

> `\let\glsxtr@org@dohyperlink\glsdohyperlink`

\glsnavhyperlink Since \glsnavhyperlink uses \@glslink, it's necessary to patch it uses \glsdohyperlink instead of \glsxtrdohyperlink. The simplest way to achieve this is to locally let \glsxtrdohyperlink to \glsdohyperlink.

This command is provided by glossary-hypernav so it may not exist.

```
\ifdef\glsnavhyperlink
{
  \renewcommand*{\glsnavhyperlink}[3][\@glo@type]{%

    \protected@edef\gls@grplabel{#2}\protected@edef\@gls@grptitle{#3}%
```

Scope:

```
    {%
      \let\glsxtrdohyperlink\glsxtr@org@dohyperlink
      \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
    }%
  }%
}
{}
```

Patch if glossaries pre 4.50.

```
\ifdef\@@gls@navhypertarget
{}
{%
```

\glsnavhypertarget

```
  \renewcommand*{\glsnavhypertarget}{\protect\@@gls@navhypertarget}
```

\@@gls@navhypertarget

```
  \newcommand*{\@@gls@navhypertarget}[3][\@glo@type]{%
    \@glsnavhypertarget{#1}{#2}{#3}%
  }

}%
```

NB glossary-hypernav v4.53 switched to LaTeX3 sequences, so check for the existence of \glsnavhypergroupdotarget:

```
\ifdef\glsnavhypergroupdotarget
{
```

\glsnavhypergroupdotarget

```
  \renewcommand\glsnavhypergroupdotarget[3]{%
    \@glsxtr@do@org@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  }

}
{
```

\@glsnavhypertarget Similarly for \@glsnavhypertarget. (NB this patch should not be used with glossaries v4.53+)

```
    \ifdef\@glsnavhypertarget
```

147

```
                        {%
                          \renewcommand*{\@glsnavhypertarget}[3]{%
                            \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
                            \@glsxtr@do@org@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
                            \ifcsdef{@gls@hypergrouplist@#1}%
                            {%
                              \letcs\@gls@list{@gls@hypergrouplist@#1}%
                              \protected@edef\@gls@thishypernavlabel{#2}%
                              \expandafter\DTLifinlist\expandafter{\@gls@thishypernavlabel}\@gls@list{}%
                              {%
                                \GlossariesWarningNoLine{Navigation panel
                                  for glossary type '#1'^^Jmissing group '#2'}%
                                \gdef\gls@hypergrouprerun{%
                                  \GlossariesWarningNoLine{Navigation panel
                                  has changed. Rerun LaTeX}}%
                              }%
                            }%
                            {%
                              \GlossariesWarningNoLine{Navigation panel
                                for glossary type '#1'^^Jmissing group '#2'}%
                              \gdef\gls@hypergrouprerun{%
                                \GlossariesWarningNoLine{Navigation panel
                                has changed. Rerun LaTeX}}%
                            }%
                          }%
                        }
                        {}
                      }
```

The redefinition of `\glsdohyperlink` has been causing problems so introduce a new command instead.

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

```
\newcommand*{\glsxtrdohyperlink}[2]{%
  \glshasattribute{\glslabel}{targeturl}%
  {%
    \glshasattribute{\glslabel}{targetname}%
    {%
      \glshasattribute{\glslabel}{targetcategory}%
      {%
        \hyperref{\glsgetattribute{\glslabel}{targeturl}}%
```

```
                {\glsgetattribute{\glslabel}{targetcategory}}%
                {\glsgetattribute{\glslabel}{targetname}}%
                {{\glsxtrprotectlinks#2}}%
          }%
          {%
             \hyperref{\glsgetattribute{\glslabel}{targeturl}}%
                {}%
                {\glsgetattribute{\glslabel}{targetname}}%
                {{\glsxtrprotectlinks#2}}%
          }%
        }%
        {%
          \href{\glsgetattribute{\glslabel}{targeturl}}%
             {{\glsxtrprotectlinks#2}}%
        }%
     }%
     {%
```

Check for alias.

```
        \glsfieldfetch{\glslabel}{alias}{\gloaliaslabel}%
        \ifdefvoid\gloaliaslabel
        {%
           \glsxtrhyperlink{#1}{{\glsxtrprotectlinks#2}}%
        }%
        {%
```

Is the alias a multi-entry?

```
           \glsxtrifmulti\gloaliaslabel
           {%
```

Get the main target.

```
              \letcs\gloaliaslabel{@gls@combined@\gloaliaslabel @main}%
           }%
           {}%
```

Redirect link to the alias target.

```
           \glsxtrhyperlink
           {\glolinkprefix\glsdetoklabel{\gloaliaslabel}}%
           {{\glsxtrprotectlinks#2}}%
        }%
     }%
   }
```

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

```
   \ifdef\glsdohyperlinkhook
   {
     \newcommand{\glsxtrhyperlink}[2]{%
       \glsdoshowtarget{#1}{\glsdohyperlinkhook{#1}{#2}\hyperlink{#1}{#2}}%
     }%
   }
   {
```

```
      \newcommand{\glsxtrhyperlink}[2]{%
        \glsdoshowtarget{#1}{\hyperlink{#1}{#2}}%
      }%
    }
```

\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 prob-
lems.

```
    \renewrobustcmd*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
      \glsdoifexists{#2}%
      {%
        \def\@glo@label{#2}%

        {\protected@edef\glslabel{#2}%
        \@glslink{\glolinkprefix\glslabel}{#1}}%
      }%
    }
```

\glsdisablehyper Redefine in case we have an old version of glossaries. This now uses \def rather
than \let to allow for redefinitions of \glsdonohyperlink.

```
    \renewcommand{\glsdisablehyper}{%
      \KV@glslink@hyperfalse
      \def\@glslink{\glsdonohyperlink}%
      \let\@glstarget\@secondoftwo
    }
```

\glsenablehyper This now uses \def rather than \let to allow for redefinitions of \glsdohypertarget
and \glsdohyperlink.

```
    \renewcommand{\glsenablehyper}{%
      \KV@glslink@hypertrue
      \def\@glslink{\glsxtrdohyperlink}%
      \def\@glstarget{\glsdohypertarget}%
    }
```

\glsdonohyperlink This command was only introduced in glossaries v4.20, so it may not be defined
(therefore use \def). 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 (the link text in \hyperlink is also scoped, so it's consistent).

```
    \def\glsdonohyperlink#1#2{{\glsxtrprotectlinks #2}}
```

\@glslink Reset \@glslink with patched versions:

```
    \ifcsundef{hyperlink}%
    {%
      \def\@glslink{\glsdonohyperlink}
    }%
    {%
      \def\@glslink{\glsxtrdohyperlink}
    }
```

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

```
\newcommand*{\glsxtrprotectlinks}{%
  \KV@glslink@hyperfalse
  \KV@glslink@noindextrue
  \let\@gls@\@glsxtr@p@text@
  \let\@Gls@\@Glsxtr@p@text@
  \let\@GLS@\@GLSxtr@p@text@
  \let\@glspl@\@glsxtr@p@plural@
  \let\@Glspl@\@Glsxtr@p@plural@
  \let\@GLSpl@\@GLSxtr@p@plural@
  \let\@glsxtrshort\@glsxtr@p@short@
  \let\@Glsxtrshort\@Glsxtr@p@short@
  \let\@GLSxtrshort\@GLSxtr@p@short@
  \let\@glsxtrlong\@glsxtr@p@long@
  \let\@Glsxtrlong\@Glsxtr@p@long@
  \let\@GLSxtrlong\@GLSxtr@p@long@
  \let\@glsxtrshortpl\@glsxtr@p@shortpl@
  \let\@Glsxtrshortpl\@Glsxtr@p@shortpl@
  \let\@GLSxtrshortpl\@GLSxtr@p@shortpl@
  \let\@glsxtrlongpl\@glsxtr@p@longpl@
  \let\@Glsxtrlongpl\@Glsxtr@p@longpl@
  \let\@GLSxtrlongpl\@GLSxtr@p@longpl@
  \let\@acrshort\@glsxtr@p@acrshort@
  \let\@Acrshort\@Glsxtr@p@acrshort@
  \let\@ACRshort\@GLSxtr@p@acrshort@
  \let\@acrshortpl\@glsxtr@p@acrshortpl@
  \let\@Acrshortpl\@Glsxtr@p@acrshortpl@
  \let\@ACRshortpl\@GLSxtr@p@acrshortpl@
  \let\@acrlong\@glsxtr@p@acrlong@
  \let\@Acrlong\@Glsxtr@p@acrlong@
  \let\@ACRlong\@GLSxtr@p@acrlong@
  \let\@acrlongpl\@glsxtr@p@acrlongpl@
  \let\@Acrlongpl\@Glsxtr@p@acrlongpl@
  \let\@ACRlongpl\@GLSxtr@p@acrlongpl@
}
```

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

\@glsxtr@p@text@

```
\def\@glsxtr@p@text@#1#2[#3]{{\@glstext@{#1}{#2}[#3]}}
```

\@Glsxtr@p@text@

```
\def\@Glsxtr@p@text@#1#2[#3]{{\@Glstext@{#1}{#2}[#3]}}
```

\@GLSxtr@p@text@

```
\def\@GLSxtr@p@text@#1#2[#3]{{\@GLStext@{#1}{#2}[#3]}}
```

`\@glsxtr@p@plural@`

```
\def\@glsxtr@p@plural@#1#2[#3]{{\@glsplural@{#1}{#2}[#3]}}
```

`\@Glsxtr@p@plural@`

```
\def\@Glsxtr@p@plural@#1#2[#3]{{\@Glsplural@{#1}{#2}[#3]}}
```

`\@GLSxtr@p@plural@`

```
\def\@GLSxtr@p@plural@#1#2[#3]{{\@GLSplural@{#1}{#2}[#3]}}
```

`\@glsxtr@p@short@`

```
\def\@glsxtr@p@short@#1#2[#3]{%
 {%
  \glssetabbrvfmt{\glscategory{#2}}%
  \glsabbrvfont{\glsentryshort{#2}}#3%
 }%
}
```

`\@Glsxtr@p@short@`

```
\def\@Glsxtr@p@short@#1#2[#3]{%
 {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \glsabbrvfont{\Glsentryshort{#2}}#3%
 }%
}
```

`\@GLSxtr@p@short@`

```
\def\@GLSxtr@p@short@#1#2[#3]{%
  {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \glsuppercase{\glsabbrvfont{\glsentryshort{#2}}#3}%
  }%
}
```

`\@glsxtr@p@shortpl@`

```
\def\@glsxtr@p@shortpl@#1#2[#3]{%
 {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \glsabbrvfont{\glsentryshortpl{#2}}#3%
 }%
}
```

`\@Glsxtr@p@shortpl@`

```
\def\@Glsxtr@p@shortpl@#1#2[#3]{%
 {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \glsabbrvfont{\Glsentryshortpl{#2}}#3%
 }%
}
```

\@GLSxtr@p@shortpl@

```
\def\@GLSxtr@p@shortpl@#1#2[#3]{%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \glsuppercase{\glsabbrvfont{\glsentryshortpl{#2}}#3}%
  }%
}
```

\@glsxtr@p@long@

```
\def\@glsxtr@p@long@#1#2[#3]{{\glsentrylong{#2}#3}}
```

\@Glsxtr@p@long@

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

\@GLSxtr@p@long@

```
\def\@GLSxtr@p@long@#1#2[#3]{%
  {\glsuppercase{\glslongfont{\glsentrylong{#2}}#3}}}
```

\@glsxtr@p@longpl@

```
\def\@glsxtr@p@longpl@#1#2[#3]{{\glsentrylongpl{#2}#3}}
```

\@Glsxtr@p@longpl@

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

\@GLSxtr@p@longpl@

```
\def\@GLSxtr@p@longpl@#1#2[#3]{%
  {\glsuppercase{\glslongfont{\glsentrylongpl{#2}}#3}}}
```

\@glsxtr@p@acrshort@

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

\@Glsxtr@p@acrshort@

```
\def\@Glsxtr@p@acrshort@#1#2[#3]{{\acronymfont{\Glsentryshort{#2}}#3}}
```

\@GLSxtr@p@acrshort@

```
\def\@GLSxtr@p@acrshort@#1#2[#3]{%
  {\glsuppercase{\acronymfont{\glsentryshort{#2}}#3}}}
```

\@glsxtr@p@acrshortpl@

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

\@Glsxtr@p@acrshortpl@

```
\def\@Glsxtr@p@acrshortpl@#1#2[#3]{{\acronymfont{\Glsentryshortpl{#2}}#3}}
```

\@GLSxtr@p@acrshortpl@

```
\def\@GLSxtr@p@acrshortpl@#1#2[#3]{%
  {\glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}}}
```

\@glsxtr@p@acrlong@

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

\@Glsxtr@p@acrlong@

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

\@GLSxtr@p@acrlong@

```
\def\@GLSxtr@p@acrlong@#1#2[#3]{%
 {\glsuppercase{\glsentrylong{#2}#3}}}
```

\@glsxtr@p@acrlongpl@

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

\@Glsxtr@p@acrlongpl@

```
\def\@Glsxtr@p@acrlongpl@#1#2[#3]{{\Glsentrylongpl{#2}#3}}
```

\@GLSxtr@p@acrlongpl@

```
\def\@GLSxtr@p@acrlongpl@#1#2[#3]{%
 {\glsuppercase{\glsentrylongpl{#2}#3}}}
```

Commands to minimise conflict.

\@glsxtrp@opt

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

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

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

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

```
\newcommand*{\glossxtrsetpopts}{%
  \glsxtrsetpopts{noindex}%
}
```

\glsxtrpInit Initialisation code at the start of the group inserted by \@@glsxtrp.

```
\newcommand{\glsxtrpInit}[2]{\let\glspostlinkhook\relax}
```

\@@glsxtrp

```
\newrobustcmd*{\@@glsxtrp}[2]{%
```

Add scope.

```
  {%
    \glsxtrpInit{#1}{#2}%
    \csname#1\expandafter\endcsname\expandafter[\@glsxtrp@opt]{#2}[]%
  }%
}
```

\@glsxtrp

```
\newrobustcmd*{\@glsxtrp}[2]{%
  \ifcsdef{gls#1}%
  {%
    \@@glsxtrp{gls#1}{#2}%
  }%
  {%
    \ifcsdef{glsxtr#1}%
    {%
      \@@glsxtrp{glsxtr#1}{#2}%
    }%
    {%
      \PackageError{glossaries-extra}{'#1' not recognised by
        \string\glsxtrp}{}%
    }%
  }%
}
```

\@Glsxtrp

```
\newrobustcmd*{\@Glsxtrp}[2]{%
  \ifcsdef{Gls#1}%
  {%
    \@@glsxtrp{Gls#1}{#2}%
  }%
  {%
    \ifcsdef{Glsxtr#1}%
    {%
      \@@glsxtrp{Glsxtr#1}{#2}%
    }%
    {%
      \PackageError{glossaries-extra}{'#1' not recognised by
        \string\Glsxtrp}{}%
    }%
  }%
}
```

\@GLSxtrp

```
\newrobustcmd*{\@GLSxtrp}[2]{%
  \ifcsdef{GLS#1}%
  {%
    \@@glsxtrp{GLS#1}{#2}%
  }%
  {%
    \ifcsdef{GLSxtr#1}%
    {%
      \@@glsxtrp{GLSxtr#1}{#2}%
    }%
    {%
      \PackageError{glossaries-extra}{'#1' not recognised by
```

```
                      \string\GLSxtrp}{}%
                  }%
                }%
              }
```

\glsxtrifintoc

```
                  \newcommand{\glsxtrifintoc}[2]{#2}
```

\glsxtrifheaduc

```
                  \newcommand*{\glsxtrifheaduc}[3]{%
                   \glsxtrifintoc{#3}{\glsifattribute{#1}{headuc}{true}{#2}{#3}}%
                  }
```

\glsxtr@entry@p

```
                  \newrobustcmd*{\glsxtr@headentry@p}[2]{%
                   \glsxtrifheaduc{#1}%
                   {%
                     \glsuppercase{\@gls@entry@field{#1}{#2}}%
                   }%
                   {%
                     \@gls@entry@field{#1}{#2}%
                   }%
                  }
```

\glsxtrp Not robust as it needs to expand somewhat.

```
                  \newcommand{\glsxtrp}[2]{%
                   \protect\NoCaseChange
                   {%
                     \protect\glstexorpdfstring
                     {%
                       \protect\glsxtrifinmark
                       {%
                         \ifcsdef{glsxtrhead#1}%
                         {%
                           {\protect\csuse{glsxtrhead#1}{#2}}%
                         }%
                         {%
                           \glsxtr@headentry@p{#2}{#1}%
                         }%
                       }%
                       {%
                         \@glsxtrp{#1}{#2}%
                       }%
                     }%
                     {%
                       \protect\@gls@entry@field{#2}{#1}%
                     }%
                   }%
                  }
```

156

Provide short synonyms for the most common option.

\glsps

```
\newcommand*{\glsps}{\glsxtrp{short}}
```

\glspt

```
\newcommand*{\glspt}{\glsxtrp{text}}
```

\Glsxtrp As above but use first letter upper case.

```
\newcommand{\Glsxtrp}[2]{%
  \protect\NoCaseChange
  {%
    \protect\glstexorpdfstring
    {%
      \protect\glsxtrifinmark
      {%
        \ifcsdef{Glsxtrhead#1}%
        {%
          {\protect\csuse{Glsxtrhead#1}{#2}}%
        }%
        {%
```

\@Gls@entry@field is robust as from glossaries v4.50, but continue to use \protect in case an older version is installed.

```
          \protect\@Gls@entry@field{#2}{#1}%
        }%
      }%
      {%
        \@Glsxtrp{#1}{#2}%
      }%
    }%
    {%
      \MFUsentencecase{\@gls@entry@field{#2}{#1}}%
    }%
  }%
}
\glsmfuaddmap{\glsxtrp}{\Glsxtrp}
```

\GLSxtrp As above but all upper case. The bookmarks use \glsuppercase, which is expandable as from mfirstuc v2.08+.

```
\newcommand{\GLSxtrp}[2]{%
  \protect\NoCaseChange
  {%
    \protect\glstexorpdfstring
    {%
      \protect\glsxtrifinmark
      {%
        \ifcsdef{GLSxtr#1}%
        {%
          {\protect\GLSxtrshort[noindex,hyper=false]{#1}[]}%
```

```
        }%
        {%
          \protect\glsuppercase
          {%
            \protect\@gls@entry@field{#2}{#1}%
          }%
        }%
      }%
      {%
        \@GLSxtrp{#1}{#2}%
      }%
    }%
    {%
      \protect\GLSxtrusefield{#2}{#1}%
    }%
  }%
}
\glsmfublocker{\GLSxtrp}
```

Provide case-changing versions of synonyms.

\Glsps

```
\newcommand*{\Glsps}{\Glsxtrp{short}}
\glsmfuaddmap{\glsps}{\Glsps}
```

\GLSps

```
\newcommand*{\GLSps}{\GLSxtrp{short}}
\glsmfublocker{\GLSps}
```

\Glspt

```
\newcommand*{\Glspt}{\Glsxtrp{text}}
\glsmfuaddmap{\glspt}{\Glspt}
```

\GLSpt

```
\newcommand*{\GLSpt}{\GLSxtrp{text}}
\glsmfublocker{\GLSpt}
```

1.3.5 Entry Counting

The (use) 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 \cgls instead. This form of entry counting is provided to adjust the formatting if the number of times an entry has been used (through commands that unset the first use flag) doesn't exceeding the specified threshold. For link counting, see §1.4.

First adjust definitions of the unset and reset commands to provide a hook, but changing the flag can cause problems in certain situations, so to allow the normal unsetting to be temporarily disabled, \@glsunset is let to

\@glsxtr@unset, which performs the actual unsetting through \@@glsunset and then does the hook. This means that the unsetting (and the hook) can switched off by redefining \@glsunset and then switched back on again by changing the definition back to \@glsxtr@unset.

\@glsxtr@unset Global unset.

```
\newcommand*{\@glsxtr@unset}[1]{%
  \@@glsunset{#1}%
  \glsxtrpostunset{#1}%
}%
```

\@glsunset Global unset.

```
\let\@glsunset\@glsxtr@unset
```

\glsxtrpostunset

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

Provide a command to store a list of labels that will need unsetting.

\GlsXtrStartUnsetBuffering

```
\newcommand*{\GlsXtrStartUnsetBuffering}{%
  \@ifstar\s@GlsXtrStartUnsetBuffering\@GlsXtrStartUnsetBuffering
}
```

\@GlsXtrStartUnsetBuffering Unstarred version doesn't check for duplicates.

```
\newcommand*{\@GlsXtrStartUnsetBuffering}{%
  \let\@glsxtr@org@unset@buffer\@glsxtr@unset@buffer
  \GlsXtrClearUnsetBuffer
  \let\@glsunset\@glsxtrbuffer@unset
  \let\org@glsxtrbuffer@check@repeats\@glsxtrbuffer@check@repeats
  \renewcommand*{\@glsxtrbuffer@check@repeats}{%
   \@@glsxtrbuffer@check@repeats
  }%
}
```

\s@GlsXtrStartUnsetBuffering Starred version checks for duplicates.

```
\newcommand*{\s@GlsXtrStartUnsetBuffering}{%
  \let\@glsxtr@org@unset@buffer\@glsxtr@unset@buffer
  \GlsXtrClearUnsetBuffer
  \let\@glsunset\@glsxtrbuffer@nodup@unset
  \let\org@glsxtrbuffer@check@repeats\@glsxtrbuffer@check@repeats
  \renewcommand*{\@glsxtrbuffer@check@repeats}{%
   \@@glsxtrbuffer@check@repeats
  }%
}
```

\@glsxtrbuffer@unset This must use a global change since \gls may have to be placed inside \mbox (for example, with soul commands).

```
\newcommand*{\@glsxtrbuffer@unset}[1]{%
```

159

```
                                  \listxadd\@glsxtr@unset@buffer{#1}%
                               }
```

\@glsxtrbuffer@nodup@unset Alternative version that avoids duplicates. One level of expansion is performed
on the argument in case it's a control sequence containing the label. (Not using
\xifinlist as the added complexity might cause problems that the buffering
is trying to overcome.)

```
\newcommand*{\@glsxtrbuffer@nodup@unset}[1]{%
  \expandafter\ifinlist\expandafter{#1}{\@glsxtr@unset@buffer}{}%
  {\listxadd\@glsxtr@unset@buffer{#1}}%
}
```

\@glsxtrbuffer@check@repeats

```
\newcommand*{\@glsxtrbuffer@check@repeats}{}
```

\@@glsxtrbuffer@check@repeats

```
\newcommand*{\@@glsxtrbuffer@check@repeats}{}
```

\buffer@check@repeats@notused

```
\newcommand*{\@@glsxtrbuffer@check@repeats@notused}{}
```

\glsxtrbuffer@do@check@repeat

```
\newrobustcmd*{\@@glsxtrbuffer@do@check@repeat}{%
  \expandafter\ifinlist\expandafter{\glslabel}{\@glsxtr@unset@buffer}%
  {\@glslocalunset{\glslabel}}%
  {\GlsXtrIfUnusedOrUndefined\glslabel
    {\listxadd\@@glsxtrbuffer@check@repeats@notused{\glslabel}}{}}%
}
```

\UnsetBufferEnableRepeatLocal

```
\newcommand*{\GlsXtrUnsetBufferEnableRepeatLocal}{%
  \def\@@glsxtrbuffer@check@repeats{\@@glsxtrbuffer@do@check@repeat}%
  \def\@@glsxtrbuffer@check@repeats@notused{}%
}
```

\setBufferDisableRepeatLocal

```
\newcommand*{\GlsXtrUnsetBufferDisableRepeatLocal}{%
  \def\@@glsxtrbuffer@check@repeats{}%
  \def\@@glsxtrbuffer@check@repeats@notused{}%
}
```

\GlsXtrResetLocalBuffer

```
\newcommand*{\GlsXtrResetLocalBuffer}{%
 \forlistloop\@glslocalreset\@@glsxtrbuffer@check@repeats@notused
 \GlsXtrClearUnsetBuffer
}
```

`\GlsXtrClearUnsetBuffer`

```
\newcommand*{\GlsXtrClearUnsetBuffer}{%
  \def\@@glsxtrbuffer@check@repeats@notused{}%
  \def\@glsxtr@unset@buffer{}%
}
```

`\GlsXtrStopUnsetBuffering`

```
\newcommand*{\GlsXtrStopUnsetBuffering}{%
  \@ifstar\s@GlsXtrStopUnsetBuffering\@GlsXtrStopUnsetBuffering
}
```

`\@GlsXtrStopUnsetBuffering` Unstarred form (global unset).

```
\newcommand*{\@GlsXtrStopUnsetBuffering}{%
  \let\@glsunset\@glsxtr@unset
  \forlistloop\@glsunset\@glsxtr@unset@buffer
  \let\@glsxtr@unset@buffer\@glsxtr@org@unset@buffer
  \let\@glsxtrbuffer@check@repeats\org@glsxtrbuffer@check@repeats
}
```

`\s@GlsXtrStopUnsetBuffering` Starred form (local unset).

```
\newcommand*{\s@GlsXtrStopUnsetBuffering}{%
  \forlistloop\@glslocalunset\@glsxtr@unset@buffer
  \let\@glsunset\@glsxtr@unset
  \let\@glsxtrbuffer@check@repeats\org@glsxtrbuffer@check@repeats
}
```

`\GlsXtrDiscardUnsetBuffering` Discards pending buffer and restores `\glsunset`.

```
\newcommand*{\GlsXtrDiscardUnsetBuffering}{%
  \let\@glsunset\@glsxtr@unset
  \let\@glsxtr@unset@buffer\@glsxtr@org@unset@buffer
  \let\@glsxtrbuffer@check@repeats\org@glsxtrbuffer@check@repeats
}
```

`\GlsXtrForUnsetBufferedList` Iterate over labels stored in the current buffer. The argument is the handler macro.

```
\newcommand*{\GlsXtrForUnsetBufferedList}[1]{%
  \forlistloop#1\@glsxtr@unset@buffer
}
```

`\@glslocalunset` Local unset.

```
\renewcommand*{\@glslocalunset}[1]{%
\@@glslocalunset{#1}%
\glsxtrpostlocalunset{#1}%
}%
```

`\glsxtrpostlocalunset`

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

\@glsreset Global reset.

```
\renewcommand*{\@glsreset}[1]{%
  \@@glsreset{#1}%
  \glsxtrpostreset{#1}%
}%
```

\glsxtrpostreset

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

\@glslocalreset Local reset.

```
\renewcommand*{\@glslocalreset}[1]{%
  \@@glslocalreset{#1}%
  \glsxtrpostlocalreset{#1}%
}%
```

\glsxtrpostlocalreset

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

\glslocalreseteach Locally reset a list of entries.

```
\newcommand*{\glslocalreseteach}[1]{%
  \gls@ifnotmeasuring
  {%
    \@for\@gls@thislabel:=#1\do{%
      \glsdoifexists{\@gls@thislabel}%
      {%
        \@glslocalreset{\@gls@thislabel}%
      }%
    }%
  }%
}
```

\glslocalunseteach Locally unset a list of entries.

```
\newcommand*{\glslocalunseteach}[1]{%
  \gls@ifnotmeasuring
  {%
    \@for\@gls@thislabel:=#1\do{%
      \glsdoifexists{\@gls@thislabel}%
      {%
        \@glslocalunset{\@gls@thislabel}%
      }%
    }%
  }%
}
```

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

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

Enable entry counting:

```
\glsenableentrycount
```

Redefine \gls etc:

```
\renewcommand*{\gls}{\cgls}%
\renewcommand*{\Gls}{\cGls}%
\renewcommand*{\glspl}{\cglspl}%
\renewcommand*{\Glspl}{\cGlspl}%
\renewcommand*{\GLS}{\cGLS}%
\renewcommand*{\GLSpl}{\cGLSpl}%
```

Set the entrycount attribute:

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

In case this command is used again:

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

\lsxtr@setentrycountunsetattr

```
\newcommand*{\@glsxtr@setentrycountunsetattr}[2]{%
 \@for\@glsxtr@cat:=#1\do
 {%
   \ifdefempty{\@glsxtr@cat}{}%
   {%
     \glssetcategoryattribute{\@glsxtr@cat}{entrycount}{#2}%
   }%
 }%
}
```

\ifglsresetcurrcount   Determine whether or not to reset the entry counter when the first use flag is reset. This conditional will already be defined with glossaries v4.50+.

```
\ifdef\glsresetcurrcountfalse{}{\newif\ifglsresetcurrcount}
\glsresetcurrcountfalse
```

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

\glsenableentrycount

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

Enable new fields:

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

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

```
\renewcommand*{\gls@defdocnewglossaryentry}{%
  \renewcommand*\newglossaryentry[2]{%
    \PackageError{glossaries}{\string\newglossaryentry\space
    may only be used in the preamble when entry counting has
    been activated}{If you use \string\glsenableentrycount\space
    you must place all entry definitions in the preamble not in
```

163

```
      the document environment}%
    }%
  }%
```

New commands to access new fields:

```
\newcommand*{\glsentrycurrcount}[1]{%
 \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
 {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
 \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
 {0}{\@gls@entry@field{##1}{prevcount}}%
}%
```

Adjust post unset and reset:

```
\let\@glsxtr@entrycount@org@unset\glsxtrpostunset
\renewcommand*{\glsxtrpostunset}[1]{%
  \@glsxtr@entrycount@org@unset{##1}%
  \@gls@increment@currcount{##1}%
}%
\let\@glsxtr@entrycount@org@localunset\glsxtrpostlocalunset
\renewcommand*{\glsxtrpostlocalunset}[1]{%
  \@glsxtr@entrycount@org@localunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\let\@glsxtr@entrycount@org@reset\glsxtrpostreset
\renewcommand*{\glsxtrpostreset}[1]{%
  \@glsxtr@entrycount@org@reset{##1}%
  \ifglsresetcurrcount
      \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
  \fi
}%
\let\@glsxtr@entrycount@org@localreset\glsxtrpostlocalreset
\renewcommand*{\glsxtrpostlocalreset}[1]{%
  \@glsxtr@entrycount@org@localreset{##1}%
  \ifglsresetcurrcount
      \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
  \fi
}%
```

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

```
\let\@cgls@\@@cgls@
\let\@cglspl@\@@cglspl@

\let\@cGls@\@@cGls@
\let\@cGlspl@\@@cGlspl@
\let\@cGLS@\@@cGLS@
\let\@cGLSpl@\@@cGLSpl@
```

The rest is as the original definition.

```
                        \AtEndDocument{\@gls@write@entrycounts}%
                        \renewcommand*{\@gls@entry@count}[2]{%
                          \csgdef{glo@\glsdetoklabel{##1}@prevcount}{##2}%
                        }%
                        \let\glsenableentrycount\relax
                        \renewcommand*{\glsenableentryunitcount}{%
                          \PackageError{glossaries-extra}{\string\glsenableentryunitcount\space
                           can't be used with \string\glsenableentrycount}%
                          {Use one or other but not both commands}%
                        }%
                      }
```

newglossaryentry@defcounters Allow for docdef=restricted.

```
                      \renewcommand*{\@@newglossaryentry@defcounters}{%
                       \csdef{glo@\@glo@label @currcount}{0}%
                       \ifnum\@glsxtr@docdefval=2\relax
                        \ifcsdef{glo@\@glo@label @prevcount}{}{\csdef{glo@\@glo@label @prevcount}{0}}%
                       \else
                        \csdef{glo@\@glo@label @prevcount}{0}%
                       \fi
                      }
```

\@gls@write@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.

```
                      \renewcommand*{\@gls@write@entrycounts}{%
                        \immediate\write\@auxout
                          {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
                        \count@=0\relax
                        \forallglsentries{\@glsentry}{%
                          \glshasattribute{\@glsentry}{entrycount}%
                          {%
                            \ifglsused{\@glsentry}%
                            {%
                              \immediate\write\@auxout
                                {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
                            }%
                            {}%
                            \advance\count@ by \@ne
                          }%
                          {}%
                        }%
                        \ifnum\count@=0
                          \GlossariesExtraWarningNoLine{Entry counting has been enabled
                           \MessageBreak with \string\glsenableentrycount\space but the
                           \MessageBreak attribute 'entrycount' hasn't
                           \MessageBreak been assigned to any of the defined
                           \MessageBreak entries}%
                        \fi
                      }
```

\glsxtrifcounttrigger

<div style="border: 1px solid; background: #FFFFCC; padding: 10px;">
$\glsxtrifcounttrigger\{\langle label\rangle\}\{\langle trigger\ format\rangle\}\{\langle normal\rangle\}$
</div>

```
\newcommand*{\glsxtrifcounttrigger}[3]{%
 \glshasattribute{#1}{entrycount}%
 {%
   \ifnum\glsentryprevcount{#1}>\glsgetattribute{#1}{entrycount}\relax
    #3%
   \else
    #2%
   \fi
 }%
 {#3}%
}
```

Actual internal definitions of `\cgls` used when entry counting is enabled.

\@@cgls@

```
\def\@@cgls@#1#2[#3]{%
 \glsxtrifcounttrigger{#2}%
 {%
   \cglsformat{#2}{#3}%
   \glsunset{#2}%
 }%
 {%
   \@gls@{#1}{#2}[#3]%
 }%
}%
```

\@@cglspl@

```
\def\@@cglspl@#1#2[#3]{%
 \glsxtrifcounttrigger{#2}%
 {%
   \cglsplformat{#2}{#3}%
   \glsunset{#2}%
 }%
 {%
   \@glspl@{#1}{#2}[#3]%
 }%
}%
```

\@@cGls@

```
\def\@@cGls@#1#2[#3]{%
 \glsxtrifcounttrigger{#2}%
 {%
   \cGlsformat{#2}{#3}%
   \glsunset{#2}%
 }%
```

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

\@@cGlspl@

                  \def\@@cGlspl@#1#2[#3]{%
                    \glsxtrifcounttrigger{#2}%
                    {%
                      \cGlsplformat{#2}{#3}%
                      \glsunset{#2}%
                    }%
                    {%
                      \@Glspl@{#1}{#2}[#3]%
                    }%
                  }%

  \@@cGLS@

                  \def\@@cGLS@#1#2[#3]{%
                    \glsxtrifcounttrigger{#2}%
                    {%
                      \cGLSformat{#2}{#3}%
                      \glsunset{#2}%
                    }%
                    {%
                      \@GLS@{#1}{#2}[#3]%
                    }%
                  }%

\@@cGLSpl@

                  \def\@@cGLSpl@#1#2[#3]{%
                    \glsxtrifcounttrigger{#2}%
                    {%
                      \cGLSplformat{#2}{#3}%
                      \glsunset{#2}%
                    }%
                    {%
                      \@GLSpl@{#1}{#2}[#3]%
                    }%
                  }%
```

Remove default warnings from `\cgls` etc so that it can be used interchangeable with `\gls` etc.

  \@cgls@

```
                  \def\@cgls@#1#2[#3]{\@gls@{#1}{#2}[#3]}
```

  \@cGls@

```
                  \def\@cGls@#1#2[#3]{\@Gls@{#1}{#2}[#3]}
```

\@cglspl@

```
\def\@cglspl@#1#2[#3]{\@glspl@{#1}{#2}[#3]}
```

\@cGlspl@

```
\def\@cGlspl@#1#2[#3]{\@Glspl@{#1}{#2}[#3]}
```

Add all upper case versions not provided by glossaries.

\cGLS

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

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

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

\@cGLS@

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

\cGLSformat Format used by \cGLS if entry only used once on previous run. The first argu-
ment is the label, the second argument is the insert text.

```
\newcommand*{\cGLSformat}[2]{%
  \expandafter\glsuppercase\expandafter{\cglsformat{#1}{#2}}%
}
```

\cGLSpl

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

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

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

\@cGLSpl@

```
\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.

```
\newcommand*{\cGLSplformat}[2]{%
  \expandafter\glsuppercase\expandafter{\cglsplformat{#1}{#2}}%
}
```

Modify the trigger formats to check for the regular attribute.

168

```
\renewcommand*{\cglsformat}[2]{%
  \glsifregular{#1
  {\glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}}#2%
}
```

```
\renewcommand*{\cGlsformat}[2]{%
  \glsifregular{#1
  {\Glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}}#2%
}
```

```
\renewcommand*{\cglsplformat}[2]{%
  \glsifregular{#1
  {\glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}}#2%
}
```

```
\renewcommand*{\cGlsplformat}[2]{%
  \glsifregular{#1
  {\Glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}}#2%
}
```

New code similar to above for unit counting.

```
\newcommand*{\@@newglossaryentry@defunitcounters}{%
  \protected@edef\@glo@countunit{\csuse{@glsxtr@categoryattr@@\@glo@category @unitcount}}%
  \ifdefvoid\@glo@countunit
  {}%
  {%
    \@glsxtr@ifunitcounter{\@glo@countunit}%
    {}%
    {\expandafter\@glsxtr@addunitcounter\expandafter{\@glo@countunit}}%
  }%
}
```

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

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

\@glsxtr@addunitcounter

```
\newcommand*{\@glsxtr@addunitcounter}[1]{%
  \listadd{\@glsxtr@unitcountlist}{#1}%
```

169

```
                                    \ifcsundef{glsxtr@theunit@#1}
                                    {%
                                      \ifcsdef{theH#1}%
                                      {\csdef{glsxtr@theunit@#1}{\csuse{theH#1}}}%
                                      {\csdef{glsxtr@theunit@#1}{\csuse{the#1}}}%
                                    }%
                                    {}%
                                    }

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

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

  \@glsxtr@previousunitcount
                                    \newcommand*\@glsxtr@previousunitcount[1]{%
                                    glo@\glsdetoklabel{#1}@prevunit@\glsgetattribute{#1}{unitcount}.%
                                    \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
                                    }

@gls@increment@currunitcount
                                    \newcommand*{\@gls@increment@currunitcount}[1]{%
                                      \glshasattribute{#1}{unitcount}%
                                      {%
                                        \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{#1}}%
                                        \ifcsundef{\@glsxtr@csname}%
                                        {%
                                          \csgdef{\@glsxtr@csname}{1}%
                                          \listcsxadd
                                          {glo@\glsdetoklabel{#1}@unitlist}%
                                          {\glsgetattribute{#1}{unitcount}.%
                                           \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
                                          }%
                                        }%
                                        {%
                                          \csxdef{\@glsxtr@csname}%
                                          {\number\numexpr\csname\@glsxtr@csname\endcsname+1}%
                                        }%
                                      }%
                                      {}%
                                    }

ocal@increment@currunitcount
```

170

```
\newcommand*{\@gls@local@increment@currunitcount}[1]{%
 \glshasattribute{#1}{unitcount}%
 {%
   \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{#1}}%
   \ifcsundef{\@glsxtr@csname}%
   {%
     \csdef{\@glsxtr@csname}{1}%
     \listcseadd
      {glo@\glsdetoklabel{#1}@unitlist}%
      {\glsgetattribute{#1}{unitcount}.%
       \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
      }%
   }%
   {%
     \csedef{\@glsxtr@csname}%
     {\number\numexpr\csname\@glsxtr@csname\endcsname+1}%
   }%
 }%
 {}%
}
```

```
\newcommand*{\@glsxtr@currunitcount}[2]{%
 \ifcsundef
 {glo@\glsdetoklabel{#1}@currunit@#2}%
 {0}%
 {\csuse{glo@\glsdetoklabel{#1}@currunit@#2}}%
}%
```

```
\newcommand*{\@glsxtr@prevunitcount}[2]{%
 \ifcsundef
 {glo@\glsdetoklabel{#1}@prevunit@#2}%
 {0}%
 {\csuse{glo@\glsdetoklabel{#1}@prevunit@#2}}%
}%
```

```
\newcommand*{\glsenableentryunitcount}{%
```

Enable new fields:

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

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

```
\renewcommand*{\gls@defdocnewglossaryentry}{%
 \renewcommand*\newglossaryentry[2]{%
   \PackageError{glossaries}{\string\newglossaryentry\space
   may only be used in the preamble when entry counting has
   been activated}{If you use \string\glsenableentryunitcount\space
   you must place all entry definitions in the preamble not in
```

```
      the document environment}%
    }%
  }%
```

New commands to access new fields:

```
  \newcommand*{\glsentrycurrcount}[1]{%
    \@glsxtr@currunitcount{##1}{\glsgetattribute{##1}{unitcount}.%
     \csuse{glsxtr@theunit@\glsgetattribute{##1}{unitcount}}}%
  }%
  \newcommand*{\glsentryprevcount}[1]{%
    \@glsxtr@prevunitcount{##1}{\glsgetattribute{##1}{unitcount}.%
     \csuse{glsxtr@theunit@\glsgetattribute{##1}{unitcount}}}%
  }%
```

Access total count:

```
  \newcommand*{\glsentryprevtotalcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@prevunittotal}%
    {0}%
    {%
      \number\csuse{glo@\glsdetoklabel{##1}@prevunittotal}
    }%
  }%
```

Access max value:

```
  \newcommand*{\glsentryprevmaxcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@prevunitmax}%
    {0}%
    {%
      \number\csuse{glo@\glsdetoklabel{##1}@prevunitmax}
    }%
  }%
```

Adjust post unset and reset:

```
  \let\@glsxtr@entryunitcount@org@unset\glsxtrpostunset
  \renewcommand*{\glsxtrpostunset}[1]{%
    \@glsxtr@entryunitcount@org@unset{##1}%
    \@gls@increment@currunitcount{##1}%
  }%
  \let\@glsxtr@entryunitcount@org@localunset\glsxtrpostlocalunset
  \renewcommand*{\glsxtrpostlocalunset}[1]{%
    \@glsxtr@entryunitcount@org@localunset{##1}%
    \@gls@local@increment@currunitcount{##1}%
  }%
  \let\@glsxtr@entryunitcount@org@reset\glsxtrpostreset
  \renewcommand*{\glsxtrpostreset}[1]{%
    \glshasattribute{##1}{unitcount}%
    {%

      \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{##1}}%
      \ifcsundef{\@glsxtr@csname}%
      {}%
      {\ifglsresetcurrcount\csgdef{\@glsxtr@csname}{0}\fi}%
```

```
    }%
    {}%
}%
\let\@glsxtr@entryunitcount@org@localreset\glsxtrpostlocalreset
\renewcommand*{\glsxtrpostlocalreset}[1]{%
  \@glsxtr@entryunitcount@org@localreset{##1}%
  \glshasattribute{##1}{unitcount}%
  {%

    \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{##1}}%
    \ifcsundef{\@glsxtr@csname}%
    {}%
    {\ifglsresetcurrcount\csdef{\@glsxtr@csname}{0}\fi}%
  }%
  {}%
}%
```

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

```
\let\@cgls@\@@cgls@
\let\@cglspl@\@@cglspl@

\let\@cGls@\@@cGls@
\let\@cGlspl@\@@cGlspl@
\let\@cGLS@\@@cGLS@
\let\@cGLSpl@\@@cGLSpl@
```

Write information to the aux file.

```
\AtEndDocument{\@gls@write@entryunitcounts}%
\renewcommand*{\@gls@entry@unitcount}[3]{%
  \csgdef{glo@\glsdetoklabel{##1}@prevunit@##3}{##2}%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevunittotal}%
  {\csgdef{glo@\glsdetoklabel{##1}@prevunittotal}{##2}}%
  {%
    \csxdef{glo@\glsdetoklabel{##1}@prevunittotal}{
      \number\numexpr\csuse{glo@\glsdetoklabel{##1}@prevunittotal}+##2}%
  }%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevunitmax}%
  {\csgdef{glo@\glsdetoklabel{##1}@prevunitmax}{##2}}%
  {%
    \ifnum\csuse{glo@\glsdetoklabel{##1}@prevunitmax}<##2
      \csgdef{glo@\glsdetoklabel{##1}@prevunitmax}{##2}%
    \fi
  }%
}%
\let\glsenableentryunitcount\relax
\renewcommand*{\glsenableentrycount}{%
  \PackageError{glossaries-extra}{\string\glsenableentrycount\space
   can't be used with \string\glsenableentryunitcount}%
  {Use one or other but not both commands}%
}%
```

```
                               }
                               \@onlypreamble\glsenableentryunitcount
```

\@gls@entry@unitcount

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

gls@write@entryunitcounts@do

```
                               \newcommand*{\@gls@write@entryunitcounts@do}[1]{%
                                 \immediate\write\@auxout
                                  {\string\@gls@entry@unitcount
                                    {\@glsentry}%
                                    {\@glsxtr@currunitcount{\@glsentry}{#1}%
                                    }%
                                    {#1}}%
                               }
```

\@gls@write@entryunitcounts

```
                               \newcommand*{\@gls@write@entryunitcounts}{%
                                 \immediate\write\@auxout
                                   {\string\providecommand*{\string\@gls@entry@unitcount}[3]{}}%
                                 \count@=0\relax
                                 \forallglsentries{\@glsentry}{%
                                   \glshasattribute{\@glsentry}{unitcount}%
                                   {%
                                     \ifglsused{\@glsentry}%
                                     {%
                                       \forlistcsloop
                                         {\@gls@write@entryunitcounts@do}%
                                         {glo@\glsdetoklabel{\@glsentry}@unitlist}%
                                     }%
                                     {}%
                                     \advance\count@ by \@ne
                                   }%
                                   {}%
                                 }%
                                 \ifnum\count@=0
                                   \GlossariesExtraWarningNoLine{Entry counting has been enabled
                                    \MessageBreak with \string\glsenableentryunitcount\space but the
                                    \MessageBreak attribute 'unitcount' hasn't
                                    \MessageBreak been assigned to any of the defined
                                    \MessageBreak entries}%
                                 \fi
                               }
```

lsXtrEnableEntryUnitCounting 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.

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

Enable entry counting:

```
                                 \glsenableentryunitcount
```

174

Redefine \gls etc:

```
\renewcommand*{\gls}{\cgls}%
\renewcommand*{\Gls}{\cGls}%
\renewcommand*{\glspl}{\cglspl}%
\renewcommand*{\Glspl}{\cGlspl}%
\renewcommand*{\GLS}{\cGLS}%
\renewcommand*{\GLSpl}{\cGLSpl}%
```

Set the entrycount attribute:

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

In case this command is used again:

```
\let\GlsXtrEnableEntryUnitCounting\@glsxtr@setentryunitcountunsetattr
\renewcommand*{\GlsXtrEnableEntryCounting}[2]{%
 \PackageError{glossaries-extra}{\string\GlsXtrEnableEntryCounting\space
  can't be used with \string\GlsXtrEnableEntryUnitCounting}%
 {Use one or other but not both commands}}%
}
```

```
\newcommand*{\@glsxtr@setentryunitcountunsetattr}[3]{%
 \@for\@glsxtr@cat:=#1\do
 {%
   \ifdefempty{\@glsxtr@cat}{}%
   {%
     \glssetcategoryattribute{\@glsxtr@cat}{entrycount}{#2}%
     \glssetcategoryattribute{\@glsxtr@cat}{unitcount}{#3}%
   }%
 }%
}
```

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

```
\renewcommand*{\SetGenericNewAcronym}{%
```

Make sure \RestoreAcronyms has been used.

```
\ifdefequal\@addtoacronymlists\@glsxtr@org@addtoacronynlists
{}%
{%
  \GlossariesWarning{\string\SetGenericNewAcronym\space used
  without restoring base acronym functions with
  \string\RestoreAcronyms}%
```

175

```
    }%
    \let\@Gls@entryname\@Gls@acrentryname
Redefine \newacronym:
    \renewcommand{\newacronym}[4][]{%
      \ifdefempty{\@glsacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
      {}%
      \glskeylisttok{##1}%
      \glslabeltok{##2}%
      \glsshorttok{##3}%
      \glslongtok{##4}%
      \newacronymhook
      \protected@edef\@do@newglossaryentry{%
        \noexpand\newglossaryentry{\the\glslabeltok}%
        {%
          type=\acronymtype,%
          name={\expandonce{\acronymentry{##2}}},%
          sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
          text={\the\glsshorttok},%
          short={\the\glsshorttok},%
          shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
          long={\the\glslongtok},%
          longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
          category=acronym,%
          \GenericAcronymFields,%
          \the\glskeylisttok
        }%
      }%
      \@do@newglossaryentry
    }%
    \renewcommand*{\acrfullfmt}[3]{%
      \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
    \renewcommand*{\Acrfullfmt}[3]{%
      \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
    \renewcommand*{\ACRfullfmt}[3]{%
      \glslink[##1]{##2}{%
        \glsuppercase{\genacrfullformat{##2}{##3}}}}%
    \renewcommand*{\acrfullplfmt}[3]{%
      \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
    \renewcommand*{\Acrfullplfmt}[3]{%
      \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
    \renewcommand*{\ACRfullplfmt}[3]{%
      \glslink[##1]{##2}{%
        \glsuppercase{\genplacrfullformat{##2}{##3}}}}%
    \renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}%
```

```
                      \renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
                      \renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
                      \renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
                     }
```

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:

```
\let\@glsxtr@org@setacronymstyle\setacronymstyle
\let\@glsxtr@org@newacronymstyle\newacronymstyle
```

Save the list of acronyms in case they are required.

\@glsxtr@acronymlists

```
\let\@glsxtr@acronymlists\@glsacronymlists
```

\@glsxtr@org@addtoacronynlists

```
\let\@glsxtr@org@addtoacronynlists\@addtoacronymlists
```

\@glsxtr@org@setacronymlists

```
\let\@glsxtr@org@setacronymlists\SetAcronymLists
```

Need to provide a replacement for \forallacronyms since \@glsacronymlists isn't available.

\@glsxtr@abbrlists

```
\newcommand{\@glsxtr@abbrlists}{}
```

\forallabbreviationlists

```
\newcommand*{\forallabbreviationlists}[2]{%
  \@for#1:=\@glsxtr@abbrlists\do{\ifdefempty{#1}{}{#2}}%
}
```

\@glsxtr@addabbreviationlist

```
\newcommand*{\@glsxtr@addabbreviationlist}[1]{%
  \protected@edef\@glo@type{#1}%
  \ifdefempty\@glsxtr@abbrlists
  {\let\@glsxtr@abbrlists\@glo@type}%
  {%
    \ifdefequal\@glsxtr@abbrlists\@glo@type
    {}%
    {%
      \expandafter\DTLifinlist\expandafter{\@glo@type}{\@glsxtr@abbrlists}{}%
      {\protected@eappto\@glsxtr@abbrlists{,\@glo@type}}%
    }%
  }%
}
```

`\forallacronyms` Modify to add warning.

```
\renewcommand*{\forallacronyms}[2]{%
  \@glsxtr@base@acrcmd\forallacronyms\forallabbreviationlists
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
```

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

```
\newcommand*{\MakeAcronymsAbbreviations}{%
```

Undo acronym display style:

```
\@for\@gls@type:=\@glsacronymlists\do{%
  \csgdef{gls@\@gls@type @entryfmt}{\glsentryfmt}%
}%
```

Save and clear acronym list.

```
\let\@glsxtr@acronymlists\@glsacronymlists
\let\@glsacronymlists\@empty
\let\@addtoacronymlists\@gobble
\let\SetAcronymLists\@gobble
```

Warn if `\acrshort` etc are used.

```
\let\@glsxtr@base@acrcmd\@@glsxtr@base@acrcmd@warn
```

Redefine `\newacronym` to use same interface as `\newabbreviation`.

```
\renewcommand*{\newacronym}[4][]{%
  \glsxtr@newabbreviation{type=\acronymtype,category=acronym,##1}{##2}{##3}{##4}%
}%
\renewcommand*{\firstacronymfont}[1]{\glsfirstabbrvfont{##1}}%
\renewcommand*{\acronymfont}[1]{\glsabbrvfont{##1}}%
\renewcommand*{\setacronymstyle}[1]{%
  \PackageError{glossaries-extra}{\string\setacronymstyle{##1}
  unavailable.
  Use \string\setabbreviationstyle[acronym]\space instead.
  The original acronym interface can be restored with
  \string\RestoreAcronyms}{}%
}%
\renewcommand*{\newacronymstyle}[1]{%
  \GlossariesExtraWarning{New acronym style '##1' won't be
  available unless you restore the original acronym interface with
  \string\RestoreAcronyms}%
  \@glsxtr@org@newacronymstyle{##1}%
}%
}
```

Switch acronyms to abbreviations:

```
\MakeAcronymsAbbreviations
```

`\RestoreAcronyms` Restore acronyms to glossaries interface.

```
\newcommand*{\RestoreAcronyms}{%
```

Restore acronym list.

```
\let\@glsacronymlists\@glsxtr@acronymlists
\let\@addtoacronymlists\@glsxtr@org@addtoacronynlists
\let\SetAcronymLists\@glsxtr@org@setacronymlists
```

Suppress warnings if \acrshort etc are used.

```
\let\@glsxtr@base@acrcmd\@gobbletwo
```

Restore acronym display style:

```
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDefaultAcronymDisplayStyle{\@gls@type}%
}%
```

Switch to the generic acronym mechanism.

```
\SetGenericNewAcronym
\renewcommand{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand{\acronymfont}[1]{##1}%
\let\setacronymstyle\@glsxtr@org@setacronymstyle
\let\newacronymstyle\@glsxtr@org@newacronymstyle
```

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

```
\renewcommand*\@gls@link@checkfirsthyper{%
  \ifglsused{\glslabel}%
  {\let\glsxtrifwasfirstuse\@secondoftwo}
  {\let\glsxtrifwasfirstuse\@firstoftwo}%
  \@glsxtr@org@checkfirsthyper
}
\glssetcategoryattribute{acronym}{regular}{false}%
\setacronymstyle{long-short}%
}
```

\glsacspace   Allow the user to customise the maximum value.

```
\renewcommand*{\glsacspace}[1]{%
  \glsmeasurewidth{\dimen@}{(\firstacronymfont{\glsentryshort{#1}})}%
  \ifdim\dimen@<\glsacspacemax~\else\space\fi
}
```

\glsacspacemax   Value used in the above.

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

\glsabspace   Similar to \glsacspace but includes inner formatting.

```
\newrobustcmd*{\glsabspace}[1]{%
  \glsmeasurewidth{\dimen@}{(\glsfirstabbrvfont{\glsentryshort{#1}})}%
  \ifdim\dimen@<\glsacspacemax
    \glsxtrgenentrytextfmt{~}%
  \else
    \glsxtrgenentrytextfmt{ }%
  \fi
}
```

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`.

\@glsxtr@reg@glosslist

```
\newcommand*{\@glsxtr@reg@glosslist}{}
```

Save the original definition of `\makeglossaries`:

```
\let\@glsxtr@org@makeglossaries\makeglossaries
```

…saries@warn@noprintglossary This command was only introduced to glossaries v4.47 so it may not be defined.

```
\providecommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
```

\@domakeglossaries glossaries v4.45 introduced `\@domakeglossaries` to provide a way of disabling `\makeglossaries`. If it hasn't been defined, define here to do its argument:

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

\@gls@automake@types Added to glossaries v4.50 so may not be defined.

```
\providecommand{\@gls@automake@types}{\@glo@types}
```

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. The optional argument version shouldn't be used with record.

\makeglossaries

```
\renewcommand*{\makeglossaries}[1][]{%
 \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
   \edef\glsindexingsetting{bib2gls-\ifglsxindy xindy\else makeindex\fi}%
 \else
   \ifglsxindy
     \def\glsindexingsetting{xindy}%
```

```
    \else
      \def\glsindexingsetting{makeindex}%
    \fi
  \fi
  \@domakeglossaries
  {%
    \@glsxtr@if@record@only
    {%
     \PackageError{glossaries-extra}{\string\makeglossaries\space
      not permitted\MessageBreak with record=\@glsxtr@record@setting\space
      package option}%
     {You may only use \string\makeglossaries\space with
      record=off or record=hybrid options}%
    }%
    {%
      \ifblank{#1}%
      {%
          \@glsxtr@org@makeglossaries

          \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
            \let\warn@noprintglossary\@glsxtr@warn@hybrid@noprintgloss
          \fi
      }%
      {%
        \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
          \PackageError{glossaries-extra}{\string\makeglossaries[#1]\space
           not permitted\MessageBreak with record=\@glsxtr@record@setting\space package option}%
           {You may only use the hybrid \string\makeglossaries[...]\space with
            record=off option}%
        \else
        \appto\glsindexingsetting{-noidx}%
```

`\@gls@@automake@immediate` was introduced to glossaries v4.42 so it may not
be defined.

```
            \protected@edef\@glsxtr@reg@glosslist{#1}%
```

`\@gls@@automake@immediate` uses `\@gls@automake@types` as from v4.50. Older
versions use `\@glo@types` which will include the noidx glossaries.

```
            \let\@gls@automake@types\@glsxtr@reg@glosslist
            \ifdef\@gls@@automake@immediate{\@gls@@automake@immediate}{}%
            \ifundef{\glswrite}{\newwrite\glswrite}{}%
            \protected@write\@auxout{}{\string\providecommand
              \string\@glsorder[1]{}}
            \protected@write\@auxout{}{\string\providecommand
              \string\@istfilename[1]{}}
            \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
            \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
            \protected@write\@auxout{}{\string\glsxtr@makeglossaries{#1}}
            \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
```

Iterate through each supplied glossary type and activate it.

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

New glossaries must be created before \makeglossaries:

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

Any subsequence instances of this command should have no effect.

```
        \let\@makeglossary\@gobble
```

Version 1.42 removed letting \makeglossary to \relax (no kernel redefs may
be in effect).

```
        \renewcommand\makeglossaries[1][]{}%
```

Disable all commands that have no effect after \makeglossaries

```
        \@disable@onlypremakeg
```

Allow see key:

```
        \let\gls@checkseeallowed\relax
```

Adjust \@do@seeglossary. This needs to check for the entry's existence but
don't increment associated counter.

```
        \renewcommand*{\@do@seeglossary}[2]{%
          \glsdoifexists{##1}%
          {%
            \protected@edef\@gls@label{\glsdetoklabel{##1}}%
            \protected@edef\@gls@type{\csname glo@\@gls@label @type\endcsname}%
            \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
            {\@glsxtr@org@doseeglossary{##1}{##2}}%
            {%
              \@@glsxtrwrglossmark
              \protected@write\@auxout{}{%
                \string\@gls@reference
                  {\gls@type}{\@gls@label}{\string\glsseeformat##2{}}%
              }%
              \@gls@noidx@addtorefs{\@gls@label}%
            }%
          }%
        }%
```

Adjust \@@do@@wrglossary

```
        \let\@glsxtr@@do@@wrglossary\@@do@@wrglossary
        \def\@@do@@wrglossary{%
          \protected@edef\@gls@type{\csname glo@\@gls@label @type\endcsname}%
          \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
          {\@glsxtr@@do@@wrglossary}%
          {\gls@noidxglossary}%
        }%
```

Suppress warning about no `\makeglossaries`

```
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
```

Only warn for glossaries not listed.

```
\renewcommand{\@gls@noref@warn}[1]{%

  \protected@edef\@gls@type{##1}%
  \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
  {%
    \GlossariesExtraWarning{Can't use
       \string\printnoidxglossary[type={\@gls@type}]
       when '\@gls@type' is listed in the optional argument of
       \string\makeglossaries}%
  }%
  {%
    \GlossariesWarning{Empty glossary for
    \string\printnoidxglossary[type={##1}].
    Rerun may be required (or you may have forgotten to use
    commands like \string\gls)}%
  }%
}%
```

Adjust display number list to check for type:

```
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
  {\@glsxtr@idx@displaynumberlist{##1}}%
  {\@glsxtr@noidx@displaynumberlist{##1}}%
}%
```

Adjust entry list:

```
\renewcommand*{\glsentrynumberlist}[1]{%
  \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
  {\@glsxtr@idx@entrynumberlist{##1}}%
  {\@glsxtr@noidx@entrynumberlist{##1}}%
}%
```

Adjust number list loop

```
\renewcommand*{\glsnumberlistloop}[2]{%
  \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
  {%
     \PackageError{glossaries-extra}{\string\glsnumberlistloop\space
       not available for glossary '##1'}{}%
  }%
  {\@glsxtr@noidx@numberlistloop{##1}{##2}}%
}%
```

Only sanitize sort for normal indexing glossaries.

```
\renewcommand*{\glsprestandardsort}[3]{%
  \expandafter\DTLifinlist\expandafter{##2}{\@glsxtr@reg@glosslist}%
  {%
    \glsdosanitizesort
```

183

```
      }%
      {%
        \ifglssanitizesort
         \@gls@noidx@sanitizesort
        \else

         \@@gls@noidx@nosanitizesort
        \fi
      }%
    }%
```

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

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

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).

```
    \let\@glo@assign@sortkey\@glsxtr@mixed@assign@sortkey
    \renewcommand*{\@printgloss@setsort}{%
```

Need to extract just the type value.

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

Check automake setting:

```
    \ifglsautomake
      \renewcommand*{\@gls@doautomake}{%
        \@for\@gls@type:=\@glsxtr@reg@glosslist\do{%
          \ifdefempty{\@gls@type}{}{\@gls@automake{\@gls@type}}%
        }%
      }%
    \fi
```

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

```
    \ifdef\@glo@check@sortallowed{\@glo@check@sortallowed\makeglossaries}{}%
```

Version 1.6: add rerun check if it's defined.

```
    \ifdef\GlsNoIdxDoRerunCheck
    {\AtEndDocument{\GlsNoIdxDoRerunCheck}}%
    {}%
  \fi
}%
```

Prohibit the use of `\glsxtrnoidxgroups`.

```
\prohibit@glsxtrnoidxgroups
```

Activate warnings for incompatible options.

```
\let\gls@warn@makegloss@incompatible\@gls@warn@makegloss@incompatible
      }%
   }%
}
```

```
\newcommand*{\gls@warn@makegloss@incompatible}[2]{}
```

```
\newcommand*{\@gls@warn@makegloss@incompatible}[2]{%
   #2\GlossariesExtraWarning{#1\space is incompatible with \string\makeglossaries}%
}
```

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

Earlier versions of glossaries-extra simply saved the original version of `\@printglossary` with `\let \@glsxtr@orgprintglossary`. This was later changed to actually defining `\@glsxtr@orgprintglossary` to something similar with some alterations to allow for ignored glossaries, which don't have an associated title and to by-pass the existence check with `\ifglossaryexists` which doesn't recognise ignored glossaries. (bib2gls writes `\provideignoredglossary` to the glstex file for some settings, so the glossary might not been defined on the first LaTeX run and it needs to be allowed with `\printunsrtglossary` on subsequent runs.)

Unfortunately, removing the existence check will cause an error if `\printglossary` is used with an ignored glossary.

As from glossaries v4.46, some new commands have been included to allow the existence check to be varied depending on whether or not ignored glossaries should be allowed, so check for them:

```
\ifdef\@printgloss@checkexists
{\newcommand{\glsxtr@printgloss@checkexists}{\@printgloss@checkexists}}
{\newcommand{\glsxtr@printgloss@checkexists}[2]{#2}}
```

`\@glsxtr@orgprintglossary` (This command is also used for on-the-fly setting.)

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

Add check here.

```
   \def\glossarytitle{%
      \ifcsdef{@glotype@\@glo@type @title}%
      {\csuse{@glotype@\@glo@type @title}}%
```

185

```
        {\glossaryname}}}%
      \def\glossarytoctitle{\glossarytitle}%
      \let\org@glossarytitle\glossarytitle
      \def\@glossarystyle{%
        \ifx\@glossary@default@style\relax
          \GlossariesWarning{No default glossary style provided \MessageBreak
            for the glossary '\@glo@type'. \MessageBreak
            Using fallback. \MessageBreak
            To fix this set the style with \MessageBreak
            \string\setglossarystyle\space or use the \MessageBreak
            style key=value option}%
        \fi
      }%
      \def\gls@dotoctitle{\glssettoctitle{\@glo@type}}%
      \let\@org@glossaryentrynumbers\glossaryentrynumbers
      \bgroup
        \@printgloss@setsort
        \setkeys{printgloss}{#1}%
        \ifx\glossarytitle\org@glossarytitle
        \else
          \cslet{@glotype@\@glo@type @title}{\glossarytitle}%
        \fi
        \let\currentglossary\@glo@type
        \let\org@glossaryentrynumbers\glossaryentrynumbers
        \let\glsnonextpages\@glsnonextpages
        \let\glsnextpages\@glsnextpages

        \glsxtractivatenopost
        \gls@dotoctitle
        \@glossarystyle
        \let\gls@org@glossaryentryfield\glossentry
        \let\gls@org@glossarysubentryfield\subglossentry
        \renewcommand{\glossentry}[1]{%
          \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
          \gls@org@glossaryentryfield{##1}%
        }%
        \renewcommand{\subglossentry}[2]{%
          \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
          \gls@org@glossarysubentryfield{##1}{##2}%
        }%
        \@gls@preglossaryhook
        \glsxtr@printgloss@checkexists{\@glo@type}{#2}%
      \egroup
      \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
      \global\let\warn@noprintglossary\relax
    }
```

\glsxtractivatenopost Change \nopostdesc and \glsxtrnopostpunc to behave as they do in the glossary.

```
\newcommand*{\glsxtractivatenopost}{%
```

```
                        \let\nopostdesc\@nopostdesc
                        \let\glsxtrnopostpunc\@glsxtr@nopostpunc
                      }
```

\glsxtrnopostpunc

```
                      \newrobustcmd*{\glsxtrnopostpunc}{}
```

\@glsxtr@nopostpunc   Provide a command that works like \nopostdesc but only switches off the
                      punctuation without suppressing the post-description hook.

```
                      \newcommand{\@glsxtr@nopostpunc}{%
                       \let\@@glsxtr@org@postdescription\glspostdescription
                       \ifglsnopostdot
                         \renewcommand{\glspostdescription}{%
                           \glsnopostdottrue
                           \let\glspostdescription\@@glsxtr@org@postdescription
                           \let\glsxtrrestorepostpunc\@glsxtr@restore@postpunc
                           \glsxtrpostdescription
                           \@glsxtr@nopostpunc@postdesc}%
                       \else
                         \renewcommand{\glspostdescription}{%
                           \let\glspostdescription\@@glsxtr@org@postdescription
                           \let\glsxtrrestorepostpunc\@glsxtr@restore@postpunc
                           \glsxtrpostdescription
                           \@glsxtr@nopostpunc@postdesc}%
                       \fi
                       \glsnopostdotfalse
                      }
```

\@glsxtr@nopostpunc@postdesc

```
                      \newcommand*{\@glsxtr@nopostpunc@postdesc}{}
```

\@glsxtr@restore@postpunc

```
                      \newcommand*{\@glsxtr@restore@postpunc}{%
                       \def\@glsxtr@nopostpunc@postdesc{%
                         \@glsxtr@org@postdescription
                         \let\@glsxtr@nopostpunc@postdesc\@empty
                         \let\glsxtrrestorepostpunc\@empty
                       }%
                      }
```

\glsxtrrestorepostpunc Does nothing outside of glossary.

```
                      \newcommand*{\glsxtrrestorepostpunc}{}
```

\@printglossary   Redefine.

```
                      \renewcommand{\@printglossary}[2]{%
                       \def\@glsxtr@printglossopts{#1}%
                       \@glsxtr@orgprintglossary{#1}{#2}%
                      }
```

Add a key that switches off the entry targets:

```
\define@choicekey{printgloss}{target}
[\@glsxtr@printglossval\@glsxtr@printglossnr]%
{true,false}[true]%
{%
  \ifcase\@glsxtr@printglossnr

    \def\@glstarget{\glsdohypertarget}%
  \else
    \let\@glstarget\@secondoftwo
  \fi
}
```

`\@glsxtrhypernameprefix`

```
\newcommand{\@glsxtrhypernameprefix}{}
```

New to v1.20:

```
\define@key{printgloss}{targetnameprefix}{%
  \renewcommand{\@glsxtrhypernameprefix}{#1}%
}

\define@key{printgloss}{prefix}{%
  \renewcommand{\glolinkprefix}{#1}%
}

\define@key{printgloss}{label}{%
  \glsxtrsetglossarylabel{#1}%
}

\define@key{printgloss}{preamble}{%
  \renewcommand{\glossarypreamble}{#1}%
}

\define@key{printgloss}{postamble}{%
  \renewcommand{\glossarypostamble}{#1}%
}
```

`\glsxtrsetglossarylabel` Set the label for subsequent glossaries. If the label is fixed (that is, doesn't change with each glossary) this will need to be scoped or changed again to prevent duplicate labels.

```
\newcommand{\glsxtrsetglossarylabel}[1]{%
  \ifstrempty{#1}%
  {%
    \renewcommand*{\@@glossaryseclabel}{}%
  }%
  {%
    \renewcommand*{\@@glossaryseclabel}{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{#1}%
    }%
  }%
}
```

`\@glsxtr@leveloffset`

```
\newcount\@glsxtr@leveloffset
```

New to v1.44:

```
\define@key{printgloss}{leveloffset}{%
  \@glsxtr@assign@leveloffset#1\relax
  \gls@warn@noidxmakegloss@incompatible{option 'leveloffset'}
    {\@glsxtr@leveloffset=0\relax}%
}
```

`\@glsxtr@assign@leveloffset`

```
\newcommand*{\@glsxtr@assign@leveloffset}{%
 \@ifnextchar+{\p@glsxtr@assign@leveloffset}{\np@glsxtr@assign@leveloffset}%
}
```

`\p@glsxtr@assign@leveloffset` Discard initial "+" character.

```
\newcommand*{\p@glsxtr@assign@leveloffset}[1]{%
 \@ifnextchar+{\pp@glsxtr@assign@leveloffset}{\np@glsxtr@assign@leveloffset}%
}
```

`\np@glsxtr@assign@leveloffset`

```
\def\np@glsxtr@assign@leveloffset#1\relax{\@glsxtr@leveloffset=#1\relax}
```

`\pp@glsxtr@assign@leveloffset`

```
\def\pp@glsxtr@assign@leveloffset#1\relax{\advance\@glsxtr@leveloffset by #1\relax}
```

```
\define@boolkey{printgloss}[glsxtr@printgloss@]{groups}[true]{%
 \ifglsxtr@printgloss@groups
 \else
  \gls@warn@noidxmakegloss@incompatible{option 'groups'}%
    {\glsxtr@printgloss@groupstrue}%
 \fi
}
\glsxtr@printgloss@groupstrue

\define@boolkey{printgloss}[glsxtrprintgloss]{flatten}[true]{%
 \ifglsxtrprintglossflatten
  \gls@warn@noidxmakegloss@incompatible{option 'flatten'}%
    {\glsxtrprintglossflattenfalse}%
 \fi
}
\glsxtrprintglossflattenfalse
```

`\glsdohypertarget` Redefine to insert `\@glsxtrhypernameprefix` before the target name.

```
\let\@glsxtr@org@glsdohypertarget\glsdohypertarget
\renewcommand{\glsdohypertarget}[2]{%
  \@glsxtr@org@glsdohypertarget{\@glsxtrhypernameprefix#1}{#2}%
}
```

Update `\@glstarget` to use `\def` instead being assigned with `\let` so that it can pick up the new definition and allow any further redefinitions:

```
\ifx\@glstarget\@glsxtr@org@glsdohypertarget
 \def\@glstarget{\glsdohypertarget}%
\fi
```

`\@glsxtr@do@org@target` Provide a way to locally do the original.

```
\newcommand{\@glsxtr@do@org@target}[2]{%
 {%
  \let\glsdohypertarget\@glsxtr@org@glsdohypertarget
  \@glstarget{#1}{#2}%
 }%
}
```

`\glsxtr@makeglossaries` For the benefit of `makeglossaries`

```
\newcommand*{\glsxtr@makeglossaries}[1]{}
```

`\@glsxtr@gettype` Get just the type.

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

`@glsxtr@mixed@assign@sortkey` Assign the sort key.

```
\newcommand\@glsxtr@mixed@assign@sortkey[1]{%

  \protected@edef\@glo@type{\@glo@type}%
  \expandafter\DTLifinlist\expandafter{\@glo@type}{\@glsxtr@reg@glosslist}%
  {%
    \@glo@no@assign@sortkey{#1}%
  }%
  {%
    \@@glo@assign@sortkey{#1}%
  }%
}%
```

Display number list for the regular version:

`glsxtr@idx@displaynumberlist`

```
\let\@glsxtr@idx@displaynumberlist\glsdisplaynumberlist
```

Display number list for the "noidx" version:

`sxtr@noidx@displaynumberlist`

```
\newcommand*{\@glsxtr@noidx@displaynumberlist}[1]{%
\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
\ifdef\@gls@loclist
{%
  \def\@gls@noidxloclist@sep{%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
```

190

```
            \glsnumlistsep
          }%
          \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
        }%
      }%
      \def\@gls@noidxloclist@finalsep{}%
      \def\@gls@noidxloclist@prev{}%
      \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
      \@gls@noidxloclist@finalsep
      \@gls@noidxloclist@prev
    }%
    {%

      \glsxtrundeftag
      \glsdoifexists{#1}%
      {%
        \GlossariesWarning{Missing location list for '#1'. Either
          a rerun is required or you haven't referenced the entry.}%
      }%
    }%
  }%
}%
```

And for the number list loop:

```
\newcommand*{\@glsxtr@noidx@numberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc#2\relax
  \let\glsseeformat#3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%

    \glsxtrundeftag
    \glsdoifexists{#1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry.}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
```

Same for entry number list.
```

191
```

```
\newcommand*{\@glsxtr@noidx@entrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%

    \glsxtrundeftag
    \glsdoifexists{#1}%
    {%
      \GlossariesWarning{Missing location list for '#1'. Either
        a rerun is required or you haven't referenced the entry.}%
    }%
  }%
}%
```

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

\@gls@noidx@getgrouptitle Patch. Need to take into account new internal token list variable used with new datatool integration.

```
\renewcommand*{\@gls@noidx@getgrouptitle}[2]{%
\protected@edef\@glsxtr@titlelabel{#1}%
\ifdefvoid\@glsxtr@titlelabel
{}%
{%
 \protected@edef\@glsxtr@titlelabel{\csuse{glsxtr@grouptitle@#1}}%
}%
\ifdefvoid{\@glsxtr@titlelabel}%
{%
  \ifcsvoid{l__glossaries_noidx_#1_grouptitle_tl}%
  {%
```

Use old method.

```
    \DTLifint{#1}%
    {%
      \ifnum#1<256\relax
        \edef#2{\char#1\relax}%
      \else
        \edef#2{#1}%
      \fi
    }%
    {%
      \ifcsundef{#1groupname}%
      {\def#2{#1}}%
      {\letcs#2{#1groupname}}%
    }%
  }%
```

```
                    {%
                      \letcs#2{l__glossaries_noidx_#1_grouptitle_tl}%
                    }%
                  }%
                  {%
                    \let#2\@glsxtr@titlelabel
                  }%
                }
```

\glsxtr@org@getgrouptitle Save original definition of \@gls@getgrouptitle

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

\glsxtrnoidxgroups Provide the ability to switch from unsrt to noidx code, but only for record=off.

```
                  \newcommand*{\glsxtrnoidxgroups}{%
                    \ifdefequal\@glsxtr@record@setting\@glsxtr@record@setting@off
                    {%
                     \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
                     \let\glsxtr@org@getgrouptitle\@gls@getgrouptitle
                    }%
                    {\PackageError{glossaries-extra}{Can't use
                     \string\glsxtrunsrtgrouptonoidx\space with record=\@glsxtr@record@setting}
                     {\string\glsxtrunsrtgrouptonoidx\space is only available with record=off}}%
                     \global\let\prohibit@glsxtrnoidxgroups\@glsxtrnoidxgroups@nomakegloss
                  }
```

sxtrnoidxgroups@nomakegloss

```
                  \newcommand{\@glsxtrnoidxgroups@nomakegloss}{%
                   \PackageError{glossaries-extra}{Can't use
                     \string\glsxtrunsrtgrouptonoidx\space with \string\makeglossaries}{}
                  }
```

\prohibit@glsxtrnoidxgroups

```
                  \newcommand{\prohibit@glsxtrnoidxgroups}{%
                   \global\let\glsxtrnoidxgroups\@glsxtrnoidxgroups@nomakegloss
                  }
```

\glsxtrgetgrouptitle 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.

```
                  \newrobustcmd{\glsxtrgetgrouptitle}[2]{%
                    \protected@edef\@glsxtr@titlecsname{glsxtr@grouptitle@#1}%
                    \@onelevel@sanitize\@glsxtr@titlecsname
                    \ifcsdef{\@glsxtr@titlecsname}
                    {\letcs{#2}{\@glsxtr@titlecsname}}%
                    {\glsxtr@org@getgrouptitle{#1}{#2}}%
                  }
                  \let\@gls@getgrouptitle\glsxtrgetgrouptitle
```

\glsxtrsetgrouptitle Sets the title for the given group label.

```
\newcommand{\glsxtrsetgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
  \@onelevel@sanitize\@glsxtr@titlelabel
  \protected@csxdef{\@glsxtr@titlelabel}{#2}%
}
```

\glsxtrlocalsetgrouptitle As above put only locally defines the title.

```
\newcommand{\glsxtrlocalsetgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
  \@onelevel@sanitize\@glsxtr@titlelabel
  \protected@csedef{\@glsxtr@titlelabel}{#2}%
}

\ifdef\glsnavigationitem
{
```

\glsnavigationitem
```
\renewcommand\glsnavigationitem[1]{%
  \glsxtrgetgrouptitle{#1}{\@gls@grptitle}%
  \glsnavhyperlink{#1}{\@gls@grptitle}%
}

}
{
```

\glsnavigation Redefine to use new user-level command. This patch should not be used with glossaries v4.53+.

```
\renewcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergrouplist@\@glo@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergrouplist@\@glo@type\endcsname
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \glsxtrgetgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}
}
```

\@print@noidx@glossary Only redefine if old version (pre v4.57) of glossaries.

```
\ExplSyntaxOn
```

194

```
                    \cs_if_exist:NF \__glossaries_print_noidx:
                     {
                      \renewcommand*{\@print@noidx@glossary}{%
                        \ifcsdef{@glsref@\@glo@type}%
                        {%
                          \ifcsdef{@glo@sortmacro@\@glo@sorttype}%
                          {%
                            \csuse{@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
                          }%
                          {%
                            \PackageError{glossaries}{Unknown ~ sort ~ handler ~ `\@glo@sorttype'}{}%
                          }%
                          \glossarysection[\glossarytoctitle]{\glossarytitle}%
                          \glossarypreamble
```
Moved this command definition outside of environment in case of scoping issues
(e.g. in tabular-like styles).
```
                          \def\@gls@currentlettergroup{}%
                          \begin{theglossary}%
                          \glossaryheader
                          \glsresetentrylist
                          \forlistcsloop{\@gls@noidx@do}{@glsref@\@glo@type}%
                          \end{theglossary}%
                          \glossarypostamble
                        }%
                        {%
```
Add section header if there are actually entries defined in this glossary as the
document is likely pending a re-run.
```
                          \glsxtrifemptyglossary{\@glo@type}%
                          {}%
                          {\glossarysection[\glossarytoctitle]{\glossarytitle}}%
                          \@gls@noref@warn{\@glo@type}%
                        }%
                      }
                     }
                    \ExplSyntaxOff
```

\glsnoidxdisplayloc Patch to check for range formations.
```
                    \renewcommand*{\glsnoidxdisplayloc}[4]{%
                      \setentrycounter[#1]{#2}%
                      \@glsxtr@display@loc#3\empty\end@glsxtr@display@loc{#4}%
                    }
```

\@glsxtr@display@loc Patch to check for range formations.
```
                    \def\@glsxtr@display@loc#1#2\end@glsxtr@display@loc#3{%
                      \ifx#1(\relax
                        \glsxtrdisplaystartloc{#2}{#3}%
                      \else
                        \ifx#1)\relax
```

```
                        \glsxtrdisplayendloc{#2}{#3}%
                      \else
                        \glsxtrdisplaysingleloc{#1#2}{#3}%
                      \fi
                    \fi
                  }
```

\glsxtrdisplaysingleloc Single location.

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

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.

\glsxtrdisplaystartloc Start of a location range.

```
\newcommand*{\glsxtrdisplaystartloc}[2]{%
  \protected@edef\glsxtrlocrangefmt{#1}%
  \ifx\glsxtrlocrangefmt\empty
    \def\glsxtrlocrangefmt{glsnumberformat}%
  \fi
  \expandafter\glsxtrdisplaysingleloc
    \expandafter{\glsxtrlocrangefmt}{#2}%
}
```

\glsxtrdisplayendloc End of a location range.

```
\newcommand*{\glsxtrdisplayendloc}[2]{%
  \protected@edef\@glsxtr@tmp{#1}%
  \ifdefempty{\@glsxtr@tmp}{\def\@glsxtr@tmp{glsnumberformat}}{}%
  \ifx\glsxtrlocrangefmt\@glsxtr@tmp
  \else
    \GlossariesExtraWarning{Mismatched end location range
      (start=\glsxtrlocrangefmt, end=\@glsxtr@tmp)}%
  \fi
  \expandafter\glsxtrdisplayendlochook\expandafter{\@glsxtr@tmp}{#2}%
  \expandafter\glsxtrdisplaysingleloc
    \expandafter{\glsxtrlocrangefmt}{#2}%
  \def\glsxtrlocrangefmt{}%
}
```

\glsxtrdisplayendlochook Allow the user to hook into the end of range command.

```
\newcommand*{\glsxtrdisplayendlochook}[2]{}
```

\glsxtrlocrangefmt Current range format. Empty if not in a range.

```
\newcommand*{\glsxtrlocrangefmt}{}
```

\setentrycounter Adjust \setentrycounter to save the original prefix.

```
\renewcommand*{\setentrycounter}[2][]{%
```

```
                    \def\glsxtrcounterprefix{#1}%
                    \ifx\glsxtrcounterprefix\@empty
                      \def\@glo@counterprefix{.}%
                    \else
                      \def\@glo@counterprefix{.#1.}%
                    \fi
                    \def\glsentrycounter{#2}%
                }
```

\@gls@removespaces Redefine to allow adjustments to location hyperlink.

```
                \def\@gls@removespaces#1 #2\@nil{%
                 \toks@=\expandafter{\the\toks@#1}%
                 \ifx\\#2\\%

                   \edef\@glo@tmp{\the\toks@}%
                   \ifx\@glo@tmp\empty
                   \else
```

Expand location (just in case \toks@ is needed for something else).

```
                    \expandafter\glsxtrlocationhyperlink\expandafter
                      \glsentrycounter\expandafter\@glo@counterprefix\expandafter{\the\toks@}%
                   \fi
                 \else
                   \@gls@ReturnAfterFi{%
                     \@gls@removespaces#2\@nil
                   }%
                 \fi
                }
```

<div style="border:1px solid; background:#ffffcc; padding:8px;">

$\glsxtrlocationhyperlink\{\langle counter\rangle\}\{\langle prefix\rangle\}\{\langle location\rangle\}$

</div>

\glsxtrlocationhyperlink

```
                \newcommand*{\glsxtrlocationhyperlink}[3]{%
                 \ifdefvoid\glsxtrsupplocationurl
                 {%
                   \GlsXtrInternalLocationHyperlink{#1}{#2}{#3}%
                 }%
                 {%
                   \hyperref{\glsxtrsupplocationurl}{}{#1#2#3}{#3}%
                 }%
                }
```

\glsxtrsupphypernumber

```
                \newcommand*{\glsxtrsupphypernumber}[1]{%
                 {%
                   \glshasattribute{\glscurrententrylabel}{externallocation}%
                   {%
                     \def\glsxtrsupplocationurl{%
                       \glsgetattribute{\glscurrententrylabel}{externallocation}}%
```

```
    }%
    {%
      \def\glsxtrsupplocationurl{}%
    }%
    \glshypernumber{#1}%
  }%
}
```

Give a bit of assistance to new users who are confused and don't know how
to read transcript messages.

\@print@glossary

```
\renewcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  {}%
  {\glsxtrNoGlossaryWarning{\@glo@type}}%
  \ifglsxindy
    \ifcsundef{@xdy@\@glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
        }%
      }%
    }%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
              @language\endcsname}}%
        }%
      }%
    }%
  }%
  \@do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}}%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\gls@codepage}}%
    }%
  }%
```

198

```
      \@do@auxoutstuff
   \fi
   \renewcommand*{\@warn@nomakeglossaries}{%
     \GlossariesWarningNoLine{\string\makeglossaries\space
     hasn't been used,^^Jthe glossaries will not be updated}%
   }%
}
```

Setup the warning text to display if the external file for the given glossary is missing.

\GlsXtrNoGlsWarningHead Header message.

```
\newcommand{\GlsXtrNoGlsWarningHead}[2]{%
This document is incomplete. The external file associated with
the glossary '#1' (which should be called \texttt{#2})
hasn't been created.%
}
```

\GlsXtrNoGlsWarningEmptyStart No entries have been added to the glossary.

```
\newcommand{\GlsXtrNoGlsWarningEmptyStart}{%
  This has probably happened because there are no entries defined
  in this glossary.%
}
```

\GlsXtrNoGlsWarningEmptyMain The default "main" glossary is empty.

```
\newcommand{\GlsXtrNoGlsWarningEmptyMain}{%
If you don't want this glossary,
add \texttt{nomain} to your package option list when you load
\texttt{glossaries-extra.sty}. For example:%
}
```

\GlsXtrNoGlsWarningEmptyNotMain A glossary that isn't the default "main" glossary is empty.

```
\newcommand{\GlsXtrNoGlsWarningEmptyNotMain}[1]{%
Did you forget to use \texttt{type=#1} when you defined your
entries? If you tried to load entries into this glossary with
\texttt{\string\loadglsentries} did you remember to use
\texttt{[#1]} as the optional argument? If you did, check that
the definitions in the file you loaded all had the type set
to \texttt{\string\glsdefaulttype}.%
}
```

\GlsXtrNoGlsWarningCheckFile Advisory message to check the file contents.

```
\newcommand{\GlsXtrNoGlsWarningCheckFile}[1]{%
  Check the contents of the file \texttt{#1}. If
  it's empty, that means you haven't indexed any of your entries in this
  glossary (using commands like \texttt{\string\gls} or
  \texttt{\string\glsadd}) so this list can't be generated.
  If the file isn't empty, the document build process hasn't been
  completed.%
}
```

`\GlsXtrNoGlsWarningAutoMake` Message when **automake** option has been used.

```
\newcommand{\GlsXtrNoGlsWarningAutoMake}[1]{%
  You may need to rerun \LaTeX. If you already have, it may be that
  \TeX's shell escape doesn't allow you to run
  \ifglsxindy xindy\else makeindex\fi. Check the
  transcript file \texttt{\jobname.log}. If the shell escape is
  disabled, try one of the following:

  \begin{itemize}
    \item Run the external (Lua) application:

       \texttt{makeglossaries-lite \string"\jobname\string"}

    \item Run the external (Perl) application:

       \texttt{makeglossaries \string"\jobname\string"}
  \end{itemize}

  Then rerun \LaTeX\ on this document.
  \GlossariesExtraWarning{Rerun required to build the
  glossary '#1' or check TeX's shell escape allows
  you to run \ifglsxindy xindy\else makeindex\fi}%
}
```

`\GlsXtrNoGlsWarningMisMatch` Mismatching `\makenoidxglossaries`.

```
\newcommand{\GlsXtrNoGlsWarningMisMatch}{%
  You need to either replace \texttt{\string\makenoidxglossaries}
  with \texttt{\string\makeglossaries} or replace
  \texttt{\string\printglossary} (or \texttt{\string\printglossaries}) with
  \texttt{\string\printnoidxglossary}
  (or \texttt{\string\printnoidxglossaries}) and then rebuild
  this document.%
}
```

`\GlsXtrNoGlsWarningBuildInfo` Build advice.

```
\newcommand{\GlsXtrNoGlsWarningBuildInfo}{%
  Try one of the following:
  \begin{itemize}
    \item Add \texttt{automake} to your package option list when you load
          \texttt{glossaries-extra.sty}. For example:

          \texttt{\string\usepackage[automake]%
              \glsopenbrace glossaries-extra\glsclosebrace}

    \item Run the external (Lua) application:

          \texttt{makeglossaries-lite.lua \string"\jobname\string"}

    \item Run the external (Perl) application:
```

200

```
                        \texttt{makeglossaries \string"\jobname\string"}
                    \end{itemize}

                    Then rerun \LaTeX\ on this document.%
                }
```

\GlsXtrRecordWarning Paragraph for record=only.

```
\newcommand{\GlsXtrRecordWarning}[1]{%
\texttt{\string\printglossary} doesn't work
with the \texttt{record=\@glsxtr@record@setting} package option
use\par\texttt{\string\printunsrtglossary[type=#1]}\par
instead (or change the package option).%
}
```

\GlsXtrNoGlsWarningTail Final paragraph.

```
\newcommand{\GlsXtrNoGlsWarningTail}{%
This message will be removed once the problem has been fixed.%
}
```

\GlsXtrNoGlsWarningNoOut No out file created. Build advice.

```
\newcommand{\GlsXtrNoGlsWarningNoOut}[1]{%
  The file \texttt{#1} doesn't exist. This most likely means you haven't used
  \texttt{\string\makeglossaries} or you have used
  \texttt{\string\nofiles}. If this is just a draft version of the
  document, you can suppress this message using the
  \texttt{nomissingglstext} package option.%
}
```

xtr@defaultnoglossarywarning

```
\newcommand*{\@glsxtr@defaultnoglossarywarning}[1]{%
\glossarysection[\glossarytoctitle]{\glossarytitle}
\GlsXtrNoGlsWarningHead{#1}{\jobname.\csname @glotype@\@glo@type @in\endcsname}
\par
\glsxtrifemptyglossary{#1}%
{%
    \GlsXtrNoGlsWarningEmptyStart\space
    \ifthenelse{\equal{#1}{main}}{\GlsXtrNoGlsWarningEmptyMain\par
    \medskip
    \noindent\texttt{\string\usepackage[nomain\ifglsacronym ,acronym\fi]%
        \glsopenbrace glossaries-extra\glsclosebrace}
    \medskip
    }%
    {\GlsXtrNoGlsWarningEmptyNotMain{#1}}%
}%
{%
    \IfFileExists{\jobname.\csname @glotype@\@glo@type @out\endcsname}
    {%
        \GlsXtrNoGlsWarningCheckFile
```

```
                    {\jobname.\csname @glotype@\@glo@type @out\endcsname}

                \ifglsautomake

                 \GlsXtrNoGlsWarningAutoMake{#1}

                \else

                    \ifthenelse{\equal{#1}{main}}%
                    {%
                      \GlsXtrNoGlsWarningEmptyMain\par
                      \medskip
                      \noindent\texttt{\string\usepackage[nomain]%
                        \glsopenbrace glossaries-extra\glsclosebrace}
                      \medskip
                    }%
                    {}%

                    \ifdefequal\makeglossaries\@no@makeglossaries
                    {%
                      \GlsXtrNoGlsWarningMisMatch
                    }%
                    {%
                      \GlsXtrNoGlsWarningBuildInfo
                    }%
                  \fi
                }%
                {%
                  \GlsXtrNoGlsWarningNoOut
                    {\jobname.\csname @glotype@\@glo@type @out\endcsname}%
                }%
              }%
              \par
              \GlsXtrNoGlsWarningTail
              }
```

tr@record@noglossarywarning Warn about using \printglossary with record

```
\newcommand*{\@glsxtr@record@noglossarywarning}[1]{%
 \GlossariesExtraWarning{\string\printglossary\space doesn't work\MessageBreak
 with record=\@glsxtr@record@setting\space package option\MessageBreak(use
 \string\printunsrtglossary[type=#1])\MessageBreak
 instead (or change the package option)}%
 \glossarysection[\glossarytoctitle]{\glossarytitle}
 \GlsXtrRecordWarning{#1}
 \GlsXtrNoGlsWarningTail
 }
```

Provide some commands to accompany the record option for use with bib2gls.

GlsXtrDefaultResourceOptions Default resource options.

```
\newcommand*{\GlsXtrDefaultResourceOptions}{}
```

\BibGlsOptions Supply global `bib2gls` options. Provided as an alternative to using the command line switches, except for those that must be set on startup.

```
\NewDocumentCommand\BibGlsOptions{m}{%
 \protected@write\@auxout{}{\string\bibgls@options{#1}}%
}
\newcommand{\bibgls@options}[1]{}
\@onlypreamble\BibGlsOptions
```

Allowing arbitrary basename for the .glstex file can cause conflict if multiple documents use the same basename. This is most likely to occur because the user is simply using `\glsxtrresourcefile` as a convenient shortcut to avoid the more lengthy `\GlsXtrLoadResources`[src=⟨*bibname*⟩]. NB if multiple documents need to share the same resource set (including locations), then they should be using the "master" setting.

Therefore, as from v1.55, `\glsxtrresourcefile` is deprecated. A new shortcut command is provided as a drop-in replacement and the original `\glsxtrresourcefile` is changed to an internal command.

\glsxtrresourcefile 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.

```
\newcommand*{\glsxtrresourcefile}[2][]{%
 \GlossariesExtraWarning{\string\glsxtrresourcefile\space is now deprecated
 use \string\glsbibdata[...]{#2} instead}%
 \@glsxtr@resourcefile{#1}{#2}%
}
\@onlypreamble\glsxtrresourcefile
```

\glsbibdata

```
\NewDocumentCommand\glsbibdata{om}{%
\IfValueTF{#1}%
{\GlsXtrLoadResources[src={#2},#1]}%
{\GlsXtrLoadResources[src={#2}]}%
}
\@onlypreamble\glsbibdata
```

\@glsxtr@resourcefile This was the old `\glsxtrresourcefile` command.

```
\newcommand*{\@glsxtr@resourcefile}[2]{%
\@glsxtr@if@record@only
{\renewcommand{\glsindexingsetting}{bib2gls}}%
{\edef\glsindexingsetting{bib2gls-\ifglsxindy xindy\else makeindex\fi}}%
```

The `record` option can't be set after this command.

```
\disable@keys{glossaries-extra.sty}{record}%
\glsxtr@writefields
\glsxtr@save@mfu
\ifdefempty\GlsXtrDefaultResourceOptions
{%
```

```
                    \protected@write\@auxout{\glsxtrresourceinit}%
                    {\string\glsxtr@resource{#1}{#2}}%
                 }%
                 {%
                    \protected@write\@auxout{\glsxtrresourceinit}%
                    {\string\glsxtr@resource{\GlsXtrDefaultResourceOptions,#1}{#2}}%
                 }%
                 \let\@glsxtr@org@see@noindex\@gls@see@noindex
                 \let\@gls@see@noindex\relax
                 \IfFileExists{#2.glstex}%
                 {%
```

Can't scope `\@input` so save and restore the category code of `@` to allow for
internal commands in the location list.

```
                    \edef\@bibgls@restoreat{\noexpand\catcode\noexpand`\noexpand\@=\number\catcode`\@}%
                    \makeatletter
                    \@input{#2.glstex}%
                    \@bibgls@restoreat
```

If the record=nameref option has been set, check if this is supported by the
installed version of bib2gls.

```
                    \@glsxtr@check@bibgls@nameref
                 }%
                 {%
                    \GlossariesExtraWarning{No file '#2.glstex'}%
                 }%
                 \let\@gls@see@noindex\@glsxtr@org@see@noindex
               }
```

\@glsxtr@check@bibgls@nameref This will only warn after bib2gls has created the .glstex file, but there's way
to check before.

```
               \newcommand{\@glsxtr@check@bibgls@nameref}{%
                 \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
                   \ifdef\bibglshrefchar
                   {}%
                   {%
                     \GlossariesExtraWarning{record=nameref requires at least
                     version 1.8 of bib2gls}%
                   }%
                 \fi
                 \let\@glsxtr@check@bibgls@nameref\relax
               }
```

\glsxtrresourceinit Code used during the protected write operation.
               \newcommand*{\glsxtrresourceinit}{}

\glsxtrresourcecount

               \newcount\glsxtrresourcecount
```

\GlsXtrLoadResources Short cut that uses `\@glsxtr@resourcefile` with `\jobname` as the mandatory argument.

```
\NewDocumentCommand\GlsXtrLoadResources{O{}}{%
  \ifnum\glsxtrresourcecount=0\relax
    \@glsxtr@resourcefile{#1}{\jobname}%
  \else
    \@glsxtr@resourcefile{#1}{\jobname-\the\glsxtrresourcecount}%
  \fi
  \advance\glsxtrresourcecount by 1\relax
}
\@onlypreamble\GlsXtrLoadResources
```

\glsxtr@resource

```
\newcommand*{\glsxtr@resource}[2]{}
```

\glsxtrMFUsave

```
\newcommand*{\glsxtrMFUsave}{%
  \ifdef\MFUsave
  {%
    \AtBeginDocument{\MFUsave}%
  }%
  {%
    \GlossariesExtraWarning{mfirstuc.sty too old,
      \string\glsxtrMFUsave\space has no effect. You need to upgrade
      to mfirstuc v2.08}%
  }%
  \let\glsxtrMFUsave\relax
}
```

\glsxtr@save@mfu

```
\ifdef\MFUsave
{
  \newcommand*{\glsxtr@save@mfu}{%
    \glsxtrMFUsave
    \let\glsxtr@save@mfu\relax
  }
}
{
  \newcommand*{\glsxtr@save@mfu}{}
}
```

\glsxtr@fields

```
\newcommand*{\glsxtr@fields}[1]{}
```

\glsxtr@texencoding

```
\newcommand*{\glsxtr@texencoding}[1]{}
```

\glsxtr@locale Used to identify all languages tracked in the document.

```
\newcommand*{\glsxtr@locale}[1]{}
```

205

\glsxtr@langtag Identifies the current language at the time `\glsxtr@writefields` is used.

```
\newcommand*{\glsxtr@langtag}[1]{}
```

\glsxtr@pluralsuffixes

```
\newcommand*{\glsxtr@pluralsuffixes}[4]{}
```

\glsxtr@shortcutsval

```
\newcommand*{\glsxtr@shortcutsval}[1]{}
```

\glsxtr@linkprefix

```
\newcommand*{\glsxtr@linkprefix}[1]{}
```

`\@gls@runshell` and `\@gls@run@output@dir` new to glossaries v4.55 so may
not be defined. If they haven't been defined, provide definitions which will match
old behaviour.

```
\ifglsautomake
  \providecommand{\@gls@run@unrestricted@shell}[1]{\immediate\write18{#1}}
  \providecommand{\@gls@run@output@dir}[1]{}
\fi
```

\glsxtr@writefields This information only needs to be written once, so disable it after it's been used.

```
\newcommand*{\glsxtr@writefields}{%

  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@fields}[1]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@resource}[2]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@pluralsuffixes}[4]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@shortcutsval}[1]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@linkprefix}[1]{}}%
  \protected@write\@auxout{}{\string\glsxtr@fields{\@gls@keymap}}%

  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@record}[5]{}}%

  \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
    \protected@write\@auxout{}%
     {\string\providecommand*{\string\glsxtr@record@nameref}[8]{}}%
  \fi
```

If any languages have been loaded, the language tag will be available in
`\CurrentTrackedLanguageTag` (provided by tracklang). For multilingual doc-
uments, the required locale will have to be indicated in the "sort" key when
using `\GlsXtrLoadResources`.

```
\ifdef\CurrentTrackedLanguageTag
{%
```

```
    \ForEachTrackedDialect{\@glsxtr@currentdialect}{%
      \protected@write\@auxout{}{%
        \string\glsxtr@locale{\GetTrackedLanguageTag\@glsxtr@currentdialect}}%
    }%
    \protected@write\@auxout{}{%
      \string\glsxtr@langtag{\CurrentTrackedLanguageTag}}%
}%
{}%
\protected@write\@auxout{}{\string\glsxtr@pluralsuffixes
  {\glspluralsuffix}{\abbrvpluralsuffix}{\acrpluralsuffix}%
  {\glsxtrabbrvpluralsuffix}}%

\ifdefvoid\inputencodingname
{%
```
Assume UTF-8.
```
    \protected@write\@auxout{}{\string\glsxtr@texencoding{utf8}}%
}%
{%
    \protected@write\@auxout{}{\string\glsxtr@texencoding{\inputencodingname}}%
}%
\protected@write\@auxout{}{\string\glsxtr@shortcutsval{\@glsxtr@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.
```
\AtBeginDocument
  {\protected@write\@auxout{}{\string\glsxtr@linkprefix{\glolinkprefix}}}%
\let\glsxtr@writefields\relax
```
If the automake option is on, try running bib2gls if the aux file exists. This has to be done before the aux file is opened (so package options automake=immediate and automake=true are identical if just bib2gls is used). The double-quotes around \jobname have been removed (v1.19) since \jobname will include double-quotes if the file name has spaces.
```
\ifglsautomake
  \IfFileExists{\jobname.aux}%
  {%
    \@gls@run@unrestricted@shell{bib2gls \@gls@run@output@dir{--dir} \jobname}%
  }{}%
```
If \makeglossaries is also used, allow makeindex/xindy to also be run, otherwise disable the error message about requiring \makeglossaries with automake.
```
  \ifx\@gls@doautomake\@gls@doautomake@err
      \let\@gls@doautomake\relax
  \fi
\fi
```
Check if order=letter has been used by mistake (but not if record=alsoindex has been used).
```
\@glsxtr@if@record@only
{\ifdefstring{\glsorder}{letter}%
```

```
                     {\GlossariesExtraWarningNoLine{Package option 'order=letter' isn't
                      supported with 'record=\@glsxtr@record@setting'. Use 'break-at=none'
                      resource option instead}}%
                     {}%
                   }%
                   {}%
                 }
```

\@glsxtr@do@automake@err glossaries v4.50+ now provides \@gls@do@automake@err so use that if defined.

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

Allow locations specific to a particular counter to be recorded.

\glsxtr@record

```
                 \newcommand*{\glsxtr@record}[5]{}
```

\glsxtr@record@nameref Used with record=nameref to include current label information.

```
                 \newcommand*{\glsxtr@record@nameref}[8]{}
```

\glsxtr@counterrecord Aux file command.

```
                 \newcommand*{\glsxtr@counterrecord}[3]{%
                   \glsxtrfieldlistgadd{#1}{record.#2}{#3}%
                   \glsxtrAddCounterRecordHook{#1}{#2}{#3}%
                 }
```

\glsxtrAddCounterRecordHook User hook.

```
                 \newcommand{\glsxtrAddCounterRecordHook}[3]{}
```

\@glsxtr@counterrecordhook Hook used by \@glsxtr@dorecord.

```
                 \newcommand*{\@glsxtr@counterrecordhook}{}
```

\GlsXtrRecordCounter Activate recording for a particular counter (identified in the argument).

```
                 \newcommand*{\GlsXtrRecordCounter}[1]{%
                   \@@glsxtr@recordcounter{#1}%
                 }
                 \@onlypreamble\GlsXtrRecordCounter
```

```
\newcommand*{\@glsxtr@docounterrecord}[1]{%
  \@bibgls@write@aux{}{\string\glsxtr@counterrecord
    {\@gls@label}{#1}{\csuse{the#1}}}%
}
```

\glsxtrglossentry Users may prefer to have entries displayed throughout the document rather than gathered together in a list. This command emulates the way \glossentry behaves (without the style formatting commands like \item). This needs to define \currentglossary to the current glossary type (normally set at the start of \@printglossary) and needs to define \glscurrententrylabel to the entry's label (normally set before \glossentry and \subglossentry). This needs some protection in case it's used in a section heading.

```
\newcommand*{\glsxtrglossentry}[1]{%
  \glsxtrtitleorpdforheading
  {\@glsxtrglossentry{#1}}%
  {\GlsXtrStandaloneEntryPdfName{#1}}%
  {\GlsXtrStandaloneEntryHeadName{#1}}%
}
```

\@glsxtrglossentry Another test is needed in case \@glsxtrglossentry has been written to the table of contents.

```
\newrobustcmd*{\@glsxtrglossentry}[1]{%
  \glsxtrtitleorpdforheading
  {%
    \glsdoifexists{#1}%
    {%
      \begingroup

        \protected@edef\glscurrententrylabel{\glsdetoklabel{#1}}%
        \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
        \ifglshasparent{#1}%
        {\GlsXtrStandaloneSubEntryItem{#1}}%
        {\glsentryitem{#1}}%
        \GlsXtrStandaloneEntryName{#1}%
      \endgroup
    }%
  }%
  {\GlsXtrStandaloneEntryPdfName{#1}}%
  {\GlsXtrStandaloneEntryHeadName{#1}}%
}
```

```
\newcommand*{\GlsXtrStandaloneEntryHeadName}[1]{%
  \glsxtrheadname{#1}%
}
```

```
\newcommand*{\GlsXtrStandaloneEntryPdfName}[1]{%
```

```
                                 \glsentryname{#1}%
                               }
```

\GlsXtrStandaloneEntryName

```
                               \newcommand*{\GlsXtrStandaloneEntryName}[1]{%
                                 \glstarget{#1}{\glossentryname{#1}}%
                               }
```

\Glsxtrglossentry As \glsxtrglossentry but sentence case.

```
                               \newcommand*{\Glsxtrglossentry}[1]{%
                                 \glsxtrtitleorpdfforheading
                                 {\@Glsxtrglossentry{#1}}%
                                 {\GlsXtrStandaloneEntryPdfNameFirstUc{#1}}%
                                 {\GlsXtrStandaloneEntryHeadNameFirstUc{#1}}%
                               }
```

\@Glsxtrglossentry

```
                               \newrobustcmd*{\@Glsxtrglossentry}[1]{%
                                 \glsxtrtitleorpdfforheading
                                 {%
                                   \glsdoifexists{#1}%
                                   {%
                                     \begingroup
                                       \protected@edef\glscurrententrylabel{\glsdetoklabel{#1}}%
                                       \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
                                       \ifglshasparent{#1}%
                                       {\GlsXtrStandaloneSubEntryItem{#1}}%
                                       {\glsentryitem{#1}}%
                                       \GlsXtrStandaloneEntryNameFirstUc{#1}%
                                     \endgroup
                                   }%
                                 }%
                                 {\GlsXtrStandaloneEntryPdfNameFirstUc{#1}}%
                                 {\GlsXtrStandaloneEntryHeadNameFirstUc{#1}}%
                               }
```

andaloneEntryHeadNameFirstUc

```
                               \newcommand*{\GlsXtrStandaloneEntryHeadNameFirstUc}[1]{%
                                 \Glsxtrheadname{#1}%
                               }
```

:andaloneEntryPdfNameFirstUc Requires new expandable version of \Glsentryname.

```
                               \newcommand*{\GlsXtrStandaloneEntryPdfNameFirstUc}[1]{%
                                 \Glsentryname{#1}%
                               }
```

:rStandaloneEntryNameFirstUc

```
                               \newcommand*{\GlsXtrStandaloneEntryNameFirstUc}[1]{%
                                 \glstarget{#1}{\Glossentryname{#1}}%
                               }
```

To make it easier to adjust the definition of `\currentglossary` within `\glsxtrglossentry`, this expands to the default definition. (If redefined, it must fully expand to the appropriate label.)

```
\newcommand{\GlsXtrStandaloneGlossaryType}{\glsentrytype{\glscurrententrylabel}}
```

Used for sub-entries in standalone format. The argument is the entry's label.

```
\newcommand*{\GlsXtrStandaloneSubEntryItem}[1]{%
  \GlsXtrIfFieldEqNum{level}{#1}{1}{\glssubentryitem{#1}}{}%
}
```

As `\glsxtrglossentry` but uses a different field. First argument is code to use in the header. The second argument is the entry's label. The third argument is the internal field label. This needs to be expandable in case it occurs in a sectioning command so it can't have an optional argument.

```
\newcommand*{\glsxtrglossentryother}[3]{%
  \ifstrempty{#1}%
  {%

    \glsxtrtitleorpdforheading
    {\@glsxtrglossentryother{#2}{#3}{\GlsXtrStandaloneEntryHeadOther{#3}{#2}}}%
    {\GlsXtrStandaloneEntryPdfOther{#2}{#3}}%
    {\GlsXtrStandaloneEntryHeadOther{#3}{#2}}%
  }%
  {%
    \glsxtrtitleorpdforheading
    {\@glsxtrglossentryother{#2}{#3}{#1}}%
    {\GlsXtrStandaloneEntryPdfOther{#2}{#3}}%
    {#1}%
  }%
}
```

<div style="border:1px solid; background:#ffffcc; padding:8px;">

`\glsxtrglossentryother{⟨entry-label⟩}{⟨field⟩}{⟨header⟩}`

</div>

As `\@glsxtrglossentry` but uses a different field.

```
\newrobustcmd*{\@glsxtrglossentryother}[3]{%
  \glsxtrtitleorpdforheading
  {%
    \glsdoifexists{#1}%
    {%
      \begingroup

        \protected@edef\glscurrententrylabel{\glsdetoklabel{#1}}%
        \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
        \ifglshasparent{#1}%
        {\GlsXtrStandaloneSubEntryItem{#1}}%
        {\glsentryitem{#1}}%
```

```
                                  \GlsXtrStandaloneEntryOther{#1}{#2}%
                                \endgroup
                              }%
                            }%
                            {\GlsXtrStandaloneEntryPdfOther{#1}{#2}}%
                            {#3}%
                          }
```

<code>XtrStandaloneEntryHeadOther</code>

```
                          \newcommand*{\GlsXtrStandaloneEntryHeadOther}[2]{%
                          \ifcsdef{glsxtrhead#2}%
                          {\csuse{glsxtrhead#2}{#1}}%
                          {\@gls@entry@field{\NoCaseChange{#1}}{#2}}%
                          }
```

<code>lsXtrStandaloneEntryPdfOther</code>

```
                          \newcommand*{\GlsXtrStandaloneEntryPdfOther}[2]{%
                          \@gls@entry@field{#1}{#2}%
                          }
```

`\GlsXtrStandaloneEntryOther`

```
                          \newcommand*{\GlsXtrStandaloneEntryOther}[2]{%
                            \glstarget{#1}{\glossentrynameother{#1}{#2}}%
                          }
```

`\Glsxtrglossentryother` As `\glsxtrglossentryother` but sentence-case.

```
                          \newcommand*{\Glsxtrglossentryother}[3]{%
                            \ifstrempty{#1}%
                            {%
                              \glsxtrtitleorpdforheading
                              {\@Glsxtrglossentryother{#2}{#3}{\GlsXtrStandaloneEntryHeadOtherFirstUc{#3}{#2}}}%
                              {\GlsXtrStandaloneEntryPdfOtherFirstUc{#2}{#3}}%
                              {\GlsXtrStandaloneEntryHeadOtherFirstUc{#3}{#2}}%
                            }%
                            {%
                              \glsxtrtitleorpdforheading
                              {\@Glsxtrglossentryother{#2}{#3}{#1}}%
                              {\GlsXtrStandaloneEntryPdfOtherFirstUc{#2}{#3}}%
                              {#1}%
                            }%
                          }
```

$$\boxed{\texttt{\textbackslash Glsxtrglossentryother}\{\langle\textit{entry-label}\rangle\}\{\langle\textit{field}\rangle\}\{\langle\textit{header}\rangle\}}$$

`\@Glsxtrglossentryother`

As `\@glsxtrglossentry` but uses a different field.

```
                          \newrobustcmd*{\@Glsxtrglossentryother}[3]{%
                          \glsxtrtitleorpdforheading
                          {%
```

```
                              \glsdoifexists{#1}%
                              {%
                                \begingroup
                                  \protected@edef\glscurrententrylabel{\glsdetoklabel{#1}}%
                                  \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
                                  \ifglshasparent{#1}%
                                  {\GlsXtrStandaloneSubEntryItem{#1}}%
                                  {\glsentryitem{#1}}%
                                  \GlsXtrStandaloneEntryOtherFirstUc{#1}{#2}%
                                \endgroup
                              }%
                            }%
                            {\GlsXtrStandaloneEntryPdfOtherFirstUc{#1}{#2}}%
                            {#3}%
                          }
```

```
                          \newcommand*{\GlsXtrStandaloneEntryHeadOtherFirstUc}[2]{%
                          \ifcsdef{glsxtrhead#2}%
                          {\csuse{glsxtrhead#2}{#1}}%
                          {\@Gls@entry@field{\NoCaseChange{#1}}{#2}}%
                          }
```

```
                          \newcommand*{\GlsXtrStandaloneEntryPdfOtherFirstUc}[2]{%
                          \MFUsentencecase{\@gls@entry@field{#1}{#2}}%
                          }
```

```
                          \newcommand*{\GlsXtrStandaloneEntryOtherFirstUc}[2]{%
                            \glstarget{#1}{\Glossentrynameother{#1}{#2}}%
                          }
```

\glsxtrtarget Similar to \glstarget but will only create the target if the field identified by
              \glsxtrtargetfield has been defined. If the target hasn't been defined, the
              target is created and the target name is saved in the given field. If \glstarget
              is redefined to use this command then duplicate targets can be avoid if the
              same entry appears in multiple glossaries. TODO: possibly extend this to allow
              a comma-separated list of targets in the field?

```
                          \newcommand{\glsxtrtarget}[2]{%
                          \GlsXtrIfFieldUndef{\glsxtrtargetfield}{#1}%
                          {%
                            \@glstarget{\glolinkprefix #1}{#2}%
                            \xGlsXtrSetField{#1}{\glsxtrtargetfield}{\glolinkprefix #1}%
                          }%
                          {\glsxtrtargetdup{#1}{#2}}%
                          }
```

\glsxtrtargetdup

```
                          \newcommand{\glsxtrtargetdup}[2]{#2}
```

\glsxtrtargetfield The field name used by \glsxtrtarget.

```
\newcommand{\glsxtrtargetfield}{target}
```

\printunsrtglossary Similar to \printnoidxglossary but it displays all entries defined for the given glossary without sorting. Check for \@printgloss@checkexists which was introduced to glossaries v4.46.

```
\ifdef\@printgloss@checkexists
{
  \newcommand*{\printunsrtglossary}{%
    \let\@printgloss@checkexists\@printgloss@checkexists@allowignored
    \@ifstar\s@printunsrtglossary\@printunsrtglossary
  }
}
{
  \newcommand*{\printunsrtglossary}{%
    \@ifstar\s@printunsrtglossary\@printunsrtglossary
  }
}
```

\@printunsrtglossary Unstarred version.

```
\newcommand*{\@printunsrtglossary}[1][]{%
  \@printglossary{type=\glsdefaulttype,#1}{\@print@unsrt@glossary}%
}
```

\s@printunsrtglossary Starred version.

```
\newcommand*{\s@printunsrtglossary}[2][]{%
  \begingroup
    #2%
    \@printglossary{type=\glsdefaulttype,#1}{\@print@unsrt@glossary}%
  \endgroup
}
```

\printunsrtglossaries Similar to \printnoidxglossaries but it displays all entries defined for the given glossary without sorting.

```
\newcommand*{\printunsrtglossaries}{%
  \forallglossaries{\@@glo@type}{\printunsrtglossary[type=\@@glo@type]}%
}
```

\@print@unsrt@glossary

```
\newcommand*{\@print@unsrt@glossary}{%
  \glossarysection[\glossarytoctitle]{\glossarytitle}%
  \glossarypreamble
```

check for empty list

```
  \glsxtrifemptyglossary{\@glo@type}%
  {%
    \GlossariesExtraWarning{No entries defined in glossary `\@glo@type'}%
  }%
  {%
```

Setup local commands.

```
\@glsxtr@unsrt@gloss@init
```

A loop within the tabular-like styles can cause problems, so move the loop outside. The entire glossary will be saved in `\@glsxtr@doglossary`, which will be built up in the loop. Note that v1.50 has removed `\glsresetentrylist`.

```
\def\@glsxtr@doglossary{%
  \begin{theglossary}%
  \glossaryheader
}%
```

Apply the post-begin hook.

```
\printunsrtglossarypostbegin{\@glsxtr@doglossary}%
```

Iterate over all entries in the current glossary and add the relevant commands to `\@glsxtr@doglossary`.

```
\expandafter\@for\expandafter\glscurrententrylabel\expandafter
  :\expandafter=\csname glolist@\@glo@type\endcsname\do{%
  \ifdefempty{\glscurrententrylabel}
  {}%
  {%
```

Initialise hooks

```
      \@gls@xtr@initprocess
```

Process this entry (unless it has been skipped).

```
      \glsxtr@process
      {%
        \ifglsxtr@printgloss@groups
```

Check if the group heading should be added and, if so, add it. `\@glsxtr@groupheading` will be empty if no group heading.

```
        \glsxtraddgroup\glscurrententrylabel
        {%
          \@glsxtr@checkgroup\glscurrententrylabel
          \expandafter\appto\expandafter\@glsxtr@doglossary\expandafter
            {\@glsxtr@groupheading}%
        }%
        \fi
```

Apply the pre-entry hook.

```
        \printunsrtglossarypreentryprocesshook{\@glsxtr@doglossary}%

        \protected@eappto\@glsxtr@doglossary{%
          \noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}%
```

Apply the post-entry hook.

```
        \printunsrtglossarypostentryprocesshook{\@glsxtr@doglossary}%
      }%
    }%
}%
```

Apply the pre-end hook.

```
    \printunsrtglossarypreend{\@glsxtr@doglossary}%
    \appto\@glsxtr@doglossary{\end{theglossary}}%
    \printunsrtglossarypredoglossary
    \@glsxtr@doglossary
  }%
  \glossarypostamble
}
```

\@glsxtr@unsrt@gloss@init   Initialise hooks needed at the start.

```
 \newcommand*{\@glsxtr@unsrt@gloss@init}{%
```

Determine how to obtain the group information.

```
 \key@ifundefined{glossentry}{group}%
 {\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle}%
 {\let\@gls@getgrouptitle\@glsxtr@unsrt@getgrouptitle}%
```

Initialise current group information.

```
 \def\@gls@currentlettergroup{}%
```

Need to keep track of the current group hierarchical level

```
 \def\@gls@currentlettergroup@level{-1}%
```

and the current entry hierarchical level.

```
 \def\glscurrententrylevel{-1}%
```

Initialise the root entry. This will be the most recent entry that doesn't have a parent.

```
 \def\glscurrentrootentry{}%
```

Initialise the top-level entry. This will be the most recent entry that had level=0 (after adjustment).

```
 \def\glscurrenttoplevelentry{}%
}
```

\@gls@xtr@initprocess   Initialise hooks needed for each iteration of the process loop.

```
 \newcommand*{\@gls@xtr@initprocess}{%
```

Save the current hierarchical level (adjusted).

```
 \ifglsxtrprintglossflatten
 \edef\glscurrententrylevel{\number\@glsxtr@leveloffset}%
 \else
 \edef\glscurrententrylevel{%
   \number\numexpr\csname glo@\glscurrententrylabel @level\endcsname
     + \@glsxtr@leveloffset}%
 \fi
```

If this level 0, update \glscurrenttoplevelentry

```
 \ifnum\glscurrententrylevel=0\relax
   \let\glscurrenttoplevelentry\glscurrententrylabel
 \fi
```

216

If this entry doesn't have a parent, update `\glscurrentrootentry`

```
\ifglsxtrprintglossflatten
  \let\glscurrentrootentry\glscurrententrylabel
\else
  \ifglshasparent{\glscurrententrylabel}{}%
  {\let\glscurrentrootentry\glscurrententrylabel}%
\fi
```

Initialise to do the current entry.

```
\let\glsxtr@process\@firstofone
```

Provide a way to skip the current entry. This will redefine `\glsxtr@process` to ignore its argument.

```
\let\printunsrtglossaryskipentry\@glsxtr@printunsrtglossaryskipentry
\printunsrtglossaryentryprocesshook{\glscurrententrylabel}%
}
```

`\printunsrtinnerglossary` Similar to `\printunsrtglossary` but doesn't add the section heading, preamble, postamble or start and end of theglossary. Grouping is automatically applied so it may cause a problem within tabular-like environments. The beginning and ending of theglossary should be added around this command (but ensure the style has been set first). The simplest way of doing this is to place `\printunsrtinnerglossary` inside the printunsrtglossarywrap environment.

```
\newcommand*{\printunsrtinnerglossary}[3][]{%
  \begingroup
  \def\@glsxtr@printglossopts{#1}%
  \def\@glo@type{\glsdefaulttype}%
  \setkeys{printgloss}[title,toctitle,style,numberedsection,sort,label]{#1}%
  \let\currentglossary\@glo@type
  #2%
  \@print@unsrt@innerglossary
  #3%
  \endgroup
}
```

printunsrtglossarywrap (*env.*)

```
\newenvironment{printunsrtglossarywrap}[1][]%
{%
  \def\@glsxtr@printglossopts{#1}%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glotype@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '\@glo@type'. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
```

```
        style key=value option}%
    \fi
  }%
  \def\gls@dotoctitle{\glssettoctitle{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
```

The type key simply allows the title to be set if the title key isn't supplied.

```
  \ifglossaryexists*{\@glo@type}%
  {%
   \ifx\glossarytitle\org@glossarytitle
   \else
     \expandafter\let\csname @glotype@\@glo@type @title\endcsname
                     \glossarytitle
   \fi
   \let\currentglossary\@glo@type
  }%
  {}%
  \let\org@glossaryentrynumbers\glossaryentrynumbers
  \let\glsnonextpages\@glsnonextpages
  \let\glsnextpages\@glsnextpages
  \let\nopostdesc\@nopostdesc
  \gls@dotoctitle
  \@glossarystyle
  \let\gls@org@glossaryentryfield\glossentry
  \let\gls@org@glossarysubentryfield\subglossentry

  \renewcommand{\glossentry}[1]{%
    \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
    \gls@org@glossaryentryfield{##1}%
  }%
  \renewcommand{\subglossentry}[2]{%
    \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
    \gls@org@glossarysubentryfield{##1}{##2}%
  }%
  \@gls@preglossaryhook
  \glossarysection[\glossarytoctitle]{\glossarytitle}%
  \glossarypreamble
  \begin{theglossary}%
  \glossaryheader
  \glsresetentrylist
}%
{%
  \end{theglossary}%
  \glossarypostamble
  \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
  \global\let\warn@noprintglossary\relax
}
```

\@print@unsrt@innerglossary This is much like `\@print@unsrt@innerglossary` but only contains what would normally be the content of the theglossary.

```
\newcommand*{\@print@unsrt@innerglossary}{%
```

No section header or preamble.

```
\glsxtrifemptyglossary{\@glo@type}%
{%
  \GlossariesExtraWarning{No entries defined in glossary '\@glo@type'}%
}%
{%
```

Setup local commands.

```
\@glsxtr@unsrt@gloss@init
```

No header or reset.

```
\def\@glsxtr@doglossary{}%
```

Iterate over all entries in the current glossary and add the relevant commands to `\@glsxtr@doglossary`.

```
\expandafter\@for\expandafter\glscurrententrylabel\expandafter
  :\expandafter=\csname glolist@\@glo@type\endcsname\do{%
\ifdefempty{\glscurrententrylabel}
{}%
{%
```

Initialise hooks

```
\@gls@xtr@initprocess
```

Process this entry (unless it has been skipped).

```
\glsxtr@process
{%
  \ifglsxtr@printgloss@groups
```

Check if the group heading should be added and, if so, add it. `\@glsxtr@groupheading` will be empty if no group heading.

```
\glsxtraddgroup\glscurrententrylabel
{%
  \@glsxtr@checkgroup\glscurrententrylabel
  \expandafter\appto\expandafter\@glsxtr@doglossary\expandafter
    {\@glsxtr@groupheading}%
}%
\fi
```

Apply the pre-entry hook.

```
\printunsrtglossarypreentryprocesshook{\@glsxtr@doglossary}%

\protected@eappto\@glsxtr@doglossary{%
  \noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}%
```

Apply the post-entry hook.

```
    \printunsrtglossarypostentryprocesshook{\@glsxtr@doglossary}%
  }%
 }%
}%
```

`\printunsrtglossarypreend` not used.

```
      \printunsrtglossarypredoglossary
      \@glsxtr@doglossary
   }%
```

No postamble.

```
 }
```

`\glsxtraddgroup` Now that `bib2gls` v3.0+ has the ability to store group labels for sub-levels, provide a way to allow for this. This checks if the entry has a parent, which was used originally, unless the flatten option has been used. `bib2gls` will redefine this in the .glstex file if the group-level setting is used.

```
\newcommand*{\glsxtraddgroup}[2]{%
  \ifglsxtrprintglossflatten
    #2%
  \else
    \ifglshasparent{#1}{}{#2}%
  \fi
}
```

`\printunsrtglossaryentryprocesshook`

```
\newcommand*{\printunsrtglossaryentryprocesshook}[1]{}
```

`\printunsrtglossarypreentryprocesshook` This hook is performed before the entry line has been added to `\@glsxtr@do@glossary`. The argument will be `\@glsxtr@do@glossary` so that content can be appended to it. The current entry can be referenced with `\glscurrententrylabel`. The current level can be referenced with `\glscurrententrylevel`, etc.

```
\newcommand*{\printunsrtglossarypreentryprocesshook}[1]{}
```

`\printunsrtglossarypostentryprocesshook` This hook is performed after the entry line has been added to `\@glsxtr@do@glossary`. The argument will be `\@glsxtr@do@glossary` so that content can be appended to it. The current entry can be referenced with `\glscurrententrylabel`. The current level can be referenced with `\glscurrententrylevel`, etc.

```
\newcommand*{\printunsrtglossarypostentryprocesshook}[1]{}
```

`\printunsrtglossarygrouphook` Similar hook used when the group heading added. In this case the argument will be `\@glsxtr@groupheading`.

```
\newcommand*{\printunsrtglossarygrouphook}[1]{}
```

`\printunsrtglossaryskipentry`

```
\newcommand*{\printunsrtglossaryskipentry}{%
  \PackageError{glossaries-extra}{\string\printunsrtglossaryskipentry\space
can only be used within \string\printunsrtglossaryentryprocesshook}{}%
}
```

`\@glsxtr@printunsrtglossaryskipentry`

```
\newcommand*{\@glsxtr@printunsrtglossaryskipentry}{%
  \let\glsxtr@process\@gobble
}
```

```
\newcommand*{\printunsrtglossarypredoglossary}{}
```

\printunsrtglossarypreend

```
\newcommand*{\printunsrtglossarypreend}[1]{}
```

\printunsrtglossarypostbegin

```
\newcommand*{\printunsrtglossarypostbegin}[1]{}
```

\@printunsrt@glossary@handler

```
\newcommand{\@printunsrt@glossary@handler}[1]{%
  \protected@xdef\glscurrententrylabel{#1}%
  \printunsrtglossaryhandler\glscurrententrylabel
}
```

\printunsrtglossaryhandler

```
\newcommand{\printunsrtglossaryhandler}[1]{%
  \glsxtrunsrtdo{#1}%
}
```

$\glsxtriflabelinlist\{\langle label \rangle\}\{\langle list \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}$

\glsxtriflabelinlist

Might be useful for the handler to check if an entry label or category label is contained in a list, so provide a user-level version of \@gls@ifinlist which ensures the label and list are fully expanded.

```
\newrobustcmd*{\glsxtriflabelinlist}[4]{%
  \protected@edef\@glsxtr@doiflabelinlist{\noexpand\@gls@ifinlist{#1}{#2}}%
  \@glsxtr@doiflabelinlist{#3}{#4}%
}
```

\print@op@unsrtglossaryunit

```
\newcommand{\print@op@unsrtglossaryunit}[2][]{%
  \s@printunsrtglossary[type=\glsdefaulttype,#1]{%
    \printunsrtglossaryunitsetup{#2}%
  }%
}
```

\printunsrtglossaryunitsetup

```
\newcommand*{\printunsrtglossaryunitsetup}[1]{%
  \renewcommand{\printunsrtglossaryhandler}[1]{%
    \glsxtrfieldxifinlist{##1}{record.#1}{\csuse{the#1}}
    {\glsxtrunsrtdo{##1}}%
    {}%
  }%
```

Only the target names should have the prefixes adjusted as `\gls` etc need the
original `\glolinkprefix`. The `\@gobble` part discards `\glolinkprefix`.

```
  \ifcsundef{theH#1}%
  {%
    \renewcommand*{\@glsxtrhypernameprefix}{record.#1.\csuse{the#1}.\@gobble}%
  }%
  {%
    \renewcommand*{\@glsxtrhypernameprefix}{record.#1.\csuse{theH#1}.\@gobble}%
  }%
  \renewcommand*{\glossarysection}[2][]{}%
  \appto\glossarypostamble{\printunsrtglossaryunitpostskip}%
}
```

ntunsrtglossaryunitpostskip

```
\newcommand*{\printunsrtglossaryunitpostskip}{\glspar\medskip\glspar}
```

rint@noop@unsrtglossaryunit

```
\newcommand{\print@noop@unsrtglossaryunit}[2][]{%
  \PackageError{glossaries-extra}{\string\printunsrtglossaryunit\space
   requires the record=only or record=alsoindex package option}{}%
}
```

\@glsxtr@unsrt@getgrouptitle

```
\newrobustcmd*{\@glsxtr@unsrt@getgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
  \@onelevel@sanitize\@glsxtr@titlelabel
  \ifcsdef{\@glsxtr@titlelabel}
  {\letcs{#2}{\@glsxtr@titlelabel}}%
  {\def#2{#1}}%
}
```

\glsxtrunsrtdo Provide a user-level call to `\@glsxtr@noidx@do` to make it easier to define a
new handler.

```
\newcommand{\glsxtrunsrtdo}{\@glsxtr@noidx@do}
```

\glsxtrgroupfield bib2gls provides a supplementary field labelled `secondarygroup` for secondary
glossaries, so provide a way of switching to that field. (The `group` key still needs
checking. There's no associated key with the internal field).

```
\newcommand*{\glsxtrgroupfield}{group}
```

The `tabular`-like glossary styles cause quite a problem with the iterative ap-
proach. In particular for the group skip. To compensate for this, the groups are
now determined while `\@glsxtr@doglossary` is being constructed rather than
in the handler.

\@glsxtr@checkgroup The argument is the entry's label. (This block of code was formerly in
`\@glsxtr@noidx@do`.) Now that this is no longer within a tabular envi-
ronment, the global definitions aren't needed. The result is now stored in

222

`\@glsxtr@groupheading`, which will be empty if no heading is required. The
current hierarchical level must have first been saved to `\glscurrententrylevel`.

```
\newcommand*{\@glsxtr@checkgroup}[1]{%
  \def\@glsxtr@groupheading{}%
  \key@ifundefined{glossentry}{group}%
  {%
    \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
  }%
  {%

    \protected@edef\@glo@thislettergrp{%
        \csuse{glo@\glsdetoklabel{#1}@\glsxtrgroupfield}}%
  }%
```

Need to keep track of the current group for the current level.

```
  \ifcsundef{@gls@currentlettergroup\romannumeral\glscurrententrylevel}%
  {\csdef{@gls@currentlettergroup\romannumeral\glscurrententrylevel}{}}{}%
```

Has the group label changed for the current level?

```
  \ifcsequal{@glo@thislettergrp}{@gls@currentlettergroup\romannumeral\glscurrententrylevel}%
  {}%
  {%
    \ifdefempty\@glo@thislettergrp
    {}%
    {%
```

Check the hierarchical level.

```
      \ifnum\glscurrententrylevel>0\relax
        \protected@eappto\@glsxtr@groupheading{%
          \noexpand\glssubgroupheading
          {\@gls@currentlettergroup@level}{\glscurrententrylevel}%
          {\csuse{glo@\glsdetoklabel{#1}@parent}}%
          {\expandonce\@glo@thislettergrp}%
        }%
      \else
        \ifdefempty{\@gls@currentlettergroup}{}%
        {%
```

Don't add `\glsgroupskip` if nogroupskip setting is on.

```
          \ifglsnogroupskip
          \else
            \def\@glsxtr@groupheading{\glsgroupskip}%
          \fi
        }%
        \protected@eappto\@glsxtr@groupheading{%
          \noexpand\glsgroupheading{\expandonce\@glo@thislettergrp}%
        }%
      \fi
      \let\@gls@currentlettergroup@level\glscurrententrylevel
      \cslet{@gls@currentlettergroup\romannumeral\glscurrententrylevel}\@glo@thislettergrp
```

Perform the group hook, which can be used to add content.

```
        \printunsrtglossarygrouphook{\@glsxtr@groupheading}%
      }%
    }%
 }
```

> \glssubgroupheading{⟨*previous level*⟩}{⟨*level*⟩}{⟨*parent*⟩}
> {⟨*group label*⟩}

Default definition uses the same format as the top-level heading. Note that this won't include the group skip.

```
\newcommand*{\glssubgroupheading}[4]{\glsgroupheading{#4}}
```

Stores the internal name of the location field.

```
\newcommand*{\GlsXtrLocationField}{location}
```

Minor modification of \@gls@noidx@do to check for location field if present, but also need to check for the group field and flatten option.

```
\newcommand{\@glsxtr@noidx@do}[1]{%
  \ifglsentryexists{#1}%
  {%
    \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
    \global\letcs{\@gls@location}{glo@\glsdetoklabel{#1}@\GlsXtrLocationField}%
```

Use level number to determine whether or not this entry has a parent.

```
    \ifglsxtrprintglossflatten
      \gls@level=\@glsxtr@leveloffset\relax
    \else
      \gls@level=\numexpr\csuse{glo@\glsdetoklabel{#1}@level}+\@glsxtr@leveloffset\relax
    \fi
    \ifnum\gls@level>0
      \let\@glsxtr@ifischild\@firstoftwo
    \else
      \let\@glsxtr@ifischild\@secondoftwo
    \fi
```

Some glossary styles (such as topicmcols) save the level using \def so make sure \gls@level is expanded before being passed to \subglossentry.

```
    \@glsxtr@ifischild
    {%
      \ifdefvoid{\@gls@location}%
      {%
```

If \GlsXtrLocationField has been changed then don't fallback on loclist.

```
        \ifdefstring{\GlsXtrLocationField}{location}%
        {%
          \ifdefvoid{\@gls@loclist}%
          {%
```

224

```
            \expandafter\subglossentry\expandafter{\number\gls@level}{#1}{}%
          }%
          {%
            \expandafter\subglossentry\expandafter{\number\gls@level}{#1}%
            {%
              \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
            }%
          }%
        }%
        {%
          \expandafter\subglossentry\expandafter{\number\gls@level}{#1}{}%
        }%
      }%
      {%
        \expandafter\subglossentry\expandafter
          {\number\gls@level}{#1}{\glossaryentrynumbers{\@gls@location}}%
      }%
    }%
    {%
      \ifdefvoid{\@gls@location}%
      {%
```

If `\GlsXtrLocationField` has been changed then don't fallback on loclist.

```
        \ifdefstring{\GlsXtrLocationField}{location}%
        {%
          \ifdefvoid{\@gls@loclist}
          {%
            \glossentry{#1}{}%
          }%
          {%
            \glossentry{#1}%
            {%
              \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
            }%
          }%
        }%
        {%
          \glossentry{#1}{}%
        }%
      }%
      {%
        \glossentry{#1}%
        {%
          \glossaryentrynumbers{\@gls@location}%
        }%
      }%
    }%
  }%
  {}%
}
```

Provide a way to conveniently define commands that behaves like `\gls` with a label prefix.

It's possible that the user might want minor variations with the same prefix but different default options, so use a counter to provide unique inner commands.

\glsxtrnewgls

```
\newcount\@glsxtrnewgls@inner
```

(The default options supplied in ⟨*options*⟩ below could possibly be used to form the inner control sequence name to help make it unique, but it might feasibly contain thevalue where the value might contain commands.)

\glsxtrdoidentify

```
\newcommand*{\glsxtrdoidentify}[1]{%
 \ifdefequal\@glsxtr@record@setting\@glsxtr@record@setting@off{}{#1}%
}
```

\@glsxtr@providenewgls

```
\newcommand*{\@glsxtr@providenewgls}{%
  \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@newglslike}[2]{}}%
  \let\@glsxtr@providenewgls\relax
}
```

\glsxtridentifyglslike Identify the command given in the second argument for the benefit of bib2gls and also identify command as a blocker for `\makefirstuc`.

```
\newcommand{\glsxtridentifyglslike}[2]{%
 \glsmfublocker{#2}%
 \glsxtrdoidentify
 {%
   \@glsxtr@providenewgls
   \protected@write\@auxout{}{\string\@glsxtr@newglslike{#1}{\string#2}}%
 }%
}
```

\@glsxtr@providenewglsfamily

```
\newcommand*{\@glsxtr@providenewglsfamily}{%
  \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@newglslikefamily}[8]{}}%
  \let\@glsxtr@providenewglsfamily\relax
}
```

> \glsxtridentifyglsfamily{⟨*options*⟩}{⟨*prefix*⟩}{⟨*gls*⟩}
> {⟨*glspl*⟩}{⟨*Gls*⟩}{⟨*Glspl*⟩}{⟨*GLS*⟩}{⟨*GLSpl*⟩}

\glsxtridentifyglsfamily

Identify the family of commands for the benefit of bib2gls and also establishes a sentence-case mapping.

```
\newcommand{\glsxtridentifyglsfamily}[8]{%
 \glsmfuaddmap{#3}{#5}%
```

226

```
    \glsmfuaddmap{#4}{#6}%
    \glsmfublocker{#7}%
    \glsmfublocker{#8}%
    \glsxtrdoidentify
    {%
      \@glsxtr@providenewglsfamily
      \protected@write\@auxout{}{\string\@glsxtr@newglslikefamily{\detokenize{#1}}{\detokenize{#2}}{\det
    }%
}
```

\@glsxtr@providenewglslink

```
\newcommand*{\@glsxtr@providenewglslink}{%
  \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@newglslink}[2]{}}%
  \let\@glsxtr@providenewglslink\relax
}
```

\glsxtridentifyglslink Identify the command given in the second argument for the benefit of `bib2gls` and identify the command as a blocker for `\makefirstuc`.

```
\newcommand{\glsxtridentifyglslink}[2]{%
\glsmfublocker{#2}%
\glsxtrdoidentify
{%
  \@glsxtr@providenewglslink
  \protected@write\@auxout{}{\string\@glsxtr@newglslink{#1}{\string#2}}%
}%
}
```

> \@glsxtrnewglslink[⟨*options*⟩]{⟨*prefix*⟩}{⟨*cs*⟩}{⟨*inner cs name*⟩}

\@glsxtrnewglslink

```
\newcommand*{\@glsxtrnewglslink}[4]{%
  \ifdef{#3}%
  {%
    \PackageError{glossaries-extra}{Command \string#3\space already
defined}{}%
  }%
  {%
```

Write information to the aux file for bib2gls.

```
    \glsxtridentifyglslink{#2}{#3}%
    \ifcsdef{@#4link@#2}%
    {%
      \advance\@glsxtrnewgls@inner by \@ne
      \def\@glsxtrnewgls@innercsname{@#4link\number\@glsxtrnewgls@inner @#2}%
    }%
    {\def\@glsxtrnewgls@innercsname{@#4link@#2}}%
    \expandafter\newrobustcmd\expandafter*\expandafter
     #3\expandafter{\expandafter\@gls@hyp@opt\csname\@glsxtrnewgls@innercsname\endcsname}%
```

227

```
      \ifstrempty{#1}%
      {%
        \expandafter\newcommand\expandafter*\csname\@glsxtrnewgls@innercsname\endcsname[2][]{%
          \csname #4\endcsname{##1}{#2##2}%
        }%
      }%
      {%
        \expandafter\newcommand\expandafter*\csname\@glsxtrnewgls@innercsname\endcsname[2][]{%
          \csname #4\endcsname{#1,##1}{#2##2}%
        }%
      }%
    }%
  }
```

\glsxtrnewglslink

> \glsxtrnewglslink[⟨*options*⟩]{⟨*prefix*⟩}{⟨*cs*⟩}

The first argument prepends to the options and the second argument is the prefix.

```
\newrobustcmd*{\glsxtrnewglslink}[3][]{%
  \@glsxtrnewglslink{#1}{#2}{#3}{@gls@@link}%
}
```

\glsxtrnewglsdisp

> \glsxtrnewglsdisp[⟨*options*⟩]{⟨*prefix*⟩}{⟨*cs*⟩}

The first argument prepends to the options and the second argument is the prefix.

```
\newrobustcmd*{\glsxtrnewglsdisp}[3][]{%
  \@glsxtrnewglslink{#1}{#2}{#3}{@glsdisp}%
}
```

\@glsxtrnewgls

> \@glsxtrnewgls[⟨*options*⟩]{⟨*prefix*⟩}{⟨*cs*⟩}{⟨*inner cs name*⟩}

```
\newcommand*{\@glsxtrnewgls}[4]{%
  \ifdef{#3}%
  {%
    \PackageError{glossaries-extra}{Command \string#3\space already
defined}{}%
  }%
  {%
```

Write information to the aux file for bib2gls.

```
    \glsxtridentifyglslike{#2}{#3}%
    \ifcsdef{@#4like@#2}%
```

228

```
      {%
        \advance\@glsxtrnewgls@inner by \@ne
        \def\@glsxtrnewgls@innercsname{@#4like\number\@glsxtrnewgls@inner @#2}%
      }%
      {\def\@glsxtrnewgls@innercsname{@#4like@#2}}%
      \expandafter\newrobustcmd\expandafter*\expandafter
       #3\expandafter{\expandafter\@gls@hyp@opt\csname\@glsxtrnewgls@innercsname\endcsname}%
      \ifstrempty{#1}%
      {%
        \expandafter\newcommand\expandafter*\csname\@glsxtrnewgls@innercsname\endcsname[2][]{%
          \new@ifnextchar[%
          {\csname @#4@\endcsname{##1}{#2##2}}%
          {\csname @#4@\endcsname{##1}{#2##2}[]}%
        }%
      }%
      {%
        \expandafter\newcommand\expandafter*\csname\@glsxtrnewgls@innercsname\endcsname[2][]{%
          \new@ifnextchar[%
          {\csname @#4@\endcsname{#1,##1}{#2##2}}%
          {\csname @#4@\endcsname{#1,##1}{#2##2}[]}%
        }%
      }%
    }%
  }
```

\glsxtrnewgls	$\boxed{\texttt{\textbackslash glsxtrnewgls[}\langle\textit{options}\rangle\texttt{]\{}\langle\textit{prefix}\rangle\texttt{\}\{}\langle\textit{cs}\rangle\texttt{\}}}$

The first argument prepends to the options and the second argument is the
prefix.

```
\newrobustcmd*{\glsxtrnewgls}[3][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{gls}%
}
```

\glsxtrnewglslike Provide a way to conveniently define commands that behave like \gls, \glspl,
\Gls and \Glspl with a label prefix. The first argument prepends to the options
and the second argument is the prefix.

```
\newrobustcmd*{\glsxtrnewglslike}[6][]{%
  \glsxtridentifyglsfamily{#1}{#2}{#3}{#4}{#5}{#6}{}{}%
  \@glsxtrnewgls{#1}{#2}{#3}{gls}%
  \@glsxtrnewgls{#1}{#2}{#4}{glspl}%
  \@glsxtrnewgls{#1}{#2}{#5}{Gls}%
  \@glsxtrnewgls{#1}{#2}{#6}{Glspl}%
}
```

\glsxtrnewGLSlike Provide a way to conveniently define commands that behave like \GLS, \GLSpl
with a label prefix. The first argument prepends to the options and the second
argument is the prefix.

```
\newrobustcmd*{\glsxtrnewGLSlike}[4][]{%
  \glsxtridentifyglsfamily{#1}{#2}{}{}{}{}{#3}{#4}%
  \@glsxtrnewgls{#1}{#2}{#3}{GLS}%
  \@glsxtrnewgls{#1}{#2}{#4}{GLSpl}%
}
```

\glsxtrnewrgls   As \glsxtrnewgls but for \rgls.

```
\newrobustcmd*{\glsxtrnewrgls}[3][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{rgls}%
}
```

\glsxtrnewrglslike   As \glsxtrnewglslike but for \rgls etc.

```
\newrobustcmd*{\glsxtrnewrglslike}[6][]{%
  \glsxtridentifyglsfamily{#1}{#2}{#3}{#4}{#5}{#6}{}{}%
  \@glsxtrnewgls{#1}{#2}{#3}{rgls}%
  \@glsxtrnewgls{#1}{#2}{#4}{rglspl}%
  \@glsxtrnewgls{#1}{#2}{#5}{rGls}%
  \@glsxtrnewgls{#1}{#2}{#6}{rGlspl}%
}
```

\glsxtrnewrGLSlike   As \glsxtrnewGLSlike but for \rGLS etc.

```
\newrobustcmd*{\glsxtrnewrGLSlike}[4][]{%
  \glsxtridentifyglsfamily{#1}{#2}{}{}{}{}{#3}{#4}%
  \@glsxtrnewgls{#1}{#2}{#3}{rGLS}%
  \@glsxtrnewgls{#1}{#2}{#4}{rGLSpl}%
}
```

Provide easy access to record count fields.

\GlsXtrTotalRecordCount   Access total record count. This is designed to be expandable. The argument is the label.

```
\newcommand*{\GlsXtrTotalRecordCount}[1]{%
\ifcsdef{glo@\glsdetoklabel{#1}@recordcount}%
{\csname glo@\glsdetoklabel{#1}@recordcount\endcsname}%
{0}%
}
```

\GlsXtrRecordCount   Access record count for a particular counter. The first argument is the label. The second argument is the counter name.

```
\newcommand*{\GlsXtrRecordCount}[2]{%
\ifcsdef{glo@\glsdetoklabel{#1}@recordcount.#2}%
{\csname glo@\glsdetoklabel{#1}@recordcount.#2\endcsname}%
{0}%
}
```

\GlsXtrLocationRecordCount   Access record count for a particular counter and location. The first argument is the label. The second argument is the counter name. The third argument is the location. This command shouldn't be used if the location doesn't fully expand unless \glsxtrdetoklocation can be set to something sensible.

```
\newcommand*{\GlsXtrLocationRecordCount}[3]{%
 \ifcsdef{glo@\glsdetoklabel{#1}@recordcount.#2.\glsxtrdetoklocation{#3}}%
 {\csname glo@\glsdetoklabel{#1}@recordcount.#2.\glsxtrdetoklocation{#3}\endcsname}%
 {0}%
}
```

\glsxtrdetoklocation

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

\glsxtrenablerecordcount

```
\newcommand*{\glsxtrenablerecordcount}{%
  \renewcommand*{\gls}{\rgls}%
  \renewcommand*{\Gls}{\rGls}%
  \renewcommand*{\glspl}{\rglspl}%
  \renewcommand*{\Glspl}{\rGlspl}%
  \renewcommand*{\GLS}{\rGLS}%
  \renewcommand*{\GLSpl}{\rGLSpl}%
  \renewcommand{\shortcut@gls}{\rgls}%
  \renewcommand{\shortcut@glspl}{\rglspl}%
  \renewcommand{\shortcut@Gls}{\rGls}%
  \renewcommand{\shortcut@Glspl}{\rGlspl}%
  \renewcommand{\shortcut@GLS}{\rGLS}%
  \renewcommand{\shortcut@GLSpl}{\rGLSpl}%
}
```

\glsxtrrecordtriggervalue The value used by the record trigger test. The argument is the entry's label.

```
\newcommand*{\glsxtrrecordtriggervalue}[1]{%
 \GlsXtrTotalRecordCount{#1}%
}
```

\GlsXtrSetRecordCountAttribute

```
\newcommand*{\GlsXtrSetRecordCountAttribute}[2]{%
 \@for\@glsxtr@cat:=#1\do
 {%
   \ifdefempty{\@glsxtr@cat}{}%
   {%
     \glssetcategoryattribute{\@glsxtr@cat}{recordcount}{#2}%
   }%
 }%
}
```

$\boxed{\text{\glsxtrifrecordtrigger}\{\langle label\rangle\}\{\langle trigger\ format\rangle\}\{\langle normal\rangle\}}$

\glsxtrifrecordtrigger

```
\newcommand*{\glsxtrifrecordtrigger}[3]{%
 \glshasattribute{#1}{recordcount}%
 {%
   \ifnum\glsxtrrecordtriggervalue{#1}>\glsgetattribute{#1}{recordcount}\relax
```

```
       #3%
      \else
       #2%
      \fi
    }%
    {#3}%
  }
```

\@glsxtr@rglstrigger@record Still need a record to ensure that `bib2gls` selects the entry.

```
\newcommand*{\@glsxtr@rglstrigger@record}[3]{%
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \let\@gls@link@label\glslabel
  \def\@glsxtr@thevalue{}%
  \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
  \def\@glsnumberformat{glstriggerrecordformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \def\@glsxtr@thevalue{}%
  \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
```

Save local setting.

```
  \@gls@save@glslocal
```

Initialise preunset, prereset and postunset

```
  \glsinitreunsets
  \glsxtrinitwrgloss
  \glslinkpresetkeys
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \ifdefempty{\@glsxtr@thevalue}%
  {%
    \@gls@saveentrycounter
  }%
  {%
    \let\theglsentrycounter\@glsxtr@thevalue
    \def\theHglsentrycounter{\@glsxtr@theHvalue}%
  }%
  \glslinkwrcontent
  {%
    \ifglsxtrinitwrglossbefore
      \glsxtr@wrglossary@encap{#2}{\@do@wrglossary{#2}}%
    \fi
    #3%
    \ifglsxtrinitwrglossbefore
    \else
      \glsxtr@wrglossary@encap{#2}{\@do@wrglossary{#2}}%
    \fi
  }%
  \@gls@restore@glslocal
  \@gls@do@glsunset{#2}%
}
```

232

\glstriggerrecordformat Typically won't be used as it should be recognised as a special type of ignored location by `bib2gls`.

```
\newcommand*{\glstriggerrecordformat}[1]{}
```

\rgls

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

\@rgls

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

\@rgls@

```
\def\@rgls@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rglsformat{#2}{#3}}%
  }%
  {%
    \@gls@{#1}{#2}[#3]%
  }%
}%
```

\rglspl

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

\@rglspl

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

\@rglspl@

```
\def\@rglspl@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rglsplformat{#2}{#3}}%
  }%
  {%
    \@glspl@{#1}{#2}[#3]%
  }%
}%
```

\rGls

```
\newrobustcmd*{\rGls}{\@gls@hyp@opt\@rGls}
\glsmfuaddmap{\rgls}{\rGls}
```

\@rGls

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

\@rGls@

```
\def\@rGls@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGlsformat{#2}{#3}}%
  }%
  {%
    \@Gls@{#1}{#2}[#3]%
  }%
}%
```

\rGlspl

```
\newrobustcmd*{\rGlspl}{\@gls@hyp@opt\@rGlspl}
\glsmfuaddmap{\rglspl}{\rGlspl}
```

\@rGlspl

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

\@rGlspl@

```
\def\@rGlspl@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGlsplformat{#2}{#3}}%
  }%
  {%
    \@Glspl@{#1}{#2}[#3]%
  }%
}%
```

\rGLS

```
\newrobustcmd*{\rGLS}{\@gls@hyp@opt\@rGLS}
\glsmfublocker{\rGLS}%
```

\@rGLS

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

\@rGLS@

```
\def\@rGLS@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGLSformat{#2}{#3}}%
  }%
  {%
    \@GLS@{#1}{#2}[#3]%
  }%
}%
```

\rGLSpl

```
\newrobustcmd*{\rGLSpl}{\@gls@hyp@opt\@rGLSpl}
\glsmfublocker{\rGLSpl}%
```

\@rGLSpl

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

\@rGLSpl@

```
\def\@rGLSpl@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGLSplformat{#2}{#3}}%
  }%
  {%
    \@GLSpl@{#1}{#2}[#3]%
  }%
}%
```

\rglsformat

```
\newcommand*{\rglsformat}[2]{%
  \glsifregular{#1}
  {\glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}}#2%
}
```

\rglsplformat

```
\newcommand*{\rglsplformat}[2]{%
  \glsifregular{#1}
  {\glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}}#2%
}
```

\rGlsformat

```
\newcommand*{\rGlsformat}[2]{%
  \glsifregular{#1}
  {\Glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}}#2%
}
```

\rGlsplformat

```
\newcommand*{\rGlsplformat}[2]{%
  \glsifregular{#1}
  {\Glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}}#2%
}
```

\rGLSformat

```
\newcommand*{\rGLSformat}[2]{%
 \expandafter\glsuppercase\expandafter{\rglsformat{#1}{#2}}%
}
```

\rGLSplformat

```
\newcommand*{\rGLSplformat}[2]{%
 \expandafter\glsuppercase\expandafter{\rglsplformat{#1}{#2}}%
}
```

1.4   Link Counting

This is different to the entry counting provided by the base package (which counts the number of times the first use flag is unset). Instead, this method hooks into \@gls@link (through \glsxtr@inc@linkcount) to increment an associated counter. To preserve resources, the counter is only defined if it needs to be incremented. This method is independent of the presence of hyperlinks. (The "link" part of the name refers to \@gls@link not \hyperlink.)

\@glsxtr@do@inc@linkcount This performs the actual incrementing and counter definition. The counter is given by \c@glsxtr@linkcount@⟨*label*⟩ where ⟨*label*⟩ is the entry's label. Since this is performed within \@gls@link the label can be accessed with \glslabel.

```
\newcommand{\@glsxtr@do@inc@linkcount}{%
```

Does this entry have the linkcount attribute set?

```
\glsifattribute{\glslabel}{linkcount}{true}%
{%
```

Does the counter exist?

```
    \ifcsdef{c@glsxtr@linkcount@\glslabel}{}%
    {%
```

Counter doesn't exist, so define it.

```
        \newcounter{glsxtr@linkcount@\glslabel}%
```

If linkcountmaster is set, add to counter reset.

```
        \glshasattribute{\glslabel}{linkcountmaster}%
        {%
```

Need to ensure values are fully expanded.

```
            \begingroup
            \edef\@glo@tmp{\endgroup\noexpand\@addtoreset{glsxtr@linkcount@\glslabel}%
             {\glsgetattribute{\glslabel}{linkcountmaster}}}%
            \@glo@tmp
        }%
        {}%
    }%
```

236

Increment counter:
```
 \glsxtrinclinkcounter{glsxtr@linkcount@\glslabel}%
}%
{}%
}
```

\glsxtrinclinkcounter May be redefined to use \refstepcounter if required.
```
\newcommand*{\glsxtrinclinkcounter}[1]{\stepcounter{#1}}
```

\GlsXtrLinkCounterValue Expands to the associated link counter register or 0 if not defined.
```
\newcommand*{\GlsXtrLinkCounterValue}[1]{%
\ifcsundef{c@glsxtr@linkcount@#1}{0}{\csname c@glsxtr@linkcount@#1\endcsname}%
}
```

\GlsXtrTheLinkCounter Expands to the display value of the associated link counter or 0 if not defined.
```
\newcommand*{\GlsXtrTheLinkCounter}[1]{%
\ifcsundef{theglsxtr@linkcount@#1}{0}%
{\csname theglsxtr@linkcount@#1\endcsname}%
}
```

\GlsXtrIfLinkCounterDef Tests if the counter has been defined
```
\newcommand*{\GlsXtrIfLinkCounterDef}[3]{%
\ifcsundef{theglsxtr@linkcount@#1}{#3}{#2}%
}
```

\GlsXtrLinkCounterName Expands to the associated link counter name. (No check for existence.)
```
\newcommand*{\GlsXtrLinkCounterName}[1]{glsxtr@linkcount@#1}
```

\GlsXtrEnableLinkCounting

> \GlsXtrEnableLinkCounting[⟨*master counter*⟩]{⟨*categories*⟩}

Enable link counting for the given categories.
```
\newcommand*{\GlsXtrEnableLinkCounting}[2][]{%
\let\glsxtr@inc@linkcount\@glsxtr@do@inc@linkcount
\@for\@glsxtr@label:=#2\do
{%
 \glssetcategoryattribute{\@glsxtr@label}{linkcount}{true}%
 \ifstrempty{#1}{}%
 {%
   \ifcsundef{c@#1}%
   {\@nocounterr{#1}}%
   {\glssetcategoryattribute{\@glsxtr@label}{linkcountmaster}{#1}}%
 }%
}%
}
\@onlypreamble\GlsXtrEnableLinkCounting
```

1.5 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.

To allow for formatting commands that need to go inside all other commands (such as the commands provided by soul), also add version of each command that takes a text-block command as an argument.

```
\@ifpackageloaded{glossaries-accsupp}
{
```

Define (or redefine) commands to use the accessibility information.

\glsaccessname Display the name value (no link and no check for existence).

```
\newcommand*{\glsaccessname}[1]{%
  \glsnameaccessdisplay
  {%
    \glsentryname{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtname $\qquad$ **\glsaccessfmtname{⟨insert⟩}{⟨cs⟩}{⟨label⟩}**

```
\newcommand*{\glsaccessfmtname}[3]{%
  \glsnameaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{name}%
  }%
  {#3}%
}
```

\Glsaccessname Display the name value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessname}[1]{%
  \glsnameaccessdisplay
  {%
    \Glsentryname{#1}%
  }%
  {#1}%
}
```

\Glsaccessfmtname $\qquad$ **\Glsaccessfmtname{⟨insert⟩}{⟨cs⟩}{⟨label⟩}**

```
\newcommand*{\Glsaccessfmtname}[3]{%
\glsnameaccessdisplay
{%
  \Glsfmtfield{#1}{#2}{#3}{name}%
}%
{#3}%
}
```

`\GLSaccessname`   Display the name value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccessname}[1]{%
\glsnameaccessdisplay
{%
  \glsuppercase{\glsentryname{#1}}%
}%
{#1}%
}
```

`\GLSaccessfmtname{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\GLSaccessfmtname`

```
\newcommand*{\GLSaccessfmtname}[3]{%
\glsnameaccessdisplay
{%
  \GLSfmtfield{#1}{#2}{#3}{name}%
}%
{#3}%
}
```

`\glsaccesstext`   Display the text value (no link and no check for existence).

```
\newcommand*{\glsaccesstext}[1]{%
\glstextaccessdisplay
{%
  \glsentrytext{#1}%
}%
{#1}%
}
```

`\glsaccessfmttext{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\glsaccessfmttext`

```
\newcommand*{\glsaccessfmttext}[3]{%
\glstextaccessdisplay
{%
  \glsfmtfield{#1}{#2}{#3}{text}%
```

239

```
        }%
        {#3}%
      }
```

\Glsaccesstext   Display the text value (no link and no check for existence) with the first letter
                 converted to upper case.

```
      \newcommand*{\Glsaccesstext}[1]{%
        \glstextaccessdisplay
        {%
          \Glsentrytext{#1}%
        }%
        {#1}%
      }
```

\Glsaccessfmttext{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

\Glsaccessfmttext

```
      \newcommand*{\Glsaccessfmttext}[3]{%
        \glstextaccessdisplay
        {%
          \Glsfmtfield{#1}{#2}{#3}{text}%
        }%
        {#3}%
      }
```

\GLSaccesstext   Display the text value (no link and no check for existence) converted to upper
                 case.

```
      \newcommand*{\GLSaccesstext}[1]{%
        \glstextaccessdisplay
        {%
          \glsuppercase{\glsentrytext{#1}}%
        }%
        {#1}%
      }
```

\GLSaccessfmttext{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

\GLSaccessfmttext

```
      \newcommand*{\GLSaccessfmttext}[3]{%
        \glstextaccessdisplay
        {%
          \GLSfmtfield{#1}{#2}{#3}{text}%
        }%
        {#3}%
      }
```

`\glsaccessplural` Display the plural value (no link and no check for existence).

```
\newcommand*{\glsaccessplural}[1]{%
  \glspluralaccessdisplay
  {%
    \glsentryplural{#1}%
  }%
  {#1}%
}
```

`\glsaccessfmtplural`

$\boxed{\texttt{\textbackslash glsaccessfmtplural\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{cs}\rangle\texttt{\}\{}\langle\mathit{label}\rangle\texttt{\}}}$

```
\newcommand*{\glsaccessfmtplural}[3]{%
  \glspluralaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{plural}%
  }%
  {#3}%
}
```

`\Glsaccessplural` Display the plural value (no link and no check for existence) with the first letter
converted to upper case.

```
\newcommand*{\Glsaccessplural}[1]{%
  \glspluralaccessdisplay
  {%
    \Glsentryplural{#1}%
  }%
  {#1}%
}
```

`\Glsaccessfmtplural`

$\boxed{\texttt{\textbackslash Glsaccessfmtplural\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{cs}\rangle\texttt{\}\{}\langle\mathit{label}\rangle\texttt{\}}}$

```
\newcommand*{\Glsaccessfmtplural}[3]{%
  \glspluralaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{plural}%
  }%
  {#3}%
}
```

`\GLSaccessplural` Display the plural value (no link and no check for existence) converted to upper
case.

```
\newcommand*{\GLSaccessplural}[1]{%
  \glspluralaccessdisplay
  {%
```

```
    \glsuppercase{\glsentryplural{#1}}%
  }%
  {#1}%
}
```

\GLSaccessfmtplural{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtplural}[3]{%
  \glspluralaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{plural}%
  }%
  {#3}%
}
```

\glsaccessfirst Display the first value (no link and no check for existence).

```
\newcommand*{\glsaccessfirst}[1]{%
  \glsfirstaccessdisplay
  {%
    \glsentryfirst{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtfirst{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtfirst}[3]{%
  \glsfirstaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{first}%
  }%
  {#3}%
}
```

\Glsaccessfirst Display the first value (no link and no check for existence) with the first letter
converted to upper case.

```
\newcommand*{\Glsaccessfirst}[1]{%
  \glsfirstaccessdisplay
  {%
    \Glsentryfirst{#1}%
  }%
  {#1}%
}
```

\Glsaccessfmtfirst

$$\verb|\Glsaccessfmtfirst{|\langle\mathit{insert}\rangle\verb|}{|\langle\mathit{cs}\rangle\verb|}{|\langle\mathit{label}\rangle\verb|}|$$

```
\newcommand*{\Glsaccessfmtfirst}[3]{%
  \glsfirstaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{first}%
  }%
  {#3}%
}
```

\GLSaccessfirst   Display the first value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccessfirst}[1]{%
  \glsfirstaccessdisplay
  {%
    \glsuppercase{\glsentryfirst{#1}}%
  }%
  {#1}%
}
```

\GLSaccessfmtfirst

$$\verb|\GLSaccessfmtfirst{|\langle\mathit{insert}\rangle\verb|}{|\langle\mathit{cs}\rangle\verb|}{|\langle\mathit{label}\rangle\verb|}|$$

```
\newcommand*{\GLSaccessfmtfirst}[3]{%
  \glsfirstaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{first}%
  }%
  {#3}%
}
```

\glsaccessfirstplural   Display the firstplural value (no link and no check for existence).

```
\newcommand*{\glsaccessfirstplural}[1]{%
  \glsfirstpluralaccessdisplay
  {%
    \glsentryfirstplural{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtfirstplural

$$\verb|\glsaccessfmtfirstplural{|\langle\mathit{insert}\rangle\verb|}{|\langle\mathit{cs}\rangle\verb|}{|\langle\mathit{label}\rangle\verb|}|$$

```
\newcommand*{\glsaccessfmtfirstplural}[3]{%
```

```
  \glsfirstpluralaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{firstpl}%
  }%
  {#3}%
}
```

\Glsaccessfirstplural Display the firstplural value (no link and no check for existence) with the first
letter converted to upper case.

```
\newcommand*{\Glsaccessfirstplural}[1]{%
  \glsfirstpluralaccessdisplay
  {%
    \Glsentryfirstplural{#1}%
  }%
  {#1}%
}
```

\Glsaccessfmtfirstplural

\Glsaccessfmtfirstplural{⟨insert⟩}{⟨cs⟩}{⟨label⟩}

```
\newcommand*{\Glsaccessfmtfirstplural}[3]{%
  \glsfirstpluralaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{firstpl}%
  }%
  {#3}%
}
```

\GLSaccessfirstplural Display the firstplural value (no link and no check for existence) converted to
upper case.

```
\newcommand*{\GLSaccessfirstplural}[1]{%
  \glsfirstpluralaccessdisplay
  {%
    \glsuppercase{\glsentryfirstplural{#1}}%
  }%
  {#1}%
}
```

\GLSaccessfmtfirstplural

\GLSaccessfmtfirstplural{⟨insert⟩}{⟨cs⟩}{⟨label⟩}

```
\newcommand*{\GLSaccessfmtfirstplural}[3]{%
  \glsfirstpluralaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{firstpl}%
  }%
```

244

```
      {#3}%
    }
```

\glsaccesssymbol Display the symbol value (no link and no check for existence).

```
\newcommand*{\glsaccesssymbol}[1]{%
  \glssymbolaccessdisplay
  {%
    \glsentrysymbol{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtsymbol

$$\boxed{\texttt{\textbackslash glsaccessfmtsymbol}\{\langle \textit{insert} \rangle\}\{\langle \textit{cs} \rangle\}\{\langle \textit{label} \rangle\}}$$

```
\newcommand*{\glsaccessfmtsymbol}[3]{%
  \glssymbolaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{symbol}%
  }%
  {#3}%
}
```

\Glsaccesssymbol Display the symbol value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccesssymbol}[1]{%
  \glssymbolaccessdisplay
  {%
    \Glsentrysymbol{#1}%
  }%
  {#1}%
}
```

\Glsaccessfmtsymbol

$$\boxed{\texttt{\textbackslash Glsaccessfmtsymbol}\{\langle \textit{insert} \rangle\}\{\langle \textit{cs} \rangle\}\{\langle \textit{label} \rangle\}}$$

```
\newcommand*{\Glsaccessfmtsymbol}[3]{%
  \glssymbolaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{symbol}%
  }%
  {#3}%
}
```

\GLSaccesssymbol Display the symbol value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccesssymbol}[1]{%
  \glssymbolaccessdisplay
  {%
    \glsuppercase{\glsentrysymbol{#1}}%
  }%
  {#1}%
}
```

<div style="border:1px solid; background:#ffffcc;">

$\GLSaccessfmtsymbol\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}$

</div>

\GLSaccessfmtsymbol

```
\newcommand*{\GLSaccessfmtsymbol}[3]{%
  \glssymbolaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{symbol}%
  }%
  {#3}%
}
```

\glsaccesssymbolplural   Display the symbolplural value (no link and no check for existence).

```
\newcommand*{\glsaccesssymbolplural}[1]{%
  \glssymbolpluralaccessdisplay
  {%
    \glsentrysymbolplural{#1}%
  }%
  {#1}%
}
```

<div style="border:1px solid; background:#ffffcc;">

$\glsaccessfmtsymbolplural\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}$

</div>

\glsaccessfmtsymbolplural

```
\newcommand*{\glsaccessfmtsymbolplural}[3]{%
  \glssymbolpluralaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{symbolplural}%
  }%
  {#3}%
}
```

\Glsaccesssymbolplural   Display the symbolplural value (no link and no check for existence) with the first
letter converted to upper case.

```
\newcommand*{\Glsaccesssymbolplural}[1]{%
  \glssymbolpluralaccessdisplay
  {%
    \Glsentrysymbolplural{#1}%
  }%
```

```
    {#1}%
  }
```

\Glsaccessfmtsymbolplural

> \Glsaccessfmtsymbolplural{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\Glsaccessfmtsymbolplural}[3]{%
  \glssymbolpluralaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{symbolplural}%
  }%
  {#3}%
}
```

\GLSaccesssymbolplural Display the symbolplural value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccesssymbolplural}[1]{%
  \glssymbolpluralaccessdisplay
  {%
    \glsuppercase{\glsentrysymbolplural{#1}}%
  }%
  {#1}%
}
```

\GLSaccessfmtsymbolplural

> \GLSaccessfmtsymbolplural{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtsymbolplural}[3]{%
  \glssymbolpluralaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{symbolplural}%
  }%
  {#3}%
}
```

\glsaccessdesc Display the desc value (no link and no check for existence).

```
\newcommand*{\glsaccessdesc}[1]{%
  \glsdescriptionaccessdisplay
  {%
    \glsentrydesc{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtdesc

> \glsaccessfmtdesc{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtdesc}[3]{%
  \glsdescaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{desc}%
  }%
  {#3}%
}
```

\Glsaccessdesc Display the desc value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessdesc}[1]{%
  \glsdescriptionaccessdisplay
  {%
    \Glsentrydesc{#1}%
  }%
  {#1}%
}
```

\Glsaccessfmtdesc{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

\Glsaccessfmtdesc

```
\newcommand*{\Glsaccessfmtdesc}[3]{%
  \glsdescaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{desc}%
  }%
  {#3}%
}
```

\GLSaccessdesc Display the desc value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccessdesc}[1]{%
  \glsdescriptionaccessdisplay
  {%
    \glsuppercase{\glsentrydesc{#1}}%
  }%
  {#1}%
}
```

\GLSaccessfmtdesc{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

\GLSaccessfmtdesc

```
\newcommand*{\GLSaccessfmtdesc}[3]{%
  \glsdescaccessdisplay
  {%
```

```
        \GLSfmtfield{#1}{#2}{#3}{desc}%
      }%
      {#3}%
    }
```

\glsaccessdescplural Display the descplural value (no link and no check for existence).

```
\newcommand*{\glsaccessdescplural}[1]{%
  \glsdescriptionpluralaccessdisplay
  {%
    \glsentrydescplural{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtdescplural

> \glsaccessfmtdescplural{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtdescplural}[3]{%
  \glsdescpluralaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{descplural}%
  }%
  {#3}%
}
```

\Glsaccessdescplural Display the descplural value (no link and no check for existence) with the first
letter converted to upper case.

```
\newcommand*{\Glsaccessdescplural}[1]{%
  \glsdescriptionpluralaccessdisplay
  {%
    \Glsentrydescplural{#1}%
  }%
  {#1}%
}
```

\Glsaccessfmtdescplural

> \Glsaccessfmtdescplural{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\Glsaccessfmtdescplural}[3]{%
  \glsdescpluralaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{descplural}%
  }%
  {#3}%
}
```

\GLSaccessdescplural Display the descplural value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccessdescplural}[1]{%
  \glsdescriptionpluralaccessdisplay
  {%
    \glsuppercase{\glsentrydescplural{#1}}%
  }%
  {#1}%
}
```

\GLSaccessfmtdescplural

> \GLSaccessfmtdescplural{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtdescplural}[3]{%
  \glsdescpluralaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{descplural}%
  }%
  {#3}%
}
```

\glsaccessshort Display the short form (no link and no check for existence).

```
\newcommand*{\glsaccessshort}[1]{%
  \glsshortaccessdisplay
  {%
    \glsentryshort{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtshort

> \glsaccessfmtshort{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtshort}[3]{%
  \glsshortaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{short}%
  }%
  {#3}%
}
```

\Glsaccessshort Display the short form with first letter converted to uppercase (no link and no check for existence).

```
\newcommand*{\Glsaccessshort}[1]{%
  \glsshortaccessdisplay
  {%
```

```
      \Glsentryshort{#1}%
    }%
    {#1}%
  }
```

> \Glsaccessfmtshort{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\Glsaccessfmtshort}[3]{%
  \glsshortaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{short}%
  }%
  {#3}%
}
```

\GLSaccessshort Display the **short** value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccessshort}[1]{%
  \glsshortaccessdisplay
  {%
    \glsuppercase{\glsentryshort{#1}}%
  }%
  {#1}%
}
```

> \GLSaccessfmtshort{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtshort}[3]{%
  \glsshortaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{short}%
  }%
  {#3}%
}
```

\glsaccessshortpl Display the short plural form (no link and no check for existence).

```
\newcommand*{\glsaccessshortpl}[1]{%
  \glsshortpluralaccessdisplay
  {%
    \glsentryshortpl{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtshortpl

> \glsaccessfmtshortpl{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtshortpl}[3]{%
  \glsshortpluralaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{shortpl}%
  }%
  {#3}%
}
```

\Glsaccessshortpl Display the short plural form with first letter converted to uppercase (no link and no check for existence).

```
\newcommand*{\Glsaccessshortpl}[1]{%
  \glsshortpluralaccessdisplay
  {%
    \Glsentryshortpl{#1}%
  }%
  {#1}%
}
```

\Glsaccessfmtshortpl

> \Glsaccessfmtshortpl{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\Glsaccessfmtshortpl}[3]{%
  \glsshortpluralaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{shortpl}%
  }%
  {#3}%
}
```

\GLSaccessshortpl Display the shortplural value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccessshortpl}[1]{%
  \glsshortpluralaccessdisplay
  {%
    \glsuppercase{\glsentryshortpl{#1}}%
  }%
  {#1}%
}
```

\GLSaccessfmtshortpl

> \GLSaccessfmtshortpl{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtshortpl}[3]{%
  \glsshortpluralaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{shortpl}%
  }%
  {#3}%
}
```

\glsaccesslong Display the long form (no link and no check for existence).

```
\newcommand*{\glsaccesslong}[1]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}%
}
```

\glsaccessfmtlong{⟨insert⟩}{⟨cs⟩}{⟨label⟩}

\glsaccessfmtlong

```
\newcommand*{\glsaccessfmtlong}[3]{%
  \glslongaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{long}%
  }%
  {#3}%
}
```

\Glsaccesslong Display the long form (no link and no check for existence).

```
\newcommand*{\Glsaccesslong}[1]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}%
}
```

\Glsaccessfmtlong{⟨insert⟩}{⟨cs⟩}{⟨label⟩}

\Glsaccessfmtlong

```
\newcommand*{\Glsaccessfmtlong}[3]{%
  \glslongaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{long}%
  }%
  {#3}%
}
```

\GLSaccesslong Display the long value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccesslong}[1]{%
  \glslongaccessdisplay
  {%
    \glsuppercase{\glsentrylong{#1}}%
```

```
  }%
  {#1}%
}
```

> \GLSaccessfmtlong{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtlong}[3]{%
  \glslongaccessdisplay
  {%
    \GLSfmtfield{#1}{#2}{#3}{long}%
  }%
  {#3}%
}
```

\glsaccesslongpl Display the long plural form (no link and no check for existence).

```
\newcommand*{\glsaccesslongpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}%
}
```

> \glsaccessfmtlongpl{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtlongpl}[3]{%
  \glslongpluralaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{longpl}%
  }%
  {#3}%
}
```

\Glsaccesslongpl Display the long plural form (no link and no check for existence).

```
\newcommand*{\Glsaccesslongpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}%
}
```

> \Glsaccessfmtlongpl{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\Glsaccessfmtlongpl}[3]{%
  \glslongpluralaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{longpl}%
```

```
    }%
    {#3}%
  }
```

\GLSaccesslongpl Display the longplural value (no link and no check for existence) converted to upper case.

```
\newcommand*{\GLSaccesslongpl}[1]{%
\glslongpluralaccessdisplay
{%
  \glsuppercase{\glsentrylongpl{#1}}%
}%
{#1}%
}
```

\GLSaccessfmtlongpl

> \GLSaccessfmtlongpl{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtlongpl}[3]{%
\glslongpluralaccessdisplay
{%
  \GLSfmtfield{#1}{#2}{#3}{longpl}%
}%
{#3}%
}
```

The user accessibility fields were added to glossaries-accsupp v4.45 so these may not be defined.
USER1

\glsaccessuseri Display the user1 value (no link and no check for existence).

```
\ifdef\glsuseriaccessdisplay
{
  \newcommand*{\glsaccessuseri}[1]{%
    \glsuseriaccessdisplay
    {%
      \glsentryuseri{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\glsaccessuseri}[1]{%
    \glsentryuseri{#1}%
  }
}
```

\glsaccessfmtuseri

> \glsaccessfmtuseri{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\ifdef\glsuseriaccessdisplay
{
  \newcommand*{\glsaccessfmtuseri}[3]{%
    \glsuseriaccessdisplay
    {%
      \glsfmtfield{#1}{#2}{#3}{useri}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\glsaccessfmtuseri}[3]{%
    \glsfmtfield{#1}{#2}{#3}{useri}%
  }
}
```

\Glsaccessuseri   Display the user1 value (no link and no check for existence) with the first letter converted to upper case.

```
\ifdef\glsuseriaccessdisplay
{
  \newcommand*{\Glsaccessuseri}[1]{%
    \glsuseriaccessdisplay
    {%
      \Glsentryuseri{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\Glsaccessuseri}[1]{%
    \Glsentryuseri{#1}%
  }
}
```

\Glsaccessfmtuseri

> $\Glsaccessfmtuseri\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}$

```
\ifdef\glsuseriaccessdisplay
{
  \newcommand*{\Glsaccessfmtuseri}[3]{%
    \glsuseriaccessdisplay
    {%
      \Glsfmtfield{#1}{#2}{#3}{useri}%
    }%
    {#3}%
  }
}
```

256

```
    {
      \newcommand*{\Glsaccessfmtuseri}[3]{%
        \Glsfmtfield{#1}{#2}{#3}{useri}%
      }
    }
```

\GLSaccessuseri Display the user1 value (no link and no check for existence) converted to upper case.

```
    \ifdef\glsuseriaccessdisplay
    {
      \newcommand*{\GLSaccessuseri}[1]{%
        \glsuseriaccessdisplay
        {%
          \glsuppercase{\glsentryuseri{#1}}%
        }%
        {#1}%
      }
    }
    {
      \newcommand*{\GLSaccessuseri}[1]{%
        \glsuppercase{\glsentryuseri{#1}}%
      }
    }
```

\GLSaccessfmtuseri

$\boxed{\texttt{\textbackslash GLSaccessfmtuseri\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{cs}\rangle\texttt{\}\{}\langle\mathit{label}\rangle\texttt{\}}}$

```
    \ifdef\glsuseriaccessdisplay
    {
      \newcommand*{\GLSaccessfmtuseri}[3]{%
        \glsuseriaccessdisplay
        {%
          \GLSfmtfield{#1}{#2}{#3}{useri}%
        }%
        {#3}%
      }
    }
    {
      \newcommand*{\GLSaccessfmtuseri}[3]{%
        \GLSfmtfield{#1}{#2}{#3}{useri}%
      }
    }
```

USER2

\glsaccessuserii Display the user2 value (no link and no check for existence).

```
    \ifdef\glsuseriiaccessdisplay
    {
```

```
                    \newcommand*{\glsaccessuserii}[1]{%
                      \glsuseriiaccessdisplay
                      {%
                        \glsentryuserii{#1}%
                      }%
                      {#1}%
                    }
                  }
                  {
                    \newcommand*{\glsaccessuserii}[1]{%
                      \glsentryuserii{#1}%
                    }
                  }
```

\glsaccessfmtuserii

> **\glsaccessfmtuserii{⟨insert⟩}{⟨cs⟩}{⟨label⟩}**

```
                  \ifdef\glsuseriiaccessdisplay
                  {
                    \newcommand*{\glsaccessfmtuserii}[3]{%
                      \glsuseriiaccessdisplay
                      {%
                        \glsfmtfield{#1}{#2}{#3}{userii}%
                      }%
                      {#3}%
                    }
                  }
                  {
                    \newcommand*{\glsaccessfmtuserii}[3]{%
                      \glsfmtfield{#1}{#2}{#3}{userii}%
                    }
                  }
```

\Glsaccessuserii Display the user2 value (no link and no check for existence) with the first letter
converted to upper case.

```
                  \ifdef\glsuseriiaccessdisplay
                  {
                    \newcommand*{\Glsaccessuserii}[1]{%
                      \glsuseriiaccessdisplay
                      {%
                        \Glsentryuserii{#1}%
                      }%
                      {#1}%
                    }
                  }
                  {
                    \newcommand*{\Glsaccessuserii}[1]{%
                      \Glsentryuserii{#1}%
```

```
      }
  }
```

\Glsaccessfmtuserii

\Glsaccessfmtuserii{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\ifdef\glsuseriiaccessdisplay
{
  \newcommand*{\Glsaccessfmtuserii}[3]{%
    \glsuseriiaccessdisplay
    {%
      \Glsfmtfield{#1}{#2}{#3}{userii}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\Glsaccessfmtuserii}[3]{%
    \Glsfmtfield{#1}{#2}{#3}{userii}%
  }
}
```

\GLSaccessuserii Display the user2 value (no link and no check for existence) converted to upper case.

```
\ifdef\glsuseriiaccessdisplay
{
  \newcommand*{\GLSaccessuserii}[1]{%
    \glsuseriiaccessdisplay
    {%
      \glsuppercase{\glsentryuserii{#1}}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\GLSaccessuserii}[1]{%
    \glsuppercase{\glsentryuserii{#1}}%
  }
}
```

\GLSaccessfmtuserii

\GLSaccessfmtuserii{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\ifdef\glsuseriiaccessdisplay
{
  \newcommand*{\GLSaccessfmtuserii}[3]{%
```

```
      \glsuseriiaccessdisplay
      {%
        \GLSfmtfield{#1}{#2}{#3}{userii}%
      }%
      {#3}%
    }
  }
  {
    \newcommand*{\GLSaccessfmtuserii}[3]{%
      \GLSfmtfield{#1}{#2}{#3}{userii}%
    }
  }
```

USER3

\glsaccessuseriii Display the user3 value (no link and no check for existence).

```
\ifdef\glsuseriiiaccessdisplay
{
  \newcommand*{\glsaccessuseriii}[1]{%
    \glsuseriiiaccessdisplay
    {%
      \glsentryuseriii{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\glsaccessuseriii}[1]{%
    \glsentryuseriii{#1}%
  }
}
```

\glsaccessfmtuseriii{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

\glsaccessfmtuseriii

```
\ifdef\glsuseriiiaccessdisplay
{
  \newcommand*{\glsaccessfmtuseriii}[3]{%
    \glsuseriiiaccessdisplay
    {%
      \glsfmtfield{#1}{#2}{#3}{useriii}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\glsaccessfmtuseriii}[3]{%
    \glsfmtfield{#1}{#2}{#3}{useriii}%
```

```
          }
        }
```

\Glsaccessuseriii   Display the user3 value (no link and no check for existence) with the first letter
                    converted to upper case.

```
\ifdef\glsuseriiiaccessdisplay
{
  \newcommand*{\Glsaccessuseriii}[1]{%
    \glsuseriiiaccessdisplay
    {%
      \Glsentryuseriii{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\Glsaccessuseriii}[1]{%
    \Glsentryuseriii{#1}%
  }
}
```

<div style="border:1px solid black; background:#ffffcc">

\Glsaccessfmtuseriii{⟨insert⟩}{⟨cs⟩}{⟨label⟩}

</div>

\Glsaccessfmtuseriii

```
\ifdef\glsuseriiiaccessdisplay
{
  \newcommand*{\Glsaccessfmtuseriii}[3]{%
    \glsuseriiiaccessdisplay
    {%
      \Glsfmtfield{#1}{#2}{#3}{useriii}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\Glsaccessfmtuseriii}[3]{%
    \Glsfmtfield{#1}{#2}{#3}{useriii}%
  }
}
```

\GLSaccessuseriii   Display the user3 value (no link and no check for existence) converted to upper
                    case.

```
\ifdef\glsuseriiiaccessdisplay
{
  \newcommand*{\GLSaccessuseriii}[1]{%
    \glsuseriiiaccessdisplay
    {%
      \glsuppercase{\glsentryuseriii{#1}}%
```

```
    }%
    {#1}%
  }
}
{
  \newcommand*{\GLSaccessuseriii}[1]{%
    \glsuppercase{\glsentryuseriii{#1}}%
  }
}
```

\GLSaccessfmtuseriii

```
\ifdef\glsuseriiiaccessdisplay
{
  \newcommand*{\GLSaccessfmtuseriii}[3]{%
    \glsuseriiiaccessdisplay
    {%
      \GLSfmtfield{#1}{#2}{#3}{useriii}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\GLSaccessfmtuseriii}[3]{%
    \GLSfmtfield{#1}{#2}{#3}{useriii}%
  }
}
```

USER4

\glsaccessuseriv Display the user4 value (no link and no check for existence).

```
\ifdef\glsuserivaccessdisplay
{
  \newcommand*{\glsaccessuseriv}[1]{%
    \glsuserivaccessdisplay
    {%
      \glsentryuseriv{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\glsaccessuseriv}[1]{%
    \glsentryuseriv{#1}%
  }
}
```

\glsaccessfmtuseriv

<div style="border:1px solid #000; background:#ffffcc; padding:8px;">

\glsaccessfmtuseriv{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

</div>

```
\ifdef\glsuserivaccessdisplay
{
  \newcommand*{\glsaccessfmtuseriv}[3]{%
    \glsuserivaccessdisplay
    {%
      \glsfmtfield{#1}{#2}{#3}{useriv}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\glsaccessfmtuseriv}[3]{%
    \glsfmtfield{#1}{#2}{#3}{useriv}%
  }
}
```

\Glsaccessuseriv Display the user4 value (no link and no check for existence) with the first letter converted to upper case.

```
\ifdef\glsuserivaccessdisplay
{
  \newcommand*{\Glsaccessuseriv}[1]{%
    \glsuserivaccessdisplay
    {%
      \Glsentryuseriv{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\Glsaccessuseriv}[1]{%
    \Glsentryuseriv{#1}%
  }
}
```

\Glsaccessfmtuseriv

<div style="border:1px solid #000; background:#ffffcc; padding:8px;">

\Glsaccessfmtuseriv{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

</div>

```
\ifdef\glsuserivaccessdisplay
{
  \newcommand*{\Glsaccessfmtuseriv}[3]{%
    \glsuserivaccessdisplay
    {%
      \Glsfmtfield{#1}{#2}{#3}{useriv}%
    }%
```

```
      {#3}%
    }
  }
  {
    \newcommand*{\Glsaccessfmtuseriv}[3]{%
      \Glsfmtfield{#1}{#2}{#3}{useriv}%
    }
  }
```

\GLSaccessuseriv Display the user4 value (no link and no check for existence) converted to upper case.

```
  \ifdef\glsuserivaccessdisplay
  {
    \newcommand*{\GLSaccessuseriv}[1]{%
      \glsuserivaccessdisplay
      {%
        \glsuppercase{\glsentryuseriv{#1}}%
      }%
      {#1}%
    }
  }
  {
    \newcommand*{\GLSaccessuseriv}[1]{%
      \glsuppercase{\glsentryuseriv{#1}}%
    }
  }
```

\GLSaccessfmtuseriv

```
  \ifdef\glsuserivaccessdisplay
  {
    \newcommand*{\GLSaccessfmtuseriv}[3]{%
      \glsuserivaccessdisplay
      {%
        \GLSfmtfield{#1}{#2}{#3}{useriv}%
      }%
      {#3}%
    }
  }
  {
    \newcommand*{\GLSaccessfmtuseriv}[3]{%
      \GLSfmtfield{#1}{#2}{#3}{useriv}%
    }
  }
```

USER5

\glsaccessuserv Display the user5 value (no link and no check for existence).

```
\ifdef\glsuservaccessdisplay
{
  \newcommand*{\glsaccessuserv}[1]{%
    \glsuservaccessdisplay
    {%
      \glsentryuserv{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\glsaccessuserv}[1]{%
    \glsentryuserv{#1}%
  }
}
```

\glsaccessfmtuserv

$$\boxed{\texttt{\textbackslash glsaccessfmtuserv\{}\langle \textit{insert}\rangle\texttt{\}\{}\langle \textit{cs}\rangle\texttt{\}\{}\langle \textit{label}\rangle\texttt{\}}}$$

```
\ifdef\glsuservaccessdisplay
{
  \newcommand*{\glsaccessfmtuserv}[3]{%
    \glsuservaccessdisplay
    {%
      \glsfmtfield{#1}{#2}{#3}{userv}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\glsaccessfmtuserv}[3]{%
    \glsfmtfield{#1}{#2}{#3}{userv}%
  }
}
```

\Glsaccessuserv Display the user5 value (no link and no check for existence) with the first letter converted to upper case.

```
\ifdef\glsuservaccessdisplay
{
  \newcommand*{\Glsaccessuserv}[1]{%
    \glsuservaccessdisplay
    {%
      \Glsentryuserv{#1}%
    }%
    {#1}%
  }
}
```

```
{
  \newcommand*{\Glsaccessuserv}[1]{%
    \Glsentryuserv{#1}%
  }
}
```

\Glsaccessfmtuserv

$\backslash\texttt{Glsaccessfmtuserv}\{\langle \mathit{insert} \rangle\}\{\langle \mathit{cs} \rangle\}\{\langle \mathit{label} \rangle\}$

```
\ifdef\glsuservaccessdisplay
{
  \newcommand*{\Glsaccessfmtuserv}[3]{%
    \glsuservaccessdisplay
    {%
      \Glsfmtfield{#1}{#2}{#3}{userv}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\Glsaccessfmtuserv}[3]{%
    \Glsfmtfield{#1}{#2}{#3}{userv}%
  }
}
```

\GLSaccessuserv Display the user5 value (no link and no check for existence) converted to upper case.

```
\ifdef\glsuservaccessdisplay
{
  \newcommand*{\GLSaccessuserv}[1]{%
    \glsuservaccessdisplay
    {%
      \glsuppercase{\glsentryuserv{#1}}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\GLSaccessuserv}[1]{%
    \glsuppercase{\glsentryuserv{#1}}%
  }
}
```

\GLSaccessfmtuserv

$\backslash\texttt{GLSaccessfmtuserv}\{\langle \mathit{insert} \rangle\}\{\langle \mathit{cs} \rangle\}\{\langle \mathit{label} \rangle\}$

```
\ifdef\glsuservaccessdisplay
{
  \newcommand*{\GLSaccessfmtuserv}[3]{%
    \glsuservaccessdisplay
    {%
      \GLSfmtfield{#1}{#2}{#3}{userv}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\GLSaccessfmtuserv}[3]{%
    \GLSfmtfield{#1}{#2}{#3}{userv}%
  }
}
```

USER6

\glsaccessuservi Display the user6 value (no link and no check for existence).

```
\ifdef\glsuserviaccessdisplay
{
  \newcommand*{\glsaccessuservi}[1]{%
    \glsuserviaccessdisplay
    {%
      \glsentryuservi{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\glsaccessuservi}[1]{%
    \glsentryuservi{#1}%
  }
}
```

\glsaccessfmtuservi

> \glsaccessfmtuservi{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\ifdef\glsuserviaccessdisplay
{
  \newcommand*{\glsaccessfmtuservi}[3]{%
    \glsuserviaccessdisplay
    {%
      \glsfmtfield{#1}{#2}{#3}{uservi}%
    }%
    {#3}%
  }
}
```

```
    {
      \newcommand*{\glsaccessfmtuservi}[3]{%
        \glsfmtfield{#1}{#2}{#3}{uservi}%
      }
    }
```

\Glsaccessuservi Display the user6 value (no link and no check for existence) with the first letter
converted to upper case.

```
    \ifdef\glsuserviaccessdisplay
    {
      \newcommand*{\Glsaccessuservi}[1]{%
        \glsuserviaccessdisplay
        {%
          \Glsentryuservi{#1}%
        }%
        {#1}%
      }
    }
    {
      \newcommand*{\Glsaccessuservi}[1]{%
        \Glsentryuservi{#1}%
      }
    }
```

\Glsaccessfmtuservi

$\boxed{\texttt{\textbackslash Glsaccessfmtuservi}\{\langle\mathit{insert}\rangle\}\{\langle\mathit{cs}\rangle\}\{\langle\mathit{label}\rangle\}}$

```
    \ifdef\glsuserviaccessdisplay
    {
      \newcommand*{\Glsaccessfmtuservi}[3]{%
        \glsuserviaccessdisplay
        {%
          \Glsfmtfield{#1}{#2}{#3}{uservi}%
        }%
        {#3}%
      }
    }
    {
      \newcommand*{\Glsaccessfmtuservi}[3]{%
        \Glsfmtfield{#1}{#2}{#3}{uservi}%
      }
    }
```

\GLSaccessuservi Display the user6 value (no link and no check for existence) converted to upper
case.

```
    \ifdef\glsuserviaccessdisplay
    {
      \newcommand*{\GLSaccessuservi}[1]{%
```

```

```
        \glsuserviaccessdisplay
        {%
          \glsuppercase{\glsentryuservi{#1}}%
        }%
        {#1}%
      }
    }
    {
      \newcommand*{\GLSaccessuservi}[1]{%
        \glsuppercase{\glsentryuservi{#1}}%
      }
    }
```

$\boxed{\texttt{\textbackslash GLSaccessfmtuservi\{}\langle\textit{insert}\rangle\texttt{\}\{}\langle\textit{cs}\rangle\texttt{\}\{}\langle\textit{label}\rangle\texttt{\}}}$

```
\ifdef\glsuserviaccessdisplay
{
  \newcommand*{\GLSaccessfmtuservi}[3]{%
    \glsuserviaccessdisplay
    {%
      \GLSfmtfield{#1}{#2}{#3}{uservi}%
    }%
    {#3}%
  }
}
{
  \newcommand*{\GLSaccessfmtuservi}[3]{%
    \GLSfmtfield{#1}{#2}{#3}{uservi}%
  }
}
```

Keys for accessibility support while pre-parsing in `\newabbreviation`.

```
\define@key{glsxtrabbrv}{access}{%
  \def\@gls@nameaccess{#1}%
}

\define@key{glsxtrabbrv}{textaccess}{%
  \def\@gls@textaccess{#1}%
}

\define@key{glsxtrabbrv}{pluralaccess}{%
  \def\@gls@pluralaccess{#1}%
}

\define@key{glsxtrabbrv}{firstaccess}{%
  \def\@gls@firstaccess{#1}%
}

\define@key{glsxtrabbrv}{firstpluralaccess}{%
```

269

```
                        \def\@gls@firstpluralaccess{#1}%
            }

            \define@key{glsxtrabbrv}{shortaccess}{%
              \def\@gls@shortaccess{#1}%
            }

            \define@key{glsxtrabbrv}{shortpluralaccess}{%
              \def\@gls@shortaccesspl{#1}%
            }

            \define@key{glsxtrabbrv}{longaccess}{%
              \def\@gls@longaccess{#1}%
            }

            \define@key{glsxtrabbrv}{longpluralaccess}{%
              \def\@gls@longaccesspl{#1}%
            }
```

\@gls@initaccesskeys

```
            \newcommand*{\@gls@initaccesskeys}{%
              \def\@gls@nameaccess{}%
              \def\@gls@textaccess{}%
              \def\@gls@pluralaccess{}%
              \def\@gls@firstaccess{}%
              \def\@gls@firstpluralaccess{}%
              \def\@gls@shortaccess{}%
              \def\@gls@shortaccesspl{}%
              \def\@gls@longaccess{}%
              \def\@gls@longaccesspl{}%
            }
```

\@gls@ifaccessattribute@set

> \gls@ifaccessattribute@set{⟨*attribute*⟩}{⟨*true*⟩}{⟨*false*⟩}

```
            \newcommand*{\@gls@ifaccessattribute@set}[3]{%
              \glsifcategoryattribute{\glscategorylabel}{access#1}{true}%
              {#2}%
              {%
                \glsifcategoryattribute{\glscategorylabel}{access#1}{false}%
                {#3}%
                {%
                  \glsifcategoryattribute{\glscategorylabel}{#1}{true}%
                  {#2}%
                  {#3}%
                }%
              }%
            }
```

As from glossaries v4.45, the replacement text support has been corrected so that the accessibility support for abbreviations use the "E" (expanded value) element. This should actually contain the long form since it's supposed to explain the abbreviation. This is a bit redundant on first use for styles like long-short.

\glsdefaultshortaccess

\glsdefaultshortaccess{⟨*long*⟩}{⟨*short*⟩}

This command was only introduced to glossaries-accsupp 4.45 so it may not be defined. This was defined to do #1 (#2) but the original definition is more appropriate, so has been reverted back to the definition provided by glossaries-accsupp.

```
\providecommand*{\glsdefaultshortaccess}[2]{#1}
```

\glsxtrassignactualsetup

```
\newcommand{\glsxtrassignactualsetup}{%
 \let\@\empty
 \let\emph\@firstofone
 \let\textbf\@firstofone
 \let\textmd\@firstofone
 \let\textit\@firstofone
 \let\textsl\@firstofone
 \let\textsc\@firstofone
 \let\textrm\@firstofone
 \let\textsf\@firstofone
 \let\texttt\@firstofone
 \let\glstextup\@firstofone
}
```

\@gls@assign@actual

```
\newcommand{\@gls@assign@actual}{%
 \begingroup
  \glsxtrassignactualsetup
  \protected@edef\@gls@tmp{\endgroup
    \def\noexpand\@gls@actualshort{\glsxtrorgshort}%
    \def\noexpand\@gls@actuallong{\glsxtrorglong}%
    \def\noexpand\@gls@actualshortpl{\@gls@shortpl}%
    \def\noexpand\@gls@actuallongpl{\@gls@longpl}%
 }%
 \@gls@tmp
}
```

\@gls@setup@default@short@access Renamed \@gls@setup@default@access and removed argument since it can be obtained from \glsxtrorgshort.

\@gls@setup@default@access Assign the default value of the shortaccess key. The argument is the short value passed to \newabbreviation. The shortaccess value should explain the abbreviation.

```
\newcommand{\@gls@setup@default@access}{%
  \@gls@assign@actual
  \ifdefempty\@gls@shortaccess
  {%
```

Check if the **accessinsertdots** attribute has been set but only if **shortaccess** hasn't been set.

```
    \@gls@ifaccessattribute@set{insertdots}%
    {%
      \expandafter\@glsxtr@insertdots\expandafter\@gls@actualshort\expandafter
       {\@gls@actualshort}%
    }%
    {}%
    \ifdefempty\@gls@longaccess
    {%

      \protected@edef\@gls@shortaccess{\glsdefaultshortaccess
        {\expandonce\@gls@actuallong}{\expandonce\@gls@actualshort}}%
    }%
    {%
      \protected@edef\@gls@shortaccess{\glsdefaultshortaccess
        {\expandonce\@gls@longaccess}{\expandonce\@gls@actualshort}}%
    }%
    \eappto\ExtraCustomAbbreviationFields{shortaccess={\@gls@shortaccess},}%
```

If **shortaccessplural** hasn't been set, assign plural form.

```
    \ifdefempty\@gls@shortaccesspl
    {%
      \@gls@ifaccessattribute@set{aposplural}%
      {%
        \expandafter\def\expandafter\@gls@shortaccesspl\expandafter{%
          \@gls@actualshort'\glsxtrabbrvpluralsuffix}%
      }%
      {%
        \@gls@ifaccessattribute@set{noshortplural}%
        {%
           \let\@gls@shortaccesspl\@gls@shortaccess
        }%
        {%
          \let\@gls@shortaccesspl\@gls@actualshortpl
        }%
      }%
    }%
    \ifdefempty\@gls@longaccesspl
    {%

      \protected@edef\@gls@shortaccesspl{\glsdefaultshortaccess
        {\expandonce\@gls@actuallongpl}{\expandonce\@gls@actualshortpl}}%
    }%
    {%
      \protected@edef\@gls@shortaccesspl{\glsdefaultshortaccess
        {\expandonce\@gls@longaccesspl}{\expandonce\@gls@actualshort}}%
```

```
    }%
    \eappto\ExtraCustomAbbreviationFields{shortpluralaccess={\@gls@shortaccesspl},}%
    }%
    {}%
  }%
  {%
    \ifdefempty\@gls@shortaccesspl
    {\let\@gls@shortaccesspl\@gls@shortaccess}%
    {}%
  }%
```

If access key hasn't been set, check if the nameshortaccess attribute has been set.

```
    \ifdefempty\@gls@nameaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{nameshortaccess}{true}%
      {%
        \eappto\ExtraCustomAbbreviationFields{access={\@gls@shortaccess},}%
      }%
      {}%
    }%
    {}%
```

If textaccess key hasn't been set, check if the textshortaccess attribute has been set.

```
    \ifdefempty\@gls@textaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{textshortaccess}{true}%
      {%
        \eappto\ExtraCustomAbbreviationFields{textaccess={\@gls@shortaccess},}%
      }%
      {}%
    }%
    {}%
    \ifdefempty\@gls@pluralaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{textshortaccess}{true}%
      {%
        \eappto\ExtraCustomAbbreviationFields{%
          pluralaccess={\@gls@shortaccesspl},%
        }%
      }%
      {}%
    }%
    {}%
```

If firstaccess key hasn't been set, check if the firstshortaccess attribute has been set.

```
    \ifdefempty\@gls@firstaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{firstshortaccess}{true}%
```

273

```
    {%
      \eappto\ExtraCustomAbbreviationFields{firstaccess={\@gls@shortaccess},}%
    }%
    {}%
  }%
  {}%
  \ifdefempty\@gls@firstpluralaccess
  {%
    \glsifcategoryattribute{\glscategorylabel}{firstshortaccess}{true}%
    {%
      \eappto\ExtraCustomAbbreviationFields{%
         firstpluralaccess={\@gls@shortaccesspl},%
      }%
    }%
    {}%
  }%
  {}%
}
```

Provide hooks for \setabbreviationstyle that automatically set the attributes appropriate for the style. If the name is just the short form and the description contains the long form, then it may not be necessary to set nameshortaccess but it would depend on the glossary style.

Need to provide \glsxtr⟨*category*⟩⟨*field*⟩accsupp if not already defined.

\glsxtrprovideaccsuppcmd

```
\newcommand*{\glsxtrprovideaccsuppcmd}[2]{%
  \ifcsundef{glsxtr#1#2accsupp}%
  {\csdef{glsxtr#1#2accsupp}{\glsshortaccsupp}}%
  {}%
}
```

\glsxtrAccSuppAbbrSetNoLongAttrs For styles where the name, first and text are just the abbreviation.

```
\newcommand*{\glsxtrAccSuppAbbrSetNoLongAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glssetcategoryattribute{#1}{firstshortaccess}{true}%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
  \glsxtrprovideaccsuppcmd{#1}{name}%
  \glsxtrprovideaccsuppcmd{#1}{first}%
  \glsxtrprovideaccsuppcmd{#1}{firstpl}%
  \glsxtrprovideaccsuppcmd{#1}{text}%
  \glsxtrprovideaccsuppcmd{#1}{plural}%
}
```

\glsxtrAccSuppAbbrSetFirstLongAttrs For styles where the name and text are just the abbreviation. The first form may just be long or may be short and long.

```
\newcommand*{\glsxtrAccSuppAbbrSetFirstLongAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
```

```
\glsxtrprovideaccsuppcmd{#1}{name}%
\glsxtrprovideaccsuppcmd{#1}{text}%
\glsxtrprovideaccsuppcmd{#1}{plural}%
}
```

AccSuppAbbrSetTextShortAttrs For styles where only the text is just the abbreviation. The name and first form may just be long or may be short and long. The name may also be short but followed by the long form in the description.

```
\newcommand*{\glsxtrAccSuppAbbrSetTextShortAttrs}[1]{%
\glssetcategoryattribute{#1}{textshortaccess}{true}%
\glsxtrprovideaccsuppcmd{#1}{text}%
\glsxtrprovideaccsuppcmd{#1}{plural}%
}
```

AccSuppAbbrSetNameShortAttrs For styles where only the name is just the abbreviation. The first and subsequent form may just be long or may be short and long.

```
\newcommand*{\glsxtrAccSuppAbbrSetNameShortAttrs}[1]{%
\glssetcategoryattribute{#1}{nameshortaccess}{true}%
\glsxtrprovideaccsuppcmd{#1}{name}%
}
```

AccSuppAbbrSetNameLongAttrs For styles where the first and text are just the abbreviation. The name may just be long or may be short and long or the name may be short.

```
\newcommand*{\glsxtrAccSuppAbbrSetNameLongAttrs}[1]{%
\glssetcategoryattribute{#1}{firstshortaccess}{true}%
\glssetcategoryattribute{#1}{textshortaccess}{true}%
\glsxtrprovideaccsuppcmd{#1}{first}%
\glsxtrprovideaccsuppcmd{#1}{firstpl}%
\glsxtrprovideaccsuppcmd{#1}{text}%
\glsxtrprovideaccsuppcmd{#1}{plural}%
}
```

End of if accsupp part
```
}
{
```
No accessibility support. Just define these commands to do \glsentry⟨*xxx*⟩

\glsaccessname Display the name value (no link and no check for existence).

```
\newcommand*{\glsaccessname}[1]{\glsentryname{#1}}
```

\glsaccessfmtname{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

\glsaccessfmtname

```
\newcommand*{\glsaccessfmtname}[3]{%
\glsfmtfield{#1}{#2}{#3}{name}%
}
```

275

\Glsaccessname Display the name value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessname}[1]{\Glsentryname{#1}}
```

\Glsaccessfmtname

$\Glsaccessfmtname\{\langle insert \rangle\}\{\langle cs \rangle\}\{\langle label \rangle\}$

```
\newcommand*{\Glsaccessfmtname}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{name}%
}
```

\GLSaccessname Display the name value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessname}[1]{%
 \protect\glsuppercase{\glsentryname{#1}}}
```

\GLSaccessfmtname

$\GLSaccessfmtname\{\langle insert \rangle\}\{\langle cs \rangle\}\{\langle label \rangle\}$

```
\newcommand*{\GLSaccessfmtname}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{name}%
}
```

\glsaccesstext Display the text value (no link and no check for existence).

```
\newcommand*{\glsaccesstext}[1]{\glsentrytext{#1}}
```

\glsaccessfmttext

$\glsaccessfmttext\{\langle insert \rangle\}\{\langle cs \rangle\}\{\langle label \rangle\}$

```
\newcommand*{\glsaccessfmttext}[3]{%
  \glsfmtfield{#1}{#2}{#3}{text}%
}
```

\Glsaccesstext Display the text value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccesstext}[1]{\Glsentrytext{#1}}
```

\Glsaccessfmttext

$\Glsaccessfmttext\{\langle insert \rangle\}\{\langle cs \rangle\}\{\langle label \rangle\}$

```
\newcommand*{\Glsaccessfmttext}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{text}%
}
```

`\GLSaccesstext` Display the text value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccesstext}[1]{%
 \protect\glsuppercase{\glsentrytext{#1}}}
```

`\GLSaccessfmttext{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\GLSaccessfmttext`

```
\newcommand*{\GLSaccessfmttext}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{text}%
}
```

`\glsaccessplural` Display the plural value (no link and no check for existence).

```
\newcommand*{\glsaccessplural}[1]{\glsentryplural{#1}}
```

`\glsaccessfmtplural{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\glsaccessfmtplural`

```
\newcommand*{\glsaccessfmtplural}[3]{%
  \glsfmtfield{#1}{#2}{#3}{plural}%
}
```

`\Glsaccessplural` Display the plural value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessplural}[1]{\Glsentryplural{#1}}
```

`\Glsaccessfmtplural{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\Glsaccessfmtplural`

```
\newcommand*{\Glsaccessfmtplural}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{plural}%
}
```

`\GLSaccessplural` Display the plural value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessplural}[1]{%
 \protect\glsuppercase{\glsentryplural{#1}}}
```

`\GLSaccessfmtplural{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\GLSaccessfmtplural`

```
\newcommand*{\GLSaccessfmtplural}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{plural}%
}
```

`\glsaccessfirst` Display the first value (no link and no check for existence).

> `\newcommand*{\glsaccessfirst}[1]{\glsentryfirst{#1}}`

`\glsaccessfmtfirst{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\glsaccessfmtfirst`

```
\newcommand*{\glsaccessfmtfirst}[3]{%
  \glsfmtfield{#1}{#2}{#3}{first}%
}
```

`\Glsaccessfirst` Display the first value (no link and no check for existence) with the first letter converted to upper case.

> `\newcommand*{\Glsaccessfirst}[1]{\Glsentryfirst{#1}}`

`\Glsaccessfmtfirst{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\Glsaccessfmtfirst`

```
\newcommand*{\Glsaccessfmtfirst}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{first}%
}
```

`\GLSaccessfirst` Display the first value (no link and no check for existence). converted to upper case.

> ```
> \newcommand*{\GLSaccessfirst}[1]{%
>  \protect\glsuppercase{\glsentryfirst{#1}}}
> ```

`\GLSaccessfmtfirst{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\GLSaccessfmtfirst`

```
\newcommand*{\GLSaccessfmtfirst}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{first}%
}
```

`\glsaccessfirstplural` Display the firstplural value (no link and no check for existence).

> `\newcommand*{\glsaccessfirstplural}[1]{\glsentryfirstplural{#1}}`

`\glsaccessfmtfirstplural{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

`\glsaccessfmtfirstplural`

```
\newcommand*{\glsaccessfmtfirstplural}[3]{%
  \glsfmtfield{#1}{#2}{#3}{firstpl}%
}
```

\Glsaccessfirstplural Display the firstplural value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessfirstplural}[1]{\Glsentryfirstplural{#1}}
```

\Glsaccessfmtfirstplural

$\Glsaccessfmtfirstplural\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}$

```
\newcommand*{\Glsaccessfmtfirstplural}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{firstpl}%
}
```

\GLSaccessfirstplural Display the firstplural value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessfirstplural}[1]{%
  \protect\glsuppercase{\glsentryfirstplural{#1}}}
```

\GLSaccessfmtfirstplural

$\GLSaccessfmtfirstplural\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}$

```
\newcommand*{\GLSaccessfmtfirstplural}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{firstpl}%
}
```

\glsaccesssymbol Display the symbol value (no link and no check for existence).

```
\newcommand*{\glsaccesssymbol}[1]{\glsentrysymbol{#1}}
```

\glsaccessfmtsymbol

$\glsaccessfmtsymbol\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}$

```
\newcommand*{\glsaccessfmtsymbol}[3]{%
  \glsfmtfield{#1}{#2}{#3}{symbol}%
}
```

\Glsaccesssymbol Display the symbol value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccesssymbol}[1]{\Glsentrysymbol{#1}}
```

\Glsaccessfmtsymbol

$\Glsaccessfmtsymbol\{\langle insert\rangle\}\{\langle cs\rangle\}\{\langle label\rangle\}$

```
\newcommand*{\Glsaccessfmtsymbol}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{symbol}%
}
```

`\GLSaccesssymbol` Display the symbol value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccesssymbol}[1]{%
\protect\glsuppercase{\glsentrysymbol{#1}}}
```

`\GLSaccessfmtsymbol`

$\boxed{\texttt{\textbackslash GLSaccessfmtsymbol\{}\langle\textit{insert}\rangle\texttt{\}\{}\langle\textit{cs}\rangle\texttt{\}\{}\langle\textit{label}\rangle\texttt{\}}}$

```
\newcommand*{\GLSaccessfmtsymbol}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{symbol}%
}
```

`\glsaccesssymbolplural` Display the symbolplural value (no link and no check for existence).

```
\newcommand*{\glsaccesssymbolplural}[1]{\glsentrysymbolplural{#1}}
```

`\glsaccessfmtsymbolplural`

$\boxed{\texttt{\textbackslash glsaccessfmtsymbolplural\{}\langle\textit{insert}\rangle\texttt{\}\{}\langle\textit{cs}\rangle\texttt{\}\{}\langle\textit{label}\rangle\texttt{\}}}$

```
\newcommand*{\glsaccessfmtsymbolplural}[3]{%
  \glsfmtfield{#1}{#2}{#3}{symbolplural}%
}
```

`\Glsaccesssymbolplural` Display the symbolplural value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccesssymbolplural}[1]{\Glsentrysymbolplural{#1}}
```

`\Glsaccessfmtsymbolplural`

$\boxed{\texttt{\textbackslash Glsaccessfmtsymbolplural\{}\langle\textit{insert}\rangle\texttt{\}\{}\langle\textit{cs}\rangle\texttt{\}\{}\langle\textit{label}\rangle\texttt{\}}}$

```
\newcommand*{\Glsaccessfmtsymbolplural}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{symbolplural}%
}
```

`\GLSaccesssymbolplural` Display the symbolplural value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccesssymbolplural}[1]{%
\protect\glsuppercase{\glsentrysymbolplural{#1}}}
```

`\GLSaccessfmtsymbolplural`

$\boxed{\texttt{\textbackslash GLSaccessfmtsymbolplural\{}\langle\textit{insert}\rangle\texttt{\}\{}\langle\textit{cs}\rangle\texttt{\}\{}\langle\textit{label}\rangle\texttt{\}}}$

```
\newcommand*{\GLSaccessfmtsymbolplural}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{symbolplural}%
}
```

`\glsaccessdesc` Display the desc value (no link and no check for existence).

> `\newcommand*{\glsaccessdesc}[1]{\glsentrydesc{#1}}`

`\glsaccessfmtdesc`

> ```
> \glsaccessfmtdesc{⟨insert⟩}{⟨cs⟩}{⟨label⟩}
> ```

> ```
> \newcommand*{\glsaccessfmtdesc}[3]{%
>   \glsfmtfield{#1}{#2}{#3}{desc}%
> }
> ```

`\Glsaccessdesc` Display the desc value (no link and no check for existence) with the first letter converted to upper case.

> `\newcommand*{\Glsaccessdesc}[1]{\Glsentrydesc{#1}}`

`\Glsaccessfmtdesc`

> ```
> \Glsaccessfmtdesc{⟨insert⟩}{⟨cs⟩}{⟨label⟩}
> ```

> ```
> \newcommand*{\Glsaccessfmtdesc}[3]{%
>   \Glsfmtfield{#1}{#2}{#3}{desc}%
> }
> ```

`\GLSaccessdesc` Display the desc value (no link and no check for existence). converted to upper case.

> ```
> \newcommand*{\GLSaccessdesc}[1]{%
>  \protect\glsuppercase{\glsentrydesc{#1}}}
> ```

`\GLSaccessfmtdesc`

> ```
> \GLSaccessfmtdesc{⟨insert⟩}{⟨cs⟩}{⟨label⟩}
> ```

> ```
> \newcommand*{\GLSaccessfmtdesc}[3]{%
>   \GLSfmtfield{#1}{#2}{#3}{desc}%
> }
> ```

`\glsaccessdescplural` Display the descplural value (no link and no check for existence).

> `\newcommand*{\glsaccessdescplural}[1]{\glsentrydescplural{#1}}`

`\glsaccessfmtdescplural`

> ```
> \glsaccessfmtdescplural{⟨insert⟩}{⟨cs⟩}{⟨label⟩}
> ```

> ```
> \newcommand*{\glsaccessfmtdescplural}[3]{%
>   \glsfmtfield{#1}{#2}{#3}{descplural}%
> }
> ```

`\Glsaccessdescplural` Display the descplural value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessdescplural}[1]{\Glsentrydescplural{#1}}
```

`\Glsaccessfmtdescplural`

$\boxed{\texttt{\textbackslash Glsaccessfmtdescplural\{}\langle \textit{insert}\rangle \texttt{\}\{}\langle \textit{cs}\rangle \texttt{\}\{}\langle \textit{label}\rangle \texttt{\}}}$

```
\newcommand*{\Glsaccessfmtdescplural}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{descplural}%
}
```

`\GLSaccessdescplural` Display the descplural value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessdescplural}[1]{%
 \protect\glsuppercase{\glsentrydescplural{#1}}}
```

`\GLSaccessfmtdescplural`

$\boxed{\texttt{\textbackslash GLSaccessfmtdescplural\{}\langle \textit{insert}\rangle \texttt{\}\{}\langle \textit{cs}\rangle \texttt{\}\{}\langle \textit{label}\rangle \texttt{\}}}$

```
\newcommand*{\GLSaccessfmtdescplural}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{descplural}%
}
```

`\glsaccessshort` Display the short form (no link and no check for existence).

```
\newcommand*{\glsaccessshort}[1]{\glsentryshort{#1}}
```

`\glsaccessfmtshort`

$\boxed{\texttt{\textbackslash glsaccessfmtshort\{}\langle \textit{insert}\rangle \texttt{\}\{}\langle \textit{cs}\rangle \texttt{\}\{}\langle \textit{label}\rangle \texttt{\}}}$

```
\newcommand*{\glsaccessfmtshort}[3]{%
  \glsfmtfield{#1}{#2}{#3}{short}%
}
```

`\Glsaccessshort` Display the short form with first letter converted to uppercase (no link and no check for existence).

```
\newcommand*{\Glsaccessshort}[1]{\Glsentryshort{#1}}
```

`\Glsaccessfmtshort`

$\boxed{\texttt{\textbackslash Glsaccessfmtshort\{}\langle \textit{insert}\rangle \texttt{\}\{}\langle \textit{cs}\rangle \texttt{\}\{}\langle \textit{label}\rangle \texttt{\}}}$

```
\newcommand*{\Glsaccessfmtshort}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{short}%
}
```

\GLSaccessshort Display the short value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessshort}[1]{%
 \protect\glsuppercase{\glsentryshort{#1}}}
```

\GLSaccessfmtshort

$\boxed{\text{\texttt{\textbackslash GLSaccessfmtshort\{}⟨\textit{insert}⟩\texttt{\}\{}⟨\textit{cs}⟩\texttt{\}\{}⟨\textit{label}⟩\texttt{\}}}}$

```
\newcommand*{\GLSaccessfmtshort}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{short}%
}
```

\glsaccessshortpl Display the short plural form (no link and no check for existence).

```
\newcommand*{\glsaccessshortpl}[1]{\glsentryshortpl{#1}}
```

\glsaccessfmtshortpl

$\boxed{\text{\texttt{\textbackslash glsaccessfmtshortpl\{}⟨\textit{insert}⟩\texttt{\}\{}⟨\textit{cs}⟩\texttt{\}\{}⟨\textit{label}⟩\texttt{\}}}}$

```
\newcommand*{\glsaccessfmtshortpl}[3]{%
  \glsfmtfield{#1}{#2}{#3}{shortpl}%
}
```

\Glsaccessshortpl Display the short plural form with first letter converted to uppercase (no link and no check for existence).

```
\newcommand*{\Glsaccessshortpl}[1]{\Glsentryshortpl{#1}}
```

\Glsaccessfmtshortpl

$\boxed{\text{\texttt{\textbackslash Glsaccessfmtshortpl\{}⟨\textit{insert}⟩\texttt{\}\{}⟨\textit{cs}⟩\texttt{\}\{}⟨\textit{label}⟩\texttt{\}}}}$

```
\newcommand*{\Glsaccessfmtshortpl}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{shortpl}%
}
```

\GLSaccessshortpl Display the shortplural value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessshortpl}[1]{%
 \protect\glsuppercase{\glsentryshortpl{#1}}}
```

\GLSaccessfmtshortpl

$\boxed{\text{\texttt{\textbackslash GLSaccessfmtshortpl\{}⟨\textit{insert}⟩\texttt{\}\{}⟨\textit{cs}⟩\texttt{\}\{}⟨\textit{label}⟩\texttt{\}}}}$

```
\newcommand*{\GLSaccessfmtshortpl}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{shortpl}%
}
```

\glsaccesslong Display the long form (no link and no check for existence).

```
\newcommand*{\glsaccesslong}[1]{\glsentrylong{#1}}
```

\glsaccessfmtlong

> \glsaccessfmtlong{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtlong}[3]{%
  \glsfmtfield{#1}{#2}{#3}{long}%
}
```

\Glsaccesslong Display the long form (no link and no check for existence).

```
\newcommand*{\Glsaccesslong}[1]{\Glsentrylong{#1}}
```

\Glsaccessfmtlong

> \Glsaccessfmtlong{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\Glsaccessfmtlong}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{long}%
}
```

\GLSaccesslong Display the long value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccesslong}[1]{%
 \protect\glsuppercase{\glsentrylong{#1}}}
```

\GLSaccessfmtlong

> \GLSaccessfmtlong{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtlong}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{long}%
}
```

\glsaccesslongpl Display the long plural form (no link and no check for existence).

```
\newcommand*{\glsaccesslongpl}[1]{\glsentrylongpl{#1}}
```

\glsaccessfmtlongpl

> \glsaccessfmtlongpl{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtlongpl}[3]{%
  \glsfmtfield{#1}{#2}{#3}{longpl}%
}
```

`\Glsaccesslongpl` Display the long plural form (no link and no check for existence).

```
\newcommand*{\Glsaccesslongpl}[1]{\Glsentrylongpl{#1}}
```

`\Glsaccessfmtlongpl`

> `\Glsaccessfmtlongpl{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\Glsaccessfmtlongpl}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{longpl}%
}
```

`\GLSaccesslongpl` Display the longplural value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccesslongpl}[1]{%
  \protect\glsuppercase{\glsentrylongpl{#1}}}
```

`\GLSaccessfmtlongpl`

> `\GLSaccessfmtlongpl{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\GLSaccessfmtlongpl}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{longpl}%
}
```

USER1

`\glsaccessuseri` Display the user1 value (no link and no check for existence).

```
\newcommand*{\glsaccessuseri}[1]{\glsentryuseri{#1}}
```

`\glsaccessfmtuseri`

> `\glsaccessfmtuseri{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\glsaccessfmtuseri}[3]{%
  \glsfmtfield{#1}{#2}{#3}{useri}%
}
```

`\Glsaccessuseri` Display the user1 value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessuseri}[1]{\Glsentryuseri{#1}}
```

`\Glsaccessfmtuseri`

> `\Glsaccessfmtuseri{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\Glsaccessfmtuseri}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{useri}%
}
```

`\GLSaccessuseri` Display the user1 value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessuseri}[1]{%
 \protect\glsuppercase{\glsentryuseri{#1}}}
```

`\GLSaccessfmtuseri`

> `\GLSaccessfmtuseri{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\GLSaccessfmtuseri}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{useri}%
}
```

USER2

`\glsaccessuserii` Display the user2 value (no link and no check for existence).

```
\newcommand*{\glsaccessuserii}[1]{\glsentryuserii{#1}}
```

`\glsaccessfmtuserii`

> `\glsaccessfmtuserii{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\glsaccessfmtuserii}[3]{%
  \glsfmtfield{#1}{#2}{#3}{userii}%
}
```

`\Glsaccessuserii` Display the user2 value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessuserii}[1]{\Glsentryuserii{#1}}
```

`\Glsaccessfmtuserii`

> `\Glsaccessfmtuserii{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\Glsaccessfmtuserii}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{userii}%
}
```

`\GLSaccessuserii` Display the user2 value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessuserii}[1]{%
 \protect\glsuppercase{\glsentryuserii{#1}}}
```

`\GLSaccessfmtuserii`

> `\GLSaccessfmtuserii{⟨insert⟩}{⟨cs⟩}{⟨label⟩}`

```
\newcommand*{\GLSaccessfmtuserii}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{userii}%
}
```

USER3

\glsaccessuseriii Display the user3 value (no link and no check for existence).

```
\newcommand*{\glsaccessuseriii}[1]{\glsentryuseriii{#1}}
```

\glsaccessfmtuseriii

$\boxed{\text{\textbackslash glsaccessfmtuseriii}\{\langle insert \rangle\}\{\langle cs \rangle\}\{\langle label \rangle\}}$

```
\newcommand*{\glsaccessfmtuseriii}[3]{%
  \glsfmtfield{#1}{#2}{#3}{useriii}%
}
```

\Glsaccessuseriii Display the user3 value (no link and no check for existence) with the first letter
converted to upper case.

```
\newcommand*{\Glsaccessuseriii}[1]{\Glsentryuseriii{#1}}
```

\Glsaccessfmtuseriii

$\boxed{\text{\textbackslash Glsaccessfmtuseriii}\{\langle insert \rangle\}\{\langle cs \rangle\}\{\langle label \rangle\}}$

```
\newcommand*{\Glsaccessfmtuseriii}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{useriii}%
}
```

\GLSaccessuseriii Display the user3 value (no link and no check for existence). converted to upper
case.

```
\newcommand*{\GLSaccessuseriii}[1]{%
 \protect\glsuppercase{\glsentryuseriii{#1}}}
```

\GLSaccessfmtuseriii

$\boxed{\text{\textbackslash GLSaccessfmtuseriii}\{\langle insert \rangle\}\{\langle cs \rangle\}\{\langle label \rangle\}}$

```
\newcommand*{\GLSaccessfmtuseriii}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{useriii}%
}
```

USER4

\glsaccessuseriv Display the user4 value (no link and no check for existence).

```
\newcommand*{\glsaccessuseriv}[1]{\glsentryuseriv{#1}}
```

287
```

\glsaccessfmtuseriv

> \glsaccessfmtuseriv{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtuseriv}[3]{%
  \glsfmtfield{#1}{#2}{#3}{useriv}%
}
```

\Glsaccessuseriv Display the user4 value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessuseriv}[1]{\Glsentryuseriv{#1}}
```

\Glsaccessfmtuseriv

> \Glsaccessfmtuseriv{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\Glsaccessfmtuseriv}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{useriv}%
}
```

\GLSaccessuseriv Display the user4 value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessuseriv}[1]{%
 \protect\glsuppercase{\glsentryuseriv{#1}}}
```

\GLSaccessfmtuseriv

> \GLSaccessfmtuseriv{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtuseriv}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{useriv}%
}
```

USER5

\glsaccessuserv Display the user5 value (no link and no check for existence).

```
\newcommand*{\glsaccessuserv}[1]{\glsentryuserv{#1}}
```

\glsaccessfmtuserv

> \glsaccessfmtuserv{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\glsaccessfmtuserv}[3]{%
  \glsfmtfield{#1}{#2}{#3}{userv}%
}
```

\Glsaccessuserv Display the user5 value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessuserv}[1]{\Glsentryuserv{#1}}
```

\Glsaccessfmtuserv

$\boxed{\texttt{\textbackslash Glsaccessfmtuserv\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{cs}\rangle\texttt{\}\{}\langle\mathit{label}\rangle\texttt{\}}}$

```
\newcommand*{\Glsaccessfmtuserv}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{userv}%
}
```

\GLSaccessuserv Display the user5 value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessuserv}[1]{%
 \protect\glsuppercase{\glsentryuserv{#1}}}
```

\GLSaccessfmtuserv

$\boxed{\texttt{\textbackslash GLSaccessfmtuserv\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{cs}\rangle\texttt{\}\{}\langle\mathit{label}\rangle\texttt{\}}}$

```
\newcommand*{\GLSaccessfmtuserv}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{userv}%
}
```

USER6

\glsaccessuservi Display the user6 value (no link and no check for existence).

```
\newcommand*{\glsaccessuservi}[1]{\glsentryuservi{#1}}
```

\glsaccessfmtuservi

$\boxed{\texttt{\textbackslash glsaccessfmtuservi\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{cs}\rangle\texttt{\}\{}\langle\mathit{label}\rangle\texttt{\}}}$

```
\newcommand*{\glsaccessfmtuservi}[3]{%
  \glsfmtfield{#1}{#2}{#3}{uservi}%
}
```

\Glsaccessuservi Display the user6 value (no link and no check for existence) with the first letter converted to upper case.

```
\newcommand*{\Glsaccessuservi}[1]{\Glsentryuservi{#1}}
```

\Glsaccessfmtuservi

$\boxed{\texttt{\textbackslash Glsaccessfmtuservi\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{cs}\rangle\texttt{\}\{}\langle\mathit{label}\rangle\texttt{\}}}$

```
\newcommand*{\Glsaccessfmtuservi}[3]{%
  \Glsfmtfield{#1}{#2}{#3}{uservi}%
}
```

\GLSaccessuservi   Display the user6 value (no link and no check for existence). converted to upper case.

```
\newcommand*{\GLSaccessuservi}[1]{%
 \protect\glsuppercase{\glsentryuservi{#1}}}
```

\GLSaccessfmtuservi

> \GLSaccessfmtuservi{⟨*insert*⟩}{⟨*cs*⟩}{⟨*label*⟩}

```
\newcommand*{\GLSaccessfmtuservi}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{uservi}%
}
```

\@gls@initaccesskeys   This does nothing if there's no accessibility support.
```
\newcommand*{\@gls@initaccesskeys}{}
```

\@gls@setup@default@access   This does nothing if there's no accessibility support.
```
\newcommand{\@gls@setup@default@access}{}
```

xtrAccSuppAbbrSetNoLongAttrs   This does nothing if there's no accessibility support.
```
\newcommand*{\glsxtrAccSuppAbbrSetNoLongAttrs}[1]{}
```

AccSuppAbbrSetFirstLongAttrs   This does nothing if there's no accessibility support.
```
\newcommand*{\glsxtrAccSuppAbbrSetFirstLongAttrs}[1]{}
```

AccSuppAbbrSetTextShortAttrs   This does nothing if there's no accessibility support.
```
\newcommand*{\glsxtrAccSuppAbbrSetTextShortAttrs}[1]{}
```

AccSuppAbbrSetNameShortAttrs   This does nothing if there's no accessibility support.
```
\newcommand*{\glsxtrAccSuppAbbrSetNameShortAttrs}[1]{}
```

AccSuppAbbrSetNameLongAttrs   This does nothing if there's no accessibility support.
```
\newcommand*{\glsxtrAccSuppAbbrSetNameLongAttrs}[1]{}
```

End of else part
```
 }
```

Identify sentence-case mappings:
```
\glsmfuaddmap{\glsaccessname}{\Glsaccessname}
\glsmfuaddmap{\glsaccessfmtname}{\Glsaccessfmtname}
\glsmfublocker{\GLSaccessname}
\glsmfublocker{\GLSaccessfmtname}
\glsmfuaddmap{\glsaccesstext}{\Glsaccesstext}
\glsmfuaddmap{\glsaccessfmttext}{\Glsaccessfmttext}
```

290

```
\glsmfublocker{\GLSaccesstext}
\glsmfublocker{\GLSaccessfmttext}
\glsmfuaddmap{\glsaccessplural}{\Glsaccessplural}
\glsmfuaddmap{\glsaccessfmtplural}{\Glsaccessfmtplural}
\glsmfublocker{\GLSaccessplural}
\glsmfublocker{\GLSaccessfmtplural}
\glsmfuaddmap{\glsaccessfirst}{\Glsaccessfirst}
\glsmfuaddmap{\glsaccessfmtfirst}{\Glsaccessfmtfirst}
\glsmfublocker{\GLSaccessfirst}
\glsmfublocker{\GLSaccessfmtfirst}
\glsmfuaddmap{\glsaccessfirstplural}{\Glsaccessfirstplural}
\glsmfuaddmap{\glsaccessfmtfirstplural}{\Glsaccessfmtfirstplural}
\glsmfublocker{\GLSaccessfirstplural}
\glsmfublocker{\GLSaccessfmtfirstplural}
\glsmfuaddmap{\glsaccesssymbol}{\Glsaccesssymbol}
\glsmfuaddmap{\glsaccessfmtsymbol}{\Glsaccessfmtsymbol}
\glsmfublocker{\GLSaccesssymbol}
\glsmfublocker{\GLSaccessfmtsymbol}
\glsmfuaddmap{\glsaccesssymbolplural}{\Glsaccesssymbolplural}
\glsmfuaddmap{\glsaccessfmtsymbolplural}{\Glsaccessfmtsymbolplural}
\glsmfublocker{\GLSaccesssymbolplural}
\glsmfublocker{\GLSaccessfmtsymbolplural}
\glsmfuaddmap{\glsaccessdesc}{\Glsaccessdesc}
\glsmfuaddmap{\glsaccessfmtdesc}{\Glsaccessfmtdesc}
\glsmfublocker{\GLSaccessdesc}
\glsmfublocker{\GLSaccessfmtdesc}
\glsmfuaddmap{\glsaccessdescplural}{\Glsaccessdescplural}
\glsmfuaddmap{\glsaccessfmtdescplural}{\Glsaccessfmtdescplural}
\glsmfublocker{\GLSaccessdescplural}
\glsmfublocker{\GLSaccessfmtdescplural}
\glsmfuaddmap{\glsaccessshort}{\Glsaccessshort}
\glsmfuaddmap{\glsaccessfmtshort}{\Glsaccessfmtshort}
\glsmfublocker{\GLSaccessshort}
\glsmfublocker{\GLSaccessfmtshort}
\glsmfuaddmap{\glsaccessshortpl}{\Glsaccessshortpl}
\glsmfuaddmap{\glsaccessfmtshortpl}{\Glsaccessfmtshortpl}
\glsmfublocker{\GLSaccessshortpl}
\glsmfublocker{\GLSaccessfmtshortpl}
\glsmfuaddmap{\glsaccesslong}{\Glsaccesslong}
\glsmfuaddmap{\glsaccessfmtlong}{\Glsaccessfmtlong}
\glsmfublocker{\GLSaccesslong}
\glsmfublocker{\GLSaccessfmtlong}
\glsmfuaddmap{\glsaccesslongpl}{\Glsaccesslongpl}
\glsmfuaddmap{\glsaccessfmtlongpl}{\Glsaccessfmtlongpl}
\glsmfublocker{\GLSaccesslongpl}
\glsmfublocker{\GLSaccessfmtlongpl}
\glsmfuaddmap{\glsaccessuseri}{\Glsaccessuseri}
\glsmfuaddmap{\glsaccessfmtuseri}{\Glsaccessfmtuseri}
\glsmfublocker{\GLSaccessuseri}
\glsmfublocker{\GLSaccessfmtuseri}
```

```
\glsmfuaddmap{\glsaccessuserii}{\Glsaccessuserii}
\glsmfuaddmap{\glsaccessfmtuserii}{\Glsaccessfmtuserii}
\glsmfublocker{\GLSaccessuserii}
\glsmfublocker{\GLSaccessfmtuserii}
\glsmfuaddmap{\glsaccessuseriii}{\Glsaccessuseriii}
\glsmfuaddmap{\glsaccessfmtuseriii}{\Glsaccessfmtuseriii}
\glsmfublocker{\GLSaccessuseriii}
\glsmfublocker{\GLSaccessfmtuseriii}
\glsmfuaddmap{\glsaccessuseriv}{\Glsaccessuseriv}
\glsmfuaddmap{\glsaccessfmtuseriv}{\Glsaccessfmtuseriv}
\glsmfublocker{\GLSaccessuseriv}
\glsmfublocker{\GLSaccessfmtuseriv}
\glsmfuaddmap{\glsaccessuserv}{\Glsaccessuserv}
\glsmfuaddmap{\glsaccessfmtuserv}{\Glsaccessfmtuserv}
\glsmfublocker{\GLSaccessuserv}
\glsmfublocker{\GLSaccessfmtuserv}
\glsmfuaddmap{\glsaccessuservi}{\Glsaccessuservi}
\glsmfuaddmap{\glsaccessfmtuservi}{\Glsaccessfmtuservi}
\glsmfublocker{\GLSaccessuservi}
\glsmfublocker{\GLSaccessfmtuservi}
```

1.6   Categories

\glscategory   Add a new storage key that can be used to indicate a category. The default category is general.

```
\glsaddstoragekey{category}{general}{\glscategory}
```

\glsifcategory   Convenient shortcut to determine if an entry has the given category.

```
\newcommand{\glsifcategory}[4]{%
 \ifglsfieldeq{#1}{category}{#2}{#3}{#4}%
}
```

Categories can have attributes.

> \glssetcategoryattribute{⟨*category*⟩}{⟨*attribute-label*⟩}
> {⟨*value*⟩}

\glssetcategoryattribute

Set (or override if already set) an attribute for the given category.

```
\newcommand*{\glssetcategoryattribute}[3]{%
  \csdef{@glsxtr@categoryattr@@#1@#2}{#3}%
}
```

> \glssetcategoriesattribute{⟨*category list*⟩}
> {⟨*attribute-label*⟩}{⟨*value*⟩}

\glssetcategoriesattribute

Similar to above, but globally apply to each category in the list.

```
\newcommand*{\glssetcategoriesattribute}[3]{%
  \@for\@gls@thiscatlabel:=#1\do{%
    \csgdef{@glsxtr@categoryattr@@\@gls@thiscatlabel @#2}{#3}%
  }%
}
```

\glssetcategoriesattributes{⟨*category list*⟩}
{⟨*attribute-label list*⟩}{⟨*value*⟩}

\glssetcategoriesattributes

Similar to above, but apply to each category and attribute in the list.

```
\newcommand*{\glssetcategoriesattributes}[3]{%
```

Group to avoid problems with nested \@for.

```
  {%
    \@for\@gls@thisattrlabel:=#2\do{%
      \glssetcategoriesattribute{#1}{\@gls@thisattrlabel}{#3}%
    }%
  }%
}
```

\glssetcategoryattributes{⟨*category*⟩}{⟨*attribute list*⟩}
{⟨*value*⟩}

\glssetcategoryattributes

Similar to above, but globally apply to each attribute in the list to the given category.

```
\newcommand*{\glssetcategoryattributes}[3]{%
  \@for\@gls@thisattrlabel:=#2\do{%
    \csgdef{@glsxtr@categoryattr@@#1@\@gls@thisattrlabel}{#3}%
  }%
}
```

\glsgetcategoryattribute{⟨*category*⟩}{⟨*attribute-label*⟩}

\glsgetcategoryattribute

Get the value of the given attribute for the given category. Does nothing if the attribute isn't defined.

```
\newcommand*{\glsgetcategoryattribute}[2]{%
  \csuse{@glsxtr@categoryattr@@#1@#2}%
}
```

\glsunsetcategoryattribute{⟨*category*⟩}{⟨*attribute-label*⟩}

\glsunsetcategoryattribute

Unsets the given attribute for the given category.

```
\newcommand*{\glsunsetcategoryattribute}[2]{%
```

```
  \csundef{@glsxtr@categoryattr@@#1@#2}%
}
```

\glshascategoryattribute

\glshascategoryattribute{⟨*category*⟩}{⟨*attribute-label*⟩}
{⟨*true*⟩}{⟨*false*⟩}

Tests if the category has the given attribute set.

```
\newcommand*{\glshascategoryattribute}[4]{%
  \ifcsvoid{@glsxtr@categoryattr@@#1@#2}{#4}{#3}%
}
```

\glssetattribute

\glssetattribute{⟨*entry label*⟩}{⟨*attribute-label*⟩}{⟨*value*⟩}

Short cut where the category label is obtained from the entry information.

```
\newcommand*{\glssetattribute}[3]{%
  \glssetcategoryattribute{\glscategory{#1}}{#2}{#3}%
}
```

\glsgetattribute

\glsgetattribute{⟨*entry label*⟩}{⟨*attribute-label*⟩}

Short cut where the category label is obtained from the entry information.

```
\newcommand*{\glsgetattribute}[2]{%
  \glsgetcategoryattribute{\glscategory{#1}}{#2}%
}
```

\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.

```
\newcommand*{\glshasattribute}[4]{%
  \ifglsentryexists{#1}%
  {\glshascategoryattribute{\glscategory{#1}}{#2}{#3}{#4}}%
  {#4}%
}
```

\glsifcategoryattribute

\glsifcategoryattribute{⟨*category*⟩}{⟨*attribute-label*⟩}
{⟨*value*⟩}{⟨*true
part*⟩}{⟨*false part*⟩}

True if category has the attribute with the given value.

294

```
\newcommand{\glsifcategoryattribute}[5]{%
 \ifcsundef{@glsxtr@categoryattr@@#1@#2}%
 {#5}%
 {\ifcsstring{@glsxtr@categoryattr@@#1@#2}{#3}{#4}{#5}}%
}
```

> $\glsifattribute\{\langle entry\ label\rangle\}\{\langle attribute\text{-}label\rangle\}\{\langle value\rangle\}$
> $\{\langle true$
> $part\rangle\}\{\langle false\ part\rangle\}$

\glsifattribute

Short cut to determine if the given entry has a category with the given attribute set.

```
\newcommand{\glsifattribute}[5]{%
  \ifglsentryexists{#1}%
  {\glsifcategoryattribute{\glscategory{#1}}{#2}{#3}{#4}{#5}}%
  {#5}%
}
```

Provide expandable test to determine if attribute is set to true.

\@glsxtr@truevalue

```
\newcommand*{\@glsxtr@truevalue}{true}
```

> $\glsifcategoryattributetrue\{\langle category\text{-}label\rangle\}\{\langle attribute\rangle\}$
> $\{\langle true\rangle\}\{\langle false\rangle\}$

\glsifcategoryattributetrue

Does $\langle false\rangle$ if the entry hasn't been defined.

```
\newcommand*{\glsifcategoryattributetrue}[4]{%
 \ifcsequal{@glsxtr@categoryattr@@#1@#2}%
 {@glsxtr@truevalue}%
 {#3}{#4}%
}
```

> $\glsifattributetrue\{\langle label\rangle\}\{\langle attribute\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}$

\glsifattributetrue

Does $\langle false\rangle$ if the entry hasn't been defined.

```
\newcommand*{\glsifattributetrue}[4]{%
 \ifcsundef{glo@\glsdetoklabel{#1}@category}%
 {#4}
 {\ifcsequal
  {@glsxtr@categoryattr@@\csname glo@\glsdetoklabel{#1}@category\endcsname @#2}%
  {@glsxtr@truevalue}%
  {#3}{#4}%
 }%
}
```

> \glsifcategoryattributehasitem{⟨category⟩}
> {⟨attribute-label⟩}{⟨item⟩}{⟨true
> part⟩}{⟨false part⟩}

True if category has the attribute (whose value is a comma-separated list) contains the given item. The ⟨item⟩ is expanded.

```
\newrobustcmd{\glsifcategoryattributehasitem}[5]{%
\ifcsundef{@glsxtr@categoryattr@@#1@#2}%
{#5}%
{%
 \protected@edef\gls@tmp{%
   \noexpand\DTLifinlist{#3}{\csuse{@glsxtr@categoryattr@@#1@#2}}}%
 \gls@tmp{#4}{#5}%
}%
}
```

Set attributes for the default general category:

```
\glssetcategoryattribute{general}{regular}{true}
```

Acronyms are regular by default, since they're typically just treated like normal words.

```
\glssetcategoryattribute{acronym}{regular}{true}
```

\glssetregularcategory Convenient shortcut to add the regular attribute.

```
\newcommand*{\glssetregularcategory}[1]{%
\glssetcategoryattribute{#1}{regular}{true}%
}
```

> \glsifregularcategory{⟨category⟩}{⟨true part⟩}{⟨false part⟩}

\glsifregularcategory

Short cut to determine if a category has the regular attribute explicitly set to true.

```
\newcommand{\glsifregularcategory}[3]{%
  \glsifcategoryattribute{#1}{regular}{true}{#2}{#3}%
}
```

> \glsifnotregularcategory{⟨category⟩}{⟨true part⟩}{⟨false
> part⟩}

\glsifnotregularcategory

Short cut to determine if a category has the regular attribute explicitly set to false.

```
\newcommand{\glsifnotregularcategory}[3]{%
  \glsifcategoryattribute{#1}{regular}{false}{#2}{#3}%
}
```

> \glsifregular{⟨*entry label*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

Short cut to determine if an entry has a regular attribute set to true.

```
\newcommand{\glsifregular}[3]{%
  \glsifregularcategory{\glscategory{#1}}{#2}{#3}%
}
```

> \glsifnotregular{⟨*entry label*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

Short cut to determine if an entry has a regular attribute set to false.

```
\newcommand{\glsifnotregular}[3]{%
  \glsifnotregularcategory{\glscategory{#1}}{#2}{#3}%
}
```

> \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.

```
\newcommand{\glsforeachincategory}[5][\@glo@types]{%
  \forallglossaries[#1]{#3}%
  {%

    \glsxtrifemptyglossary{#3}{}%
    {%
      \forglsentries[#3]{#4}%
      {%
        \glsifcategory{#4}{#2}{#5}{}%
      }%
    }%
  }%
}
```

> \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.

```
\newcommand{\glsforeachwithattribute}[6][\@glo@types]{%
  \forallglossaries[#1]{#4}%
  {%
    \forglsentries[#4]{#5}%
    {%
      \glsifattribute{#5}{#2}{#3}{#6}{}%
    }%
  }%
}
```

If `\newterm` has been defined, redefine it so that it automatically sets the category label to index and add `\glsxtrpostdescription`.

```
\ifdef\newterm
{%
```

`\newterm`

```
  \renewcommand*{\newterm}[2][]{%
    \newglossaryentry{#2}%
    {type={index},category=index,name={#2},%
     description={\glsxtrpostdescription\nopostdesc},#1}%
  }
```

Indexed terms are regular by default.

```
  \glssetcategoryattribute{index}{regular}{true}
```

`\glsxtrpostdescindex`

```
  \newcommand*{\glsxtrpostdescindex}{}
```

```
}
{}
```

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.

```
\ifdef\printsymbols
{%
```

`\glsxtrnewsymbol` Unlike `\newterm`, this has a separate argument for the label (since the symbol will likely contain commands).

```
  \newcommand*{\glsxtrnewsymbol}[3][]{%
    \newglossaryentry{#2}{name={#3},sort={#2},type=symbols,category=symbol,#1}%
  }
```

Symbols are regular by default.

```
  \glssetcategoryattribute{symbol}{regular}{true}
```

298

`\glsxtrpostdescsymbol`

```
                    \newcommand*{\glsxtrpostdescsymbol}{}

}
{}
```

Similar for the numbers option.
```
\ifdef\printnumbers
{%
```

`\glsxtrnewnumber`
```
\ifdef\printnumbers
\newcommand*{\glsxtrnewnumber}[3][]{%
  \newglossaryentry{#2}{name={#3},sort={#2},type=numbers,category=number,#1}%
}
```

Numbers are regular by default.
```
\glssetcategoryattribute{number}{regular}{true}
```

`\glsxtrpostdescnumber`
```
                    \newcommand*{\glsxtrpostdescnumber}{}

}
{}
```

`\glsxtrsetcategory` Set the category for all listed labels. The first argument is the list of entry labels and the second argument is the category label.
```
\newcommand*{\glsxtrsetcategory}[2]{%
  \@for\@glsxtr@label:=#1\do
  {%
    \glsfieldxdef{\@glsxtr@label}{category}{#2}%
  }%
}
```

`\glsxtrsetcategoryforall` 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.
```
\newcommand*{\glsxtrsetcategoryforall}[2]{%
  \forallglossaries[#1]{\@glsxtr@type}{%
    \forglsentries[\@glsxtr@type]{\@glsxtr@label}%
    {%
      \glsfieldxdef{\@glsxtr@label}{category}{#2}%
    }%
  }%
}
```

> $\text{\textbackslash glsxtrfieldtitlecase}\{\langle label \rangle\}\{\langle field \rangle\}$

`\glsxtrfieldtitlecase`

Apply title casing to the contents of the given field.

299

```
\newcommand*{\glsxtrfieldtitlecase}[2]{%
  \expandafter\glsxtrfieldtitlecasecs\expandafter
    {\csname glo@\glsdetoklabel{#1}@#2\endcsname}%
}
```

\glsxtrfieldtitlecasecs The command used by `\glsxtrfieldtitlecase`. May be redefined to use a different command, for example, `\xcapitalisefmtwords`. Check for `\glscapitalisewords`, which was added to glossaries v4.48.

```
\ifdef\glscapitalisewords
{
  \newcommand*{\glsxtrfieldtitlecasecs}[1]{%
   \expandafter\glscapitalisewords\expandafter{#1}}
}
{
  \newcommand*{\glsxtrfieldtitlecasecs}[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.

```
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
```

As from version 1.04, allow the glossdescfont attribute to determine the font applied.

```
        \glshasattribute{#1}{glossdescfont}%
        {%
          \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossdescfont}}%
          \ifcsdef{\@glsxtr@attrval}%
          {%
            \letcs\@glsxtr@glossdescfont{\@glsxtr@attrval}%
          }%
          {%
            \GlossariesExtraWarning{Unknown control sequence name
            '\@glsxtr@attrval' supplied in glossdescfont attribute
            for entry '#1'. Ignoring}%
            \let\@glsxtr@glossdescfont\@firstofone
          }%
        }%
        {\let\@glsxtr@glossdescfont\@firstofone}%
        \glsifattribute{#1}{glossdesc}{firstuc}%
        {%
          \@glsxtr@glossdescfont{\Glsaccessdesc{#1}}%
        }%
```

300

```
      {%
        \glsifattribute{#1}{glossdesc}{title}%
        {%
          \@glsxtr@do@titlecaps@warn
          \glsdescriptionaccessdisplay
          {%
            \@glsxtr@glossdescfont{\glsxtrfieldtitlecase{#1}{desc}}%
          }%
          {#1}%
        }%
        {%
          \@glsxtr@glossdescfont{\glsaccessdesc{#1}}%
        }%
      }%
    }%
  }
}
{
  \renewcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossdescfont}%
      {%

        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossdescfont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossdescfont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          '\@glsxtr@attrval' supplied in glossdescfont attribute
          for entry '#1'. Ignoring}%
          \let\@glsxtr@glossdescfont\@firstofone
        }%
      }%
      {\let\@glsxtr@glossdescfont\@firstofone}%
      \glsifattribute{#1}{glossdesc}{firstuc}%
      {%
        \@glsxtr@glossdescfont{\Glsentrydesc{#1}}%
      }%
      {%
        \glsifattribute{#1}{glossdesc}{title}%
        {%
          \@glsxtr@do@titlecaps@warn
          \@glsxtr@glossdescfont{\glsxtrfieldtitlecase{#1}{desc}}%
        }%
        {%
          \@glsxtr@glossdescfont{\glsentrydesc{#1}}%
```

```
              }%
            }%
          }%
        }
      }
```

\glossentryname If the **glossname** attribute is "firstuc" convert first letter to upper case. If the attribute is "title" use title case.

```
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\glossentryname}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
```

As from v1.54, do pre-name hook:

```
      \glsxtrprenamehook{#1}%
```

As from version 1.04, allow the **glossnamefont** attribute to determine the font applied.

```
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          '\@glsxtr@attrval' supplied in glossnamefont attribute
          for entry '#1'. Reverting to default \string\glsnamefont}%
          \let\@glsxtr@glossnamefont\glsnamefont
        }%
      }%
      {\let\@glsxtr@glossnamefont\glsnamefont}%
      \glsifattribute{#1}{glossname}{firstuc}%
      {%
        \glsnameaccessdisplay
        {%
          \@glsxtr@glossnamefont{\Glsentryname{#1}}%
        }%
        {#1}%
      }%
      {%
        \glsifattribute{#1}{glossname}{title}%
        {%
          \@glsxtr@do@titlecaps@warn
          \glsnameaccessdisplay
          {%
            \@glsxtr@glossnamefont{\glsxtrfieldtitlecase{#1}{name}}%
```

302

```
          }%
          {#1}%
        }%
        {%
          \glsifattribute{#1}{glossname}{uc}%
          {%
            \glsnameaccessdisplay
            {%
```
Hide the label from the upper-casing command.
```
              \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
              \@glsxtr@glossnamefont{\glsuppercase{\glo@name}}%
            }%
            {#1}%
          }%
          {%
            \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
            \glsnameaccessdisplay
            {%
              \expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}%
            }%
            {#1}%
          }%
        }%
      }%
```
Do post-name hook:
```
      \glsxtrpostnamehook{#1}%
    }%
  }
 }
 {
   \renewcommand*{\glossentryname}[1]{%
     \@glsdoifexistsorwarn{#1}%
     {%
       \glssetabbrvfmt{\glscategory{#1}}%
```
As from v1.54, do pre-name hook:
```
       \glsxtrprenamehook{#1}%
       \glshasattribute{#1}{glossnamefont}%
       {%

         \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
         \ifcsdef{\@glsxtr@attrval}%
         {%
           \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
         }%
         {%
           \GlossariesExtraWarning{Unknown control sequence name
            '\@glsxtr@attrval' supplied in glossnamefont attribute
            for entry '#1'. Reverting to default \string\glsnamefont}%
```

```
        \let\@glsxtr@glossnamefont\glsnamefont
      }%
    }%
    {\let\@glsxtr@glossnamefont\glsnamefont}%
    \glsifattribute{#1}{glossname}{firstuc}%
    {%
      \@glsxtr@glossnamefont{\Glsentryname{#1}}%
    }%
    {%
      \glsifattribute{#1}{glossname}{title}%
      {%
        \@glsxtr@do@titlecaps@warn
        \@glsxtr@glossnamefont{\glsxtrfieldtitlecase{#1}{name}}%
      }%
      {%
        \glsifattribute{#1}{glossname}{uc}%
        {%
```

Hide the label from the upper-casing command.

```
          \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
          \@glsxtr@glossnamefont{\glsuppercase{\glo@name}}%
        }%
        {%
```

This little trick is used by glossaries to allow the user to redefine `\glsnamefont`
to use `\makefirstuc`. Support it even though they can now use the firstuc
attribute.

```
          \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
          \expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}%
        }%
      }%
    }%
```

Do post-name hook.

```
    \glsxtrpostnamehook{#1}%
  }%
  }
}
```

`\Glossentryname` Redefine to set the abbreviation format and accessibility support.

```
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\Glossentryname}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
```

As from v1.54, do pre-name hook:

```
    \glsxtrprenamehook{#1}%
```

304

As from version 1.04, allow the glossnamefont attribute to determine the font applied.

```
        \glshasattribute{#1}{glossnamefont}%
        {%

          \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
          \ifcsdef{\@glsxtr@attrval}%
          {%
            \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
          }%
          {%
            \GlossariesExtraWarning{Unknown control sequence name
            '\@glsxtr@attrval' supplied in glossnamefont attribute
            for entry '#1'. Reverting to default \string\glsnamefont}%
            \let\@glsxtr@glossnamefont\glsnamefont
          }%
        }%
        {\let\@glsxtr@glossnamefont\glsnamefont}%
        \glsnameaccessdisplay
        {%
          \@glsxtr@glossnamefont{\Glsentryname{#1}}%
        }%
        {#1}%
```

Do post-name hook:

```
        \glsxtrpostnamehook{#1}%
      }%
    }
  }
  {
    \renewcommand*{\Glossentryname}[1]{%
      \@glsdoifexistsorwarn{#1}%
      {%
        \glssetabbrvfmt{\glscategory{#1}}%
```

As from v1.54, do pre-name hook:

```
        \glsxtrprenamehook{#1}%
        \glshasattribute{#1}{glossnamefont}%
        {%

          \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
          \ifcsdef{\@glsxtr@attrval}%
          {%
            \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
          }%
          {%
            \GlossariesExtraWarning{Unknown control sequence name
            '\@glsxtr@attrval' supplied in glossnamefont attribute
            for entry '#1'. Reverting to default \string\glsnamefont}%
            \let\@glsxtr@glossnamefont\glsnamefont
          }%
```

```
    }%
    {\let\@glsxtr@glossnamefont\glsnamefont}%
    \@glsxtr@glossnamefont{\Glsentryname{#1}}%
```

Do post-name hook:

```
    \glsxtrpostnamehook{#1}%
  }%
 }
}
```

\GlossEntryName This command was only added to glossaries v4.59 so provide a definition in case an older version is installed.

```
\providecommand*{\GlossEntryName}[1]{%
 \glsnamefont{\glsentrytitlecase{#1}{name}}%
}
```

Redefine to set the abbreviation format and accessibility support.

```
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\GlossEntryName}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
```

Do pre-name hook:

```
    \glsxtrprenamehook{#1}%
```

Allow the glossnamefont attribute to determine the font applied.

```
    \glshasattribute{#1}{glossnamefont}%
    {%
      \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
      \ifcsdef{\@glsxtr@attrval}%
      {%
        \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
      }%
      {%
        \GlossariesExtraWarning{Unknown control sequence name
         '\@glsxtr@attrval' supplied in glossnamefont attribute
         for entry '#1'. Reverting to default \string\glsnamefont}%
        \let\@glsxtr@glossnamefont\glsnamefont
      }%
    }%
    {\let\@glsxtr@glossnamefont\glsnamefont}%
    \glsnameaccessdisplay
    {%
      \@glsxtr@glossnamefont{\glsentrytitlecase{#1}{name}}%
    }%
    {#1}%
```

Do post-name hook:

```
    \glsxtrpostnamehook{#1}%
```

```
      }%
    }
  }
  {
    \renewcommand*{\GlossEntryName}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
```
Do pre-name hook:
```
      \glsxtrprenamehook{#1}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          '\@glsxtr@attrval' supplied in glossnamefont attribute
          for entry '#1'. Reverting to default \string\glsnamefont}%
          \let\@glsxtr@glossnamefont\glsnamefont
        }%
      }%
      {\let\@glsxtr@glossnamefont\glsnamefont}%
      \@glsxtr@glossnamefont{\glsentrytitlecase{#1}{name}}%
```
Do post-name hook:
```
      \glsxtrpostnamehook{#1}%
    }%
  }
}
```

\glsxtrprenamehook

```
\newcommand*{\glsxtrprenamehook}[1]{}
```

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.

\glsxtrpostnamehook Hook to append stuff after the name is displayed in the glossary. The argument
is the entry's label.

```
\newcommand*{\glsxtrpostnamehook}[1]{%
  \let\@glsnumberformat\@glsxtr@defaultnumberformat
  \glsxtrdoautoindexname{#1}{indexname}%
```
Allow additional code regardless of category:

```
  \glsextrapostnamehook{#1}%
```

Allow categories to hook in here.

```
    \csuse{glsxtrpostname\glscategory{#1}}%
  }
```

\glsextrapostnamehook

```
    \newcommand*{\glsextrapostnamehook}[1]{}%
```

\glsdefpostname Provide a convenient command for defining the post-name hook for the given category.

```
    \newcommand*{\glsdefpostname}[2]{%
      \csdef{glsxtrpostname#1}{#2}%
    }
```

\glsxtr@setaccessdisplay

```
    \@ifpackageloaded{glossaries-accsupp}
    {
      \newcommand*{\glsxtr@setaccessdisplay}[1]{%
        \ifcsdef{gls#1accessdisplay}%
        {\letcs\@glsxtr@accessdisplay{gls#1accessdisplay}}%
        {%
```

This is essentially the reverse of `\@gls@fetchfield`, since the field supplied to `\glossentryname` has to be the internal label, but the `\gls⟨field⟩accessdisplay` commands use the key name.

```
          \protected@edef\@gls@thisval{#1}%
          \@for\@gls@map:=\@gls@keymap\do{%
            \protected@edef\@this@key{\expandafter\@secondoftwo\@gls@map}%
            \ifdefequal{\@this@key}{\@gls@thisval}%
            {%
              \protected@edef\@gls@thisval{\expandafter\@firstoftwo\@gls@map}%
              \@endfortrue
            }%
            {}%
          }%
          \ifcsdef{gls\@gls@thisval accessdisplay}%
          {\letcs\@glsxtr@accessdisplay{gls\@gls@thisval accessdisplay}}%
          {\let\@glsxtr@accessdisplay\@firstoftwo}%
        }%
      }
    }
    {%
      \newcommand*{\glsxtr@setaccessdisplay}[1]{%
        \let\@glsxtr@accessdisplay\@firstoftwo}
    }
```

\glossentrynameother Provide a command that works like `\glossentryname` but accesses a different field (which must be supplied using its internal field label).

```
    \newrobustcmd*{\glossentrynameother}[2]{%
      \@glsdoifexistsorwarn{#1}%
      {%
```

Accessibility support:

```
    \glsxtr@setaccessdisplay{#2}%
```

Set the abbreviation format:

```
    \glssetabbrvfmt{\glscategory{#1}}%
```

As from v1.54, do pre-name hook:

```
    \glsxtrprenamehook{#1}%
    \glshasattribute{#1}{glossnamefont}%
    {%
      \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
      \ifcsdef{\@glsxtr@attrval}%
      {%
        \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
      }%
      {%
        \GlossariesExtraWarning{Unknown control sequence name
        '\@glsxtr@attrval' supplied in glossnamefont attribute
        for entry '#1'. Reverting to default \string\glsnamefont}%
        \let\@glsxtr@glossnamefont\glsnamefont
      }%
    }%
    {\let\@glsxtr@glossnamefont\glsnamefont}%
    \glsifattribute{#1}{glossname}{firstuc}%
    {%
      \@glsxtr@accessdisplay
      {\@glsxtr@glossnamefont{\@Gls@entry@field{#1}{#2}}}%
      {#1}%
    }%
    {%
      \glsifattribute{#1}{glossname}{title}%
      {%
        \@glsxtr@do@titlecaps@warn
        \@glsxtr@accessdisplay
        {\@glsxtr@glossnamefont{\glsxtrfieldtitlecase{#1}{#2}}}%
        {#1}%
      }%
      {%
        \glsifattribute{#1}{glossname}{uc}%
        {%
          \letcs{\glo@name}{glo@\glsdetoklabel{#1}@#2}%
          \@glsxtr@accessdisplay
          {\@glsxtr@glossnamefont{\glsuppercase{\glo@name}}}%
          {#1}%
        }%
        {%
          \letcs{\glo@name}{glo@\glsdetoklabel{#1}@#2}%
          \@glsxtr@accessdisplay
          {\expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}}%
          {#1}%
```

```
          }%
        }%
      }%
```

Do post-name hook.

```
      \glsxtrpostnamehook{#1}%
    }%
  }
```

`\Glossentrynameother` Provide a command that works like `\Glossentryname` but accesses a different field (which must be supplied using its internal field label).

```
  \newrobustcmd*{\Glossentrynameother}[2]{%
    \@glsdoifexistsorwarn{#1}%
    {%
```

Accessibility support:

```
      \glsxtr@setaccessdisplay{#2}%
```

Set the abbreviation format:

```
      \glssetabbrvfmt{\glscategory{#1}}%
```

Do pre-name hook:

```
      \glsxtrprenamehook{#1}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
           '\@glsxtr@attrval' supplied in glossnamefont attribute
           for entry '#1'. Reverting to default \string\glsnamefont}%
          \let\@glsxtr@glossnamefont\glsnamefont
        }%
      }%
      {\let\@glsxtr@glossnamefont\glsnamefont}%
      \@glsxtr@accessdisplay
      {\@glsxtr@glossnamefont{\@Gls@entry@field{#1}{#2}}}%
      {#1}%
```

Do post-name hook.

```
      \glsxtrpostnamehook{#1}%
    }%
  }
```

`\GLOSSentrynameother` Provide a command that works like `\GLOSSentryname` but accesses a different field (which must be supplied using its internal field label).

```
  \newrobustcmd*{\GLOSSentrynameother}[2]{%
    \@glsdoifexistsorwarn{#1}%
    {%
```

Accessibility support:

```
\glsxtr@setaccessdisplay{#2}%
```

Set the abbreviation format:

```
\glssetabbrvfmt{\glscategory{#1}}%
```

Do pre-name hook:

```
\glsxtrprenamehook{#1}%
\glshasattribute{#1}{glossnamefont}%
{%
  \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
  \ifcsdef{\@glsxtr@attrval}%
  {%
    \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
  }%
  {%
    \GlossariesExtraWarning{Unknown control sequence name
    '\@glsxtr@attrval' supplied in glossnamefont attribute
    for entry '#1'. Reverting to default \string\glsnamefont}%
    \let\@glsxtr@glossnamefont\glsnamefont
  }%
}%
{\let\@glsxtr@glossnamefont\glsnamefont}%
\@glsxtr@accessdisplay
{\@glsxtr@glossnamefont{\glsuppercase{\@gls@entry@field{#1}{#2}}}}%
{#1}%
```

Do post-name hook.

```
  \glsxtrpostnamehook{#1}%
  }%
}
```

`\GlossEntryNameOther` Provide a command that works like `\GlossEntryName` but accesses a different field (which must be supplied using its internal field label).

```
\newrobustcmd*{\GlossEntryNameOther}[2]{%
  \@glsdoifexistsorwarn{#1}%
  {%
```

Accessibility support:

```
\glsxtr@setaccessdisplay{#2}%
```

Set the abbreviation format:

```
\glssetabbrvfmt{\glscategory{#1}}%
```

Do pre-name hook:

```
\glsxtrprenamehook{#1}%
\glshasattribute{#1}{glossnamefont}%
{%
  \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
  \ifcsdef{\@glsxtr@attrval}%
  {%
    \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
```

```
      }%
      {%
        \GlossariesExtraWarning{Unknown control sequence name
        '\@glsxtr@attrval' supplied in glossnamefont attribute
        for entry '#1'. Reverting to default \string\glsnamefont}%
        \let\@glsxtr@glossnamefont\glsnamefont
      }%
    }%
    {\let\@glsxtr@glossnamefont\glsnamefont}%
    \@glsxtr@accessdisplay
    {\@glsxtr@glossnamefont{\glsentrytitlecase{#1}{#2}}}%
    {#1}%
```

Do post-name hook.

```
    \glsxtrpostnamehook{#1}%
  }%
}
```

\if@glsxtr@format@override Determines if the format key should override the indexing attribute value.

```
\newif\if@glsxtr@format@override
\@glsxtr@format@overridefalse
```

If overriding is enabled, the \glshypernumber command will have to be redefined in the index to use \hyperpage instead.

XtrEnableIndexFormatOverride

```
\@ifpackageloaded{hyperref}
{
```

If hyperref's hyperindex option is on, then hyperref will automatically add \hyperpage, so don't add it.

```
  \ifHy@hyperindex
    \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
      \@glsxtr@format@overridetrue
      \appto\theindex{\let\glshypernumber\@firstofone}%
    }
  \else
    \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
      \@glsxtr@format@overridetrue
      \appto\theindex{\let\glshypernumber\hyperpage}%
    }
  \fi
}
{
  \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
    \@glsxtr@format@overridetrue
  }
}
\@onlypreamble\GlsXtrEnableIndexFormatOverride
```

312

`\glsxtrdoautoindexname`

```
\newcommand*{\glsxtrdoautoindexname}[2]{%
\glshasattribute{#1}{#2}%
{%
```

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.

```
\@glsxtr@autoindex@setname{#1}%
```

If the attribute value is simply "true" don't add an encap, otherwise use the value as the encap.

```
\protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{#2}}%
\if@glsxtr@format@override

  \ifx\@glsnumberformat\@glsxtr@defaultnumberformat
  \else
    \let\@glsxtr@attrval\@glsnumberformat
  \fi
\fi

\ifdefstring{\@glsxtr@attrval}{true}%
{}%
{\protected@eappto\@glo@name{\@glsxtr@autoindex@encap\@glsxtr@attrval}}%
\expandafter\glsxtrautoindex\expandafter{\@glo@name}%
}%
{}%
}
```

`\glsxtrautoindex`

```
\newcommand*{\glsxtrautoindex}{\index}
```

`\glsxtrautoindexesc`

```
\newcommand{\glsxtrautoindexesc}{%
\@gls@checkmkidxchars\@glo@sort
\@glsxtr@autoindex@doextra@esc\@glo@sort
}
```

`\@glsxtr@autoindex@setname`   Assign `\@glo@name` for use with indexname attribute.

```
\newcommand*{\@glsxtr@autoindex@setname}[1]{%
\protected@edef\@glo@name{\glsxtrautoindexentry{#1}}%
\glsxtrautoindexassignsort{\@glo@sort}{#1}%
\glsxtrautoindexesc
\epreto\@glo@name{\@glo@sort\@glsxtr@autoindex@at}%
}
```

`\glsxtrautoindexentry`   Command used for the actual part when auto-indexing.

```
\newcommand*{\glsxtrautoindexentry}[1]{\string\glsentryname{#1}}
```

`\glsxtrautoindexassignsort` Used to assign the sort value when auto-indexing.

```
\newcommand*{\glsxtrautoindexassignsort}[2]{%
  \glsletentryfield{#1}{#2}{sort}%
}
```

`\glsxtr@autoindex@doextra@esc`

```
\newcommand*{\@glsxtr@autoindex@doextra@esc}[1]{%
```

Escape the escape character unless it has already been escaped.

```
\ifx\@glsxtr@autoindex@esc\@gls@quotechar
\else
  \def\@gls@checkedmkidx{}%
  \edef\@@glsxtr@checkspch{%
    \noexpand\@glsxtr@autoindex@escquote\expandonce{#1}%
      \noexpand\@empty\@glsxtr@autoindex@esc\noexpand\@nnil
      \@glsxtr@autoindex@esc\noexpand\@empty\noexpand\@glsxtr@endescspch}%
  \@@glsxtr@checkspch
  \let#1\@gls@checkedmkidx\relax
\fi
```

Escape actual character unless it has already been escaped.

```
\ifx\@glsxtr@autoindex@at\@gls@actualchar
\else
  \def\@gls@checkedmkidx{}%
  \edef\@@glsxtr@checkspch{%
    \noexpand\@glsxtr@autoindex@escat\expandonce{#1}%
      \noexpand\@empty\@glsxtr@autoindex@at\noexpand\@nnil
      \@glsxtr@autoindex@at\noexpand\@empty\noexpand\@glsxtr@endescspch}%
  \@@glsxtr@checkspch
  \let#1\@gls@checkedmkidx\relax
\fi
```

Escape level character unless it has already been escaped.

```
\ifx\@glsxtr@autoindex@level\@gls@levelchar
\else
  \def\@gls@checkedmkidx{}%
  \edef\@@glsxtr@checkspch{%
    \noexpand\@glsxtr@autoindex@esclevel\expandonce{#1}%
      \noexpand\@empty\@glsxtr@autoindex@level\noexpand\@nnil
      \@glsxtr@autoindex@level\noexpand\@empty\noexpand\@glsxtr@endescspch}%
  \@@glsxtr@checkspch
  \let#1\@gls@checkedmkidx\relax
\fi
```

Escape encap character unless it has already been escaped.

```
\ifx\@glsxtr@autoindex@encap\@gls@encapchar
\else
  \def\@gls@checkedmkidx{}%
  \edef\@@glsxtr@checkspch{%
    \noexpand\@glsxtr@autoindex@escencap\expandonce{#1}%
      \noexpand\@empty\@glsxtr@autoindex@encap\noexpand\@nnil
```

314

```
                \@glsxtr@autoindex@encap\noexpand\@empty\noexpand\@glsxtr@endescspch}%
             \@@glsxtr@checkspch
             \let#1\@gls@checkedmkidx\relax
          \fi
       }
```

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.

\@glsxtr@autoindex@at   Actual character for use with \index.

```
\newcommand*{\@glsxtr@autoindex@at}{}
```

\GlsXtrSetActualChar   Set the actual character.

```
\newcommand*{\GlsXtrSetActualChar}[1]{%
  \gdef\@glsxtr@autoindex@at{#1}%
  \def\@glsxtr@autoindex@escat##1#1##2#1##3\@glsxtr@endescspch{%
    \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escat}{##1}{##2}{##3}%
  }%
}
\@onlypreamble\GlsXtrSetActualChar
\makeatother
\GlsXtrSetActualChar{@}
\makeatletter
```

\@glsxtr@autoindex@encap   Encap character for use with \index.

```
\newcommand*{\@glsxtr@autoindex@encap}{}
```

\GlsXtrSetEncapChar   Set the encap character.

```
\newcommand*{\GlsXtrSetEncapChar}[1]{%
  \gdef\@glsxtr@autoindex@encap{#1}%
  \def\@glsxtr@autoindex@escencap##1#1##2#1##3\@glsxtr@endescspch{%
    \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escencap}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetEncapChar{|}
\@onlypreamble\GlsXtrSetEncapChar
```

\@glsxtr@autoindex@level   Level character for use with \index.

```
\newcommand*{\@glsxtr@autoindex@level}{}
```

\GlsXtrSetLevelChar   Set the encap character.

```
\newcommand*{\GlsXtrSetLevelChar}[1]{%
  \gdef\@glsxtr@autoindex@level{#1}%
  \def\@glsxtr@autoindex@esclevel##1#1##2#1##3\@glsxtr@endescspch{%
    \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@esclevel}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetLevelChar{!}
\@onlypreamble\GlsXtrSetLevelChar
```

`\@glsxtr@autoindex@esc` Escape character for use with `\index`.

```
\newcommand*{\@glsxtr@autoindex@esc}{"}
```

`\GlsXtrSetEscChar` Set the escape character.

```
\newcommand*{\GlsXtrSetEscChar}[1]{%
  \gdef\@glsxtr@autoindex@esc{#1}%
  \def\@glsxtr@autoindex@escquote##1#1##2#1##3\@glsxtr@endescspch{%
    \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escquote}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetEscChar{"}
\@onlypreamble\GlsXtrSetEscChar
```

Set if defined. (For example, if doc package has been loaded.) Actual character `\actualchar`:

```
\ifdef\actualchar
  {\expandafter\GlsXtrSetActualChar\expandafter{\actualchar}}
  {}
```

Quote character `\quotechar`:

```
\ifdef\quotechar
  {\expandafter\GlsXtrSetEscChar\expandafter{\quotechar}}
  {}
```

Level character `\levelchar`:

```
\ifdef\levelchar
  {\expandafter\GlsXtrSetLevelChar\expandafter{\levelchar}}
  {}
```

Encap character `\encapchar`:

```
\ifdef\encapchar
  {\expandafter\GlsXtrSetEncapChar\expandafter{\encapchar}}
  {}
```

`\@glsxtr@gobbleto@endescspch`

```
\def\@glsxtr@gobbleto@endescspch#1\@glsxtr@endescspch{}
```

`\@@glsxtr@autoindex@escspch{⟨char⟩}{⟨cs⟩}{⟨pre⟩}{⟨mid⟩}`
`{⟨post⟩}`

`\@@glsxtr@autoindex@esc@spch`

```
\newcommand*{\@@glsxtr@autoindex@escspch}[5]{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#3}%
  \ifx\@nnil#3\relax
    \def\@@glsxtr@checkspch{\@glsxtr@gobbleto@endescspch#5\@glsxtr@endescspch}%
  \else
    \ifx\@nnil#4\relax
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
```

316

```
                    \def\@@glsxtr@checkspch{\@glsxtr@gobbleto@endescspch
                        #4#5\@glsxtr@endescspch}%
                 \else
                    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                     \@glsxtr@autoindex@esc#1}%
                    \def\@@glsxtr@checkspch{#2#5#1\@nnil#1\@glsxtr@endescspch}%
                 \fi
               \fi
               \@@glsxtr@checkspch
             }
```

\Glossentrydesc Redefine to set the abbreviation format and accessibility support.

```
               \renewcommand*{\Glossentrydesc}[1]{%
               \glsdoifexistsorwarn{#1}%
               {%
                 \glssetabbrvfmt{\glscategory{#1}}%
                 \Glsaccessdesc{#1}%
               }%
             }
```

\glossentrysymbol Redefine to set the format and accessibility support. Allow for the possibility
                   of being used in a section heading for standalone entry definitions.

```
               \renewcommand*{\glossentrysymbol}[1]{%
                 \glstexorpdfstring{\@glossentrysymbol{#1}}{\glsentrypdfsymbol{#1}}%
               }
```

\glsentrypdfsymbol May be redefined to a field that expands to a value that's more suitable for PDF
                    bookmarks.

```
               \newcommand{\glsentrypdfsymbol}[1]{\glsentrysymbol{#1}}
```

\@glossentrysymbol There are no case-changing attributes as it's less usual for symbols.

```
               \newrobustcmd*{\@glossentrysymbol}[1]{%
               \glsdoifexistsorwarn{#1}%
               {%
                 \begingroup
                   \glssetabbrvfmt{\glscategory{#1}}%
                   \glshasattribute{#1}{glosssymbolfont}%
                   {%
                     \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glosssymbolfont}}%
                     \ifcsdef{\@glsxtr@attrval}%
                     {%
                       \letcs\@glsxtr@glosssymbolfont{\@glsxtr@attrval}%
                     }%
                     {%
                       \GlossariesExtraWarning{Unknown control sequence name
                        `\@glsxtr@attrval' supplied in glosssymbolfont attribute
                        for entry `#1'. Ignoring}%
                       \let\@glsxtr@glosssymbolfont\@firstofone
                     }%
```

```
          }%
          {\let\@glsxtr@glosssymbolfont\@firstofone}%
          \@glsxtr@glosssymbolfont{\glsaccesssymbol{#1}}%
        \endgroup
      }%
    }
```

\Glossentrysymbol Redefine to set the abbreviation format and accessibility support.

```
    \renewcommand*{\Glossentrysymbol}[1]{%
      \glsdoifexistsorwarn{#1}%
      {%
        \glssetabbrvfmt{\glscategory{#1}}%
        \Glsaccesssymbol{#1}%
      }%
    }
```

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.

```
    \newcommand*{\GlsXtrEnableInitialTagging}{%
      \@ifstar\s@glsxtr@enabletagging\@glsxtr@enabletagging
    }
    \@onlypreamble\GlsXtrEnableInitialTagging
```

\@glsxtr@enabletagging Starred version undefines command.

```
    \newcommand*{\s@glsxtr@enabletagging}[2]{%
      \undef#2%
      \@glsxtr@enabletagging{#1}{#2}%
    }
```

\@glsxtr@enabletagging Internal command.

```
    \newcommand*{\@glsxtr@enabletagging}[2]{%
```

Set attributes for categories given in the first argument.

```
    \@for\@glsxtr@cat:=#1\do
    {%
      \ifdefempty\@glsxtr@cat
      {}%
      {\glssetcategoryattribute{\@glsxtr@cat}{tagging}{true}}%
    }%
    \newrobustcmd*#2[1]{##1}%
    \def\@glsxtr@taggingcs{#2}%
    \renewcommand*\@glsxtr@activate@initialtagging{%
      \let#2\@glsxtr@tag
    }%
    \ifundef\@gls@preglossaryhook
    {\GlossariesExtraWarning{Initial tagging requires at least
```

```
      glossaries.sty v4.19 to work correctly}}%
    {}%
 }
```

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.

`\mfu@checkword@do` If this command hasn't been defined, then we have pre v2.02 of mfirstuc

```
\ifundef\mfu@checkword@do
{
  \newcommand*{\mfu@checkword@do}[1]{%
  \ifdefstring{\mfu@checkword@arg}{#1}%
  {%
    \let\@mfu@domakefirstuc\@firstofone
    \listbreak
  }%
  {}%
  }
}
```

`\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.

```
\ifundef\mfu@checkword
{
  \newcommand{\@glsxtr@do@titlecaps@warn}{%
  \GlossariesExtraWarning{mfirstuc.sty too old. Title Caps
    support not available}%
```

One warning should suffice.

```
    \let\@glsxtr@do@titlecaps@warn\relax
  }
}
{
  \renewcommand*{\mfu@checkword}[1]{%
    \def\mfu@checkword@arg{#1}%
    \let\@mfu@domakefirstuc\makefirstuc
    \forlistloop\mfu@checkword@do\@mfu@nocaplist
  }
 }
}
{}% no patch required
```

`\@glsxtr@do@titlecaps@warn` Do warning if title case not supported.

```
\newcommand*{\@glsxtr@do@titlecaps@warn}{}
```

`xtr@activate@initialtagging` Used in `\printglossary` but at least v4.19 of glossaries required.

```
\newcommand*\@glsxtr@activate@initialtagging{}
```

`\@glsxtr@tag` Definition of tagging command when used in glossary.

319

```
\newrobustcmd*{\@glsxtr@tag}[1]{%
  \glsifattribute{\glscurrententrylabel}{tagging}{true}%
  {\glsxtrtagfont{#1}}{#1}%
}
```

\glsxtrtagfont Used in the glossary.

```
\newcommand*{\glsxtrtagfont}[1]{\underline{#1}}
```

\@gls@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.

```
\ifdef\@gls@preglossaryhook
{
  \renewcommand*{\@gls@preglossaryhook}{%
    \@glsxtr@activate@initialtagging
```

Since the glossaries are automatically scoped, \@glsxtr@org@postdescription shouldn't already be defined, but check anyway just as a precautionary measure.

```
    \ifundef\@glsxtr@org@postdescription
    {%
      \let\@glsxtr@org@postdescription\glspostdescription
      \renewcommand*{\glspostdescription}{%
        \ifglsentryexists{\glscurrententrylabel}%
        {%
          \glsxtrpostdescription
          \@glsxtr@org@postdescription
        }%
        {}%
      }%
    }%
    {}%
```

Enable the options used by \@@glsxtrp:

```
    \glossxtrsetpopts
  }%
}
{}
```

\glsxtrpostdescription 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 predefined styles that don't include it.

```
\newcommand*{\glsxtrpostdescription}{%
  \csuse{glsxtrpostdesc\glscategory{\glscurrententrylabel}}%
}
```

\glsxtrpostdescgeneral

```
\newcommand*{\glsxtrpostdescgeneral}{}
```

\glsxtrpostdescterm This is redundant as it doesn't match any common categories. `\newterm` sets the category to index.

```
\newcommand*{\glsxtrpostdescterm}{}
```

\glsxtrpostdescacronym

```
\newcommand*{\glsxtrpostdescacronym}{}
```

\glsxtrpostdescabbreviation

```
\newcommand*{\glsxtrpostdescabbreviation}{}
```

\glsdefpostdesc Provide a convenient command for defining the post-description hook for the given category.

```
\newcommand*{\glsdefpostdesc}[2]{%
  \csdef{glsxtrpostdesc#1}{#2}%
}
```

\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.

```
\renewcommand*{\glspostlinkhook}{%
\ifglsentryexists{\glslabel}{\glsxtrpostlinkhook}{}%
}
```

\glsxtrpostlinkhook The entry label should already be stored in `\glslabel` by `\@gls@link`.

```
\newcommand*{\glsxtrpostlinkhook}{%
 \glsxtrdiscardperiod{\glslabel}%
 {\glsxtrpostlinkendsentence}%
 {\glsxtrifcustomdiscardperiod
  {\glsxtrifperiod{\glsxtrpostlinkendsentence}{\glsxtrpostlink}}%
  {\glsxtrpostlink}%
 }%
}
```

\glsxtrifcustomdiscardperiod Allow user to provide a custom check. Should expand to #2 if no check is required otherwise expand to #1.

```
\newcommand*{\glsxtrifcustomdiscardperiod}[2]{#2}
```

\glsxtrpostlink

```
\newcommand*{\glsxtrpostlink}{%
 \csuse{glsxtrpostlink\glscategory{\glslabel}}%
}
```

\glsdefpostlink Provide a convenient command for defining the post-link hook for the given category. Doesn't allow an empty argument (which) would overwrite `\glsxtrpostlink`.

```
\newcommand*{\glsdefpostlink}[2]{%
```

`\ifthenelse` is used to ensure that the expanded value is tested. (The category label must be fully expandable.)

```
    \ifthenelse{\equal{#1}{}}%
    {\PackageError{glossaries-extra}
      {Invalid empty category label in \string\glsdefpostlink}{}}%
    {\csdef{glsxtrpostlink#1}{#2}}%
  }
```

\glspretopostlink Similar to the above but prepend.

```
    \newcommand*{\glspretopostlink}[2]{%
```

`\ifthenelse` is used to ensure that the expanded value is tested. (The category label must be fully expandable.)

```
    \ifthenelse{\equal{#1}{}}%
    {\PackageError{glossaries-extra}
      {Invalid empty category label in \string\glspretopostlink}{}}%
    {%
      \ifcsundef{glsxtrpostlink#1}
      {\csdef{glsxtrpostlink#1}{#2}}%
      {\cspreto{glsxtrpostlink#1}{#2}}%
    }%
  }
```

\glsapptopostlink Similar to the above but append.

```
    \newcommand*{\glsapptopostlink}[2]{%
```

`\ifthenelse` is used to ensure that the expanded value is tested. (The category label must be fully expandable.)

```
    \ifthenelse{\equal{#1}{}}%
    {\PackageError{glossaries-extra}
      {Invalid empty category label in \string\glspretopostlink}{}}%
    {%
      \ifcsundef{glsxtrpostlink#1}
      {\csdef{glsxtrpostlink#1}{#2}}%
      {\csappto{glsxtrpostlink#1}{#2}}%
    }%
  }
```

\glsxtrpostlinkendsentence Done by `\glsxtrpostlinkhook` if a full stop is discarded.

```
    \newcommand*{\glsxtrpostlinkendsentence}{%
    \ifcsdef{glsxtrpostlink\glscategory{\glslabel}}
    {%
      \csuse{glsxtrpostlink\glscategory{\glslabel}}%
```

Put the full stop back.

```
      .\spacefactor\sfcode`\. \relax
    }%
    {%
```

322

Assume the full stop was discarded because the entry ends with a period, so adjust the spacefactor.

```
    \spacefactor\sfcode'\. \relax
  }%
 }
```

rpostlinkAddDescOnFirstUse 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.

```
\newcommand*{\glsxtrpostlinkAddDescOnFirstUse}{%
  \glsxtrifwasfirstuse{\glsxtrgenentrytextfmt{ }%
   \glsxtrparen{\glsaccessfmtdesc{}{\glsxtrgenentrytextfmt}{\glslabel}}}{}%
}
```

postlinkAddSymbolOnFirstUse 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.

```
\newcommand*{\glsxtrpostlinkAddSymbolOnFirstUse}{%
  \glsxtrifwasfirstuse
  {%
    \ifglshassymbol{\glslabel}%
    {\glsxtrgenentrytextfmt{ }%
     \glsxtrparen{\glsaccessfmtsymbol{}{\glsxtrgenentrytextfmt}{\glslabel}}}%
    {}%
  }%
  {}%
}
```

linkAddSymbolDescOnFirstUse Provide a command for appending the symbol (if defined) and description in parentheses on first use, for the convenience of users wanting to add this to the post link hook.

```
\newcommand*{\glsxtrpostlinkAddSymbolDescOnFirstUse}{%
  \glsxtrifwasfirstuse
  {%
    \glsxtrgenentrytextfmt{ }\glsxtrparen
    {%
      \ifglshassymbol{\glslabel}%
      {\glsaccessfmtsymbol{}{\glsxtrgenentrytextfmt}{\glslabel}%
       \expandafter\glsxtrgenentrytextfmt\expandafter{\glsxtrpostlinkSymbolDescSep}}%
      {}%
      \glsaccessfmtdesc{}{\glsxtrgenentrytextfmt}{\glslabel}%
    }%
  }%
  {}%
}
```

\glsxtrpostlinkSymbolDescSep Separator used in the above

```
\newcommand*{\glsxtrpostlinkSymbolDescSep}{, }
```

```
\newcommand*{\glsxtrdiscardperiodretainfirstuse}[3]{%
\glsxtrifwassubsequentorshort{\glsxtrifperiod{#2}{#3}}{#3}%
}
```

\glsxtrdiscardperiod 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).

```
\newcommand*{\glsxtrdiscardperiod}[3]{%
\glsifattribute{#1}{retainfirstuseperiod}{true}%
{\glsxtrdiscardperiodretainfirstuse{#1}{#2}{#3}}%
{%
  \glsifattribute{#1}{discardperiod}{true}%
  {%
    \glsifplural
    {%
      \glsifattribute{#1}{pluraldiscardperiod}{true}%
      {\glsxtrifperiod{#2}{#3}}%
      {#3}%
    }%
    {%
      \glsxtrifperiod{#2}{#3}%
    }%
  }%
  {#3}%
}%
}
```

\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

```
\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 short-
hands!) This doesn't allow for punctuation marks made up from multiple char-
acters (such as '').

```
\newcommand*{\glsxtr@punclist}{.,:;?!}
```

\glsxtraddpunctuationmark Add character to punctuation list.

```
\newcommand*{\glsxtraddpunctuationmark}[1]{\appto\glsxtr@punclist{#1}}
```

\glsxtrsetpunctuationmarks Reset the punctuation list.

```
\newcommand*{\glsxtrsetpunctuationmarks}[1]{\def\glsxtr@punclist{#1}}
```

\glsxtrifnextpunc

> \glsxtrifnextpunc{⟨*true part*⟩}{⟨*false part*⟩}

Test if this is followed by a punctuation mark. (Adapted from \new@ifnextchar.)

```
\newcommand*{\glsxtrifnextpunc}[2]{%
  \def\reserved@a{#1}%
  \def\reserved@b{#2}%
  \futurelet\@glspunc@token\glsxtr@ifnextpunc
}
```

\glsxtr@ifnextpunc

```
\newcommand*{\glsxtr@ifnextpunc}{%
\glsxtr@ifpunctoken{\@glspunc@token}{\let\reserved@b\reserved@a}{}%
\reserved@b
}
```

\glsxtr@ifpunctoken Test if the token given in the first argument is in the punctuation list.

```
\newcommand*{\glsxtr@ifpunctoken}[1]{%
  \expandafter\@glsxtr@ifpunctoken\expandafter#1\glsxtr@punclist\@nnil
}
```

\@glsxtr@ifpunctoken

```
\def\@glsxtr@ifpunctoken#1#2{%
  \let\reserved@d=#2%
  \ifx\reserved@d\@nnil
    \let\glsxtr@next\@glsxtr@notfoundinlist
  \else
    \ifx#1\reserved@d
     \let\glsxtr@next\@glsxtr@foundinlist
    \else
     \let\glsxtr@next\@glsxtr@ifpunctoken
    \fi
  \fi
  \glsxtr@next#1%
}
```

\@glsxtr@foundinlist

```
\def\@glsxtr@foundinlist#1\@nnil{\@firstoftwo}
```

\@glsxtr@notfoundinlist

```
\def\@glsxtr@notfoundinlist#1{\@secondoftwo}
```

\glsxtrdopostpunc

> \glsxtrdopostpunc{⟨*code*⟩}

If this is followed by a punctuation character, do ⟨*code*⟩ after the character otherwise do ⟨*code*⟩ before whatever comes next.

```
\newrobustcmd{\glsxtrdopostpunc}[1]{%
```

```
    \glsxtrifnextpunc{\@glsxtr@swaptwo{#1}}{#1}%
  }
```

```
    \newcommand{\@glsxtr@swaptwo}[2]{#2#1}
```

1.7   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, it needs to be applied by \newabbreviation.

```
    \define@key{glsxtrabbrv}{category}{%
      \protected@edef\glscategorylabel{#1}%
    }
```

The shortplural and longplural are parsed separately, so are now in another key family. Save the short plural form. This may be needed before the entry is defined.

```
    \define@key{glsxtrabbrvpl}{shortplural}{%
      \def\@gls@shortpl{#1}%
    }
```

Similarly for the long plural form.

```
    \define@key{glsxtrabbrvpl}{longplural}{%
      \def\@gls@longpl{#1}%
    }
```

   Token registers for the short plural and long plural, provided for use in the abbreviation style definitions.

```
    \newtoks\glsshortpltok
```

```
    \newtoks\glslongpltok
```

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.

```
    \newcommand*{\@glsxtr@insertdots}[2]{%
      \def#1{}%
      \@glsxtr@insert@dots#1#2\@nnil
    }
```

\@glsxtr@insert@dots

```
\newcommand*{\@glsxtr@insert@dots}[2]{%
 \ifx\@nnil#2\relax
  \let\@glsxtr@insert@dots@next\@gobble
 \else
  \ifx\relax#2\relax
  \else
    \appto#1{#2.}%
  \fi
  \let\@glsxtr@insert@dots@next\@glsxtr@insert@dots
 \fi
 \@glsxtr@insert@dots@next#1%
}
```

Similarly provide a way of replacing spaces with \glsxtrwordsep, which first needs to be defined:

\glsxtrwordsep

```
\newcommand*{\glsxtrwordsep}{\glsxtrgenentrytextfmt{ }}
```

\glsxtrwordsephyphen

```
\newcommand*{\glsxtrwordsephyphen}{\glsxtrgenentrytextfmt{-}}
```

Each word is marked with

\glsxtrword

```
\newcommand*{\glsxtrword}[1]{\glsxtrgenentrytextfmt{#1}}
```

\@glsxtr@markwordseps

```
\newcommand*{\@glsxtr@markwordseps}[2]{%
 \def#1{}%
 \@glsxtr@mark@wordseps#1#2 \@nnil
}
```

\@glsxtr@mark@wordseps

```
\def\@glsxtr@mark@wordseps#1#2 #3{%
 \ifdefempty{#1}%
 {\def#1{\protect\glsxtrword{#2}}}%
 {\appto#1{\protect\glsxtrwordsep\protect\glsxtrword{#2}}}%
 \ifx\@nnil#3\relax
  \let\@glsxtr@mark@wordseps@next\relax
 \else
  \def\@glsxtr@mark@wordseps@next{%
    \@glsxtr@mark@wordseps#1#3}%
 \fi
 \@glsxtr@mark@wordseps@next
}
```

`\newabbreviation` Define a new generic abbreviation.

```
\newcommand*{\newabbreviation}[4][]{%
  \glsxtr@newabbreviation{#1}{#2}{#3}{#4}%
}
```

`\glsxtr@newabbreviation` Internal macro. (bib2gls has an option that needs to temporarily redefine `\newabbreviation`. This is just makes it easier to save and restore the original definition.)

```
\newcommand*{\glsxtr@newabbreviation}[4]{%
  \glskeylisttok{#1}%
  \glslabeltok{#2}%
  \glsshorttok{#3}%
  \glslongtok{#4}%
```

Save the original short and long values (before attribute settings modify them).

```
  \def\glsxtrorgshort{#3}%
  \def\glsxtrorglong{#4}%

  \def\glsxtrorgkeylist{#1}%
```

Provide extra settings for hooks. Make sure to append a comma if this hook is changed.

```
  \def\ExtraCustomAbbreviationFields{}%
```

Initialise accessibility settings if required.

```
  \@gls@initaccesskeys
```

Get the category.

```
  \def\glscategorylabel{abbreviation}%
```

Ignore the shortplural and longplural keys.

```
  \setkeys*{glsxtrabbrv}{#1}%
```

Save remaining keys, just in case any hook also uses `\setkeys`

```
  \let\@glsxtrabbrv@rmkeys\XKV@rm
```

Set the abbreviation style.

```
  \ifcsdef{@glsabbrv@current@\glscategorylabel}%
  {%
```

Warning should already have been issued.

```
    \let\@glsxtr@orgwarndep\GlsXtrWarnDeprecatedAbbrStyle
    \let\GlsXtrWarnDeprecatedAbbrStyle\@gobbletwo
    \glsxtr@applyabbrvstyle{\csname @glsabbrv@current@\glscategorylabel\endcsname}%
    \let\GlsXtrWarnDeprecatedAbbrStyle\@glsxtr@orgwarndep
  }%
  {%
```

If no style has been associated with this category, fallback on the style for the abbreviation category.

```
    \glsxtr@applyabbrvstyle{\@glsabbrv@current@abbreviation}%
  }%
```

Set the default long plural

```
\def\@gls@longpl{#4\glspluralsuffix}%
```

Has the markwords attribute been set?

```
\glsifcategoryattribute{\glscategorylabel}{markwords}{true}%
{%
  \@glsxtr@markwordseps\@gls@long{#4}%
```

Update \glslongtok.

```
\expandafter\glslongtok\expandafter{\@gls@long}%
```

Mark this entry as having a description with formatting.

```
\glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
}%
{}%
```

Has the markshortwords attribute been set? (Not compatible with insertdots.)

```
\let\@glsxtr@if@markshortwords\@secondoftwo
\glsifcategoryattribute{\glscategorylabel}{markshortwords}{true}%
{%
```

Don't mark words until the default plural has been obtained.

```
  \let\@glsxtr@if@markshortwords\@firstoftwo
  \def\@gls@short{#3}%
}%
{%
```

Has the insertdots attribute been set?

```
  \glsifcategoryattribute{\glscategorylabel}{insertdots}{true}%
  {%
    \@glsxtr@insertdots\@gls@short{#3}%

    \appto\@gls@short{\@}%
  }%
  {\def\@gls@short{#3}}%
}%
```

Has the aposplural attribute been set? (Not compatible with noshortplural.)

```
\glsifcategoryattribute{\glscategorylabel}{aposplural}{true}%
{%
  \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
    '\abbrvpluralsuffix}%
}%
{%
```

Has the noshortplural attribute been set?

```
\glsifcategoryattribute{\glscategorylabel}{noshortplural}{true}%
{%
  \let\@gls@shortpl\@gls@short
}%
{%
  \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
    \abbrvpluralsuffix}%
```

```
    }%
  }%
  \@glsxtr@if@markshortwords
  {%
    \expandafter\@glsxtr@markwordseps\expandafter\@gls@short
    \expandafter{\@gls@short}%
  }%
  {}%
```

Update `\glsshorttok`:

```
  \expandafter\glsshorttok\expandafter{\@gls@short}%
```

Hook for further customisation if required:

```
  \glsxtrnewabbrevpresetkeyhook{#1}{#2}{#3}%
```

Get the short and long plurals provided by user in optional argument to override defaults, if necessary. Save the default short plural.

```
  \let\@gls@default@shortpl\@gls@shortpl
  \let\XKV@rm\@glsxtrabbrv@rmkeys
  \setrmkeys*{glsxtrabbrvpl}%
```

Update `\glskeylisttok` so that it only has the remaining keys.

```
  \expandafter\glskeylisttok\expandafter{\XKV@rm}%
```

Save in case required.

```
  \let\@gls@org@longpl\@gls@longpl
  \let\@gls@org@shortpl\@gls@shortpl
```

Has the markwords attribute been set?

```
  \glsifcategoryattribute{\glscategorylabel}{markwords}{true}%
  {%
    \expandafter\@glsxtr@markwordseps\expandafter\@gls@longpl\expandafter
    {\@gls@longpl}%
  }%
  {}%
```

Has the markshortwords attribute been set?

```
  \@glsxtr@if@markshortwords
  {%
    \expandafter\@glsxtr@markwordseps\expandafter\@gls@shortpl
    \expandafter{\@gls@shortpl}%
  }%
  {%
```

Has the insertdots attribute been set?

```
    \ifx\@gls@default@shortpl\@gls@shortpl
    \else
      \glsifcategoryattribute{\glscategorylabel}{insertdots}{true}%
      {%
        \expandafter\@glsxtr@insertdots\expandafter\@gls@shortpl
          \expandafter{\@gls@shortpl}%
        \appto\@gls@shortpl{\@}%
      }%
```

```
      {}%
    \fi
  }%
```

Set the plural token registers so the values can be accessed by the abbreviation styles.

```
\expandafter\glsshortpltok\expandafter{\@gls@shortpl}%
\expandafter\glslongpltok\expandafter{\@gls@longpl}%
```

Hook for accessibility support (does nothing if glossaries-accsupp hasn't been loaded).

```
\@gls@setup@default@access
```

Do any extra setup provided by hook:

```
\newabbreviationhook
```

Define this entry:

```
\protected@edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type={\glsxtrabbrvtype},%
    category={\glscategorylabel},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshortpltok},%
    long={\the\glslongtok},%
    longplural={\the\glslongpltok},%
    name={\the\glsshorttok},%
    \CustomAbbreviationFields,%
```

Hook may override abbreviation style default settings.

```
    \ExtraCustomAbbreviationFields
```

Any explicit fields set in the optional argument override all other settings, except for the ones that have already been processed.

```
    \the\glskeylisttok
  }%
}%
\@do@newglossaryentry
```

Obtain the type and add it to the list of abbreviations.

```
\@glsxtr@addabbreviationlist{\glsentrytype{\the\glslabeltok}}%
```

Exclude name, first, firstpl, text and plural fields from inner fmt as they include formatting commands. The description may also need adding, depending on the style.

```
\glsexclapplyinnerfmtfield{\the\glslabeltok}{first}%
\glsexclapplyinnerfmtfield{\the\glslabeltok}{firstpl}%
\glsexclapplyinnerfmtfield{\the\glslabeltok}{text}%
\glsexclapplyinnerfmtfield{\the\glslabeltok}{plural}%
\glsexclapplyinnerfmtfield{\the\glslabeltok}{name}%
\GlsXtrPostNewAbbreviation
}
```

glsxtrnewabbrevpresetkeyhook Hook for extra stuff in \newabbreviation

```
\newcommand*{\glsxtrnewabbrevpresetkeyhook}[3]{}
```

\GlsXtrPostNewAbbreviation Hook used by abbreviation styles.

```
\newcommand*{\GlsXtrPostNewAbbreviation}{}
```

\newabbreviationhook Hook for use with \newabbreviation.

```
\newcommand*{\newabbreviationhook}{}
```

\CustomAbbreviationFields

```
\newcommand*{\CustomAbbreviationFields}{}
```

\glsxtrparen For the parenthetical styles.

```
\newcommand*{\glsxtrparen}[1]{%
 \glsxtrgenentrytextfmt{(}#1\glsxtrgenentrytextfmt{)}}
```

\glsxtrfullformat Full format without case change.

```
\newcommand*{\glsxtrfullformat}[2]{%
  \ifglsxtrinsertinside
   \glsfirstlongfont{\glsaccessfmtlong{#2}{\glsxtrgenentrytextfmt}{#1}}%
  \else
   \glsfirstlongfont{\glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
   \glsxtrgenentrytextfmt{#2}%
  \fi
  \glsxtrfullsep{#1}%
  \glsxtrparen{\protect\glsfirstabbrvfont
    {\glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}}%
}
```

\Glsxtrfullformat Full format with case change.

```
\newcommand*{\Glsxtrfullformat}[2]{%
  \ifglsxtrinsertinside
   \glsfirstlongfont{\Glsaccessfmtlong{#2}{\glsxtrgenentrytextfmt}{#1}}%
  \else
   \glsfirstlongfont{\Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
   \glsxtrgenentrytextfmt{#2}%
  \fi
  \glsxtrfullsep{#1}%
  \glsxtrparen{\protect\glsfirstabbrvfont
    {\glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}}%
}
\glsmfuaddmap{\glsxtrfullformat}{\Glsxtrfullformat}
```

\GLSxtrfullformat Full format with all caps.

```
\newcommand*{\GLSxtrfullformat}[2]{%
  \ifglsxtrinsertinside
   \glsfirstlongfont{\GLSaccessfmtlong{#2}{\glsxtrgenentrytextfmt}{#1}}%
  \else
```

```
      \glsfirstlongfont{\GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsuppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\protect\glsfirstabbrvfont
      {\GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}}%
  }
  \glsmfublocker{\GLSxtrfullformat}
```

\glsxtrfullplformat Plural full format without case change.

```
  \newcommand*{\glsxtrfullplformat}[2]{%
    \ifglsxtrinsertinside
    \glsfirstlongfont{\glsaccessfmtlongpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
    \glsfirstlongfont{\glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \glsxtrgenentrytextfmt{#2}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\protect\glsfirstabbrvfont
      {\glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}}%
  }
```

\Glsxtrfullplformat Plural full format with case change.

```
  \newcommand*{\Glsxtrfullplformat}[2]{%
    \ifglsxtrinsertinside
    \glsfirstlongfont{\Glsaccessfmtlongpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
    \glsfirstlongfont{\Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \glsxtrgenentrytextfmt{#2}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\protect\glsfirstabbrvfont
      {\glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}}%
  }
  \glsmfuaddmap{\glsxtrfullplformat}{\Glsxtrfullplformat}
```

\GLSxtrfullplformat Full format with all caps.

```
  \newcommand*{\GLSxtrfullplformat}[2]{%
    \ifglsxtrinsertinside
    \glsfirstlongfont{\GLSaccessfmtlongpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
    \glsfirstlongfont{\GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \glsuppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\protect\glsfirstabbrvfont
      {\GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}}%
  }
  \glsmfublocker{\GLSxtrfullplformat}
```

\GLSxtr@fullformat@fallback Fallback for custom styles that don't implement all caps version.

```
\newcommand*{\GLSxtr@fullformat@fallback}[2]{%
  \glsuppercase{\glsxtrfullformat{##1}{##2}}%
}%
```

\GLSxtr@fullplformat@fallback Fallback for custom styles that don't implement all caps version.

```
\newcommand*{\GLSxtr@fullplformat@fallback}[2]{%
  \glsuppercase{\glsxtrfullplformat{##1}{##2}}%
}%
```

\glsxtrfullsep Separator used by full format is a space by default. The argument is the entry's label.

```
\newcommand*{\glsxtrfullsep}[1]{\glsxtrgenentrytextfmt{ }}
```

In-line formats in case first use isn't compatible with \glsentryfull (for example, first use suppresses the long form or uses a footnote).

\glsxtrinlinefullformat Full format without case change.

```
\newcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}
```

\Glsxtrinlinefullformat Full format with case change.

```
\newcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}
```

\GLSxtrinlinefullformat Full format with all caps.

```
\newcommand*{\GLSxtrinlinefullformat}{\GLSxtrfullformat}
```

\glsxtrfullplformat Plural full format without case change.

```
\newcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}
```

\Glsxtrinlinefullplformat Plural full format with case change.

```
\newcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}
```

\GLSxtrinlinefullplformat Full format with all caps.

```
\newcommand*{\GLSxtrinlinefullplformat}{\GLSxtrfullplformat}
```

Redefine \glsentryfull etc to use the inline format. Since these commands are supposed to be expandable, they can only use the currently applied style. If there are mixed styles, you'll need to use the \glsxtrfull set of commands instead. If expandable sentence case is required, use \MFUsentencecase on the non-case-change version.

\glsentryfull

```
\renewcommand*{\glsentryfull}[1]{\glsxtrinlinefullformat{#1}{}}
```

\Glsentryfull

```
\renewcommand*{\Glsentryfull}[1]{\Glsxtrinlinefullformat{#1}{}}
\glsmfuaddmap{\glsentryfull}{\Glsentryfull}
```

334

\glsentryfullpl

> `\renewcommand*{\glsentryfullpl}[1]{\glsxtrinlinefullplformat{#1}{}}`

\Glsentryfullpl

> `\renewcommand*{\Glsentryfullpl}[1]{\Glsxtrinlinefullplformat{#1}{}}`
> `\glsmfuaddmap{\glsentryfullpl}{\Glsentryfullpl}`

\glsfirstabbrvfont   Font changing command used for the abbreviation on first use or in the full format.

> `\newcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{#1}}`

\glsfirstinnerfmtabbrvfont   Include inner formatting command.

> `\newrobustcmd*{\glsfirstinnerfmtabbrvfont}[1]{%`
> `\glsfirstabbrvfont{\glsxtrgenentrytextfmt{#1}}%`
> `}`

\glsfirstxpabbrvfont   Expand to appropriate formatting command.

> `\newcommand*{\glsfirstxpabbrvfont}[2]{%`
> `\glsifcategoryattributetrue{#2}{markshortwords}%`
> `{\protect\glsfirstabbrvfont{#1}}%`
> `{\glsfirstinnerfmtabbrvfont{#1}}%`
> `}`

\glsfirstabbrvdefaultfont   Font changing command used for the abbreviation on first use or in the full format.

> `\newcommand*{\glsfirstabbrvdefaultfont}[1]{\glsabbrvdefaultfont{#1}}`

\glsabbrvfont   Font changing command used for the abbreviation on subsequent use. This is redefined by the abbreviation styles, as appropriate.

> `\newcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{#1}}`

\glsinnerfmtabbrvfont   Include inner formatting command.

> `\newrobustcmd*{\glsinnerfmtabbrvfont}[1]{%`
> `\glsabbrvfont{\glsxtrgenentrytextfmt{#1}}%`
> `}`

\glsxpabbrvfont   Expand to appropriate formatting command.

> `\newcommand*{\glsxpabbrvfont}[2]{%`
> `\glsifcategoryattributetrue{#2}{markshortwords}%`
> `{\protect\glsabbrvfont{#1}}%`
> `{\glsinnerfmtabbrvfont{#1}}%`
> `}`

\glsabbrvdefaultfont

> `\newcommand*{\glsabbrvdefaultfont}[1]{#1}`

\glslongfont   Font changing command used for the long form in commands like `\glsxtrlong`.

> `\newcommand*{\glslongfont}[1]{\glslongdefaultfont{#1}}`

`\glsinnerfmtlongfont` Include inner formatting command.

```
\newrobustcmd*{\glsinnerfmtlongfont}[1]{%
 \glslongfont{\glsxtrgenentrytextfmt{#1}}%
}
```

`\glsxplongfont` Expand to appropriate formatting command.

```
\newcommand*{\glsxplongfont}[2]{%
 \glsifcategoryattributetrue{#2}{markwords}%
 {\protect\glslongfont{#1}}%
 {\glsinnerfmtlongfont{#1}}%
}
```

`\glslongdefaultfont` Default font changing command used for the long form in commands like `\glsxtrlong`.

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

`\glsfirstlongfont` Font changing command used for the long form on first use or in the full format.

```
\newcommand*{\glsfirstlongfont}[1]{\glslongfont{#1}}
```

`\glsfirstinnerfmtlongfont` Include inner formatting command.

```
\newrobustcmd*{\glsfirstinnerfmtlongfont}[1]{%
 \glsfirstlongfont{\glsxtrgenentrytextfmt{#1}}%
}
```

`\glsfirstxplongfont` Expand to appropriate formatting command.

```
\newcommand*{\glsfirstxplongfont}[2]{%
 \glsifcategoryattributetrue{#2}{markwords}%
 {\protect\glsfirstlongfont{#1}}%
 {\glsfirstinnerfmtlongfont{#1}}%
}
```

`\glsfirstlongdefaultfont`

```
\newcommand*{\glsfirstlongdefaultfont}[1]{\glslongdefaultfont{#1}}
```

`\glsxtrabbrvpluralsuffix` Default plural suffix. Allow an alternative default suffix for abbreviations.

```
\newcommand*{\glsxtrabbrvpluralsuffix}{\glspluralsuffix}
```

`\abbrvpluralsuffix` Default plural suffix.

```
\newcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}
```

`\glsxtrrevert` Provide a way to counteract the abbreviation font.

```
\newcommand*{\glsxtrrevert}[1]{\glsxtrdefaultrevert{#1}}%
```

`\glsxtrdefaultrevert` The default simply does its argument.

```
\newcommand*{\glsxtrdefaultrevert}[1]{#1}%
```

`\glsxtrfull` Full form (no case-change).

```
\newrobustcmd*{\glsxtrfull}{\@gls@hyp@opt\ns@glsxtrfull}
\newcommand*\ns@glsxtrfull[2][]{%
  \new@ifnextchar[{\@glsxtr@full{#1}{#2}}%
                 {\@glsxtr@full{#1}{#2}[]}%
}
```

`\@glsxtr@full` Low-level macro:

```
\def\@glsxtr@full#1#2[#3]{%
  \def\glsxtrcurrentfield{}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
\@glsxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
  \glssetabbrvfmt{\glscategory{#2}}%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \let\glsxtrifwasglslike\@secondoftwo
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@firstofthree
  \glsxtrfullsaveinsert{#2}{#3}%
```

The innertextformat support should be provided within the inline command.

```
  \def\glscustomtext{\glsxtrinlinefullformat{#2}{#3}}%
```

What should `\glsxtrifwasfirstuse` 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.

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

`\glsxtrsetupfulldefs`

```
\newcommand*{\glsxtrsetupfulldefs}{%
  \let\glsxtrifwasfirstuse\@firstoftwo
}
```

`\Glsxtrfull` Full form (first letter uppercase).

```
\newrobustcmd*{\Glsxtrfull}{\@gls@hyp@opt\ns@Glsxtrfull}
\newcommand*\ns@Glsxtrfull[2][]{%
  \new@ifnextchar[{\@Glsxtr@full{#1}{#2}}%
                 {\@Glsxtr@full{#1}{#2}[]}%
}
\glsmfuaddmap{\glsxtrfull}{\Glsxtrfull}
```

`\@Glsxtr@full` Low-level macro:

```
\def\@Glsxtr@full#1#2[#3]{%
  \def\glsxtrcurrentfield{}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \glsxtrfullsaveinsert{#2}{#3}%
```

The innertextformat support should be provided within the inline command.

```
    \def\glscustomtext{\Glsxtrinlinefullformat{#2}{#3}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

`\GLSxtrfull` Full form (all uppercase).

```
\newrobustcmd*{\GLSxtrfull}{\@gls@hyp@opt\ns@GLSxtrfull}
\newcommand*\ns@GLSxtrfull[2][]{%
  \new@ifnextchar[{\@GLSxtr@full{#1}{#2}}%
                 {\@GLSxtr@full{#1}{#2}[]}%
}
```

`\@GLSxtr@full` Low-level macro:

```
\def\@GLSxtr@full#1#2[#3]{%
  \def\glsxtrcurrentfield{}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \glsxtrfullsaveinsert{#2}{#3}%
```

The innertextformat support should be provided within the inline command.

```
    \def\glscustomtext{\GLSxtrinlinefullformat{#2}{#3}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\glsmfublocker{\GLSxtrfull}
```

`\glsxtrfullpl` Plural full form (no case-change).

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

```
\newcommand*\ns@glsxtrfullpl[2][]{%
  \new@ifnextchar[{\@glsxtr@fullpl{#1}{#2}}%
                  {\@glsxtr@fullpl{#1}{#2}[]}%
}
```

\@glsxtr@fullpl Low-level macro:

```
\def\@glsxtr@fullpl#1#2[#3]{%
  \def\glsxtrcurrentfield{}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
\@glsxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
  \glssetabbrvfmt{\glscategory{#2}}%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \let\glsxtrifwasglslike\@secondoftwo
  \let\glsifplural\@firstoftwo
  \let\glscapscase\@firstofthree
  \glsxtrfullsaveinsert{#2}{#3}%
```

The innertextformat support should be provided within the inline command.

```
  \def\glscustomtext{\glsxtrinlinefullplformat{#2}{#3}}%
  \glsxtrsetupfulldefs
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
```

\Glsxtrfullpl Plural full form (first letter uppercase).

```
\newrobustcmd*{\Glsxtrfullpl}{\@gls@hyp@opt\ns@Glsxtrfullpl}
\newcommand*\ns@Glsxtrfullpl[2][]{%
  \new@ifnextchar[{\@Glsxtr@fullpl{#1}{#2}}%
                  {\@Glsxtr@fullpl{#1}{#2}[]}%
}
\glsmfuaddmap{\glsxtrfullpl}{\Glsxtrfullpl}
```

\@Glsxtr@fullpl Low-level macro:

```
\def\@Glsxtr@fullpl#1#2[#3]{%
  \def\glsxtrcurrentfield{}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
\@glsxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
  \glssetabbrvfmt{\glscategory{#2}}%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

```
        \let\glsxtrifwasglslike\@secondoftwo
        \let\glsifplural\@firstoftwo
        \let\glscapscase\@secondofthree
        \glsxtrfullsaveinsert{#2}{#3}%
```

The innertextformat support should be provided within the inline command.

```
        \def\glscustomtext{\Glsxtrinlinefullplformat{#2}{#3}}%
        \glsxtrsetupfulldefs
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%
      \glspostlinkhook
    }
```

\GLSxtrfullpl Plural full form (all upper case).

```
        \newrobustcmd*{\GLSxtrfullpl}{\@gls@hyp@opt\ns@GLSxtrfullpl}
        \newcommand*\ns@GLSxtrfullpl[2][]{%
          \new@ifnextchar[{\@GLSxtr@fullpl{#1}{#2}}%
                          {\@GLSxtr@fullpl{#1}{#2}[]}%
        }
        \glsmfublocker{\GLSxtrfullpl}
```

\@GLSxtr@fullpl Low-level macro:

```
        \def\@GLSxtr@fullpl#1#2[#3]{%
          \def\glsxtrcurrentfield{}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
        \@glsxtr@record{#1}{#2}{glslink}%
        \glsdoifexists{#2}%
        {%
          \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
          \let\glsxtrifwasglslike\@secondoftwo
          \let\glsifplural\@firstoftwo
          \let\glscapscase\@thirdofthree
          \glsxtrfullsaveinsert{#2}{#3}%
```

The innertextformat support should be provided within the inline command.

```
        \def\glscustomtext{%
          \GLSxtrinlinefullplformat{#2}{#3}}%
        \glsxtrsetupfulldefs
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%
      \glspostlinkhook
    }
```

The short and long forms work in a similar way to acronyms.

\glsxtrshort

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

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

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

Read in the final optional argument:

```
\def\@glsxtrshort#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
```

Need to make sure \glsabbrvfont is set correctly.

```
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \glsxtrsaveinsert{#2}{#3}%
    \def\glscustomtext{%
      \glsxtrshortformat{#2}{#3}{\glsabbrvfont}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

\Glsxtrshort

```
\newrobustcmd*{\Glsxtrshort}{\@gls@hyp@opt\ns@Glsxtrshort}
\glsmfuaddmap{\glsxtrshort}{\Glsxtrshort}
```

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

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

Read in the final optional argument:

```
\def\@Glsxtrshort#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
  \@glsxtr@record{#1}{#2}{glslink}%
```

```
      \glsdoifexists{#2}%
      {%
        \glssetabbrvfmt{\glscategory{#2}}%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsxtrifwasglslike\@secondoftwo
        \let\glsxtrifwasfirstuse\@secondoftwo
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@secondofthree
        \glsxtrsaveinsert{#2}{#3}%
        \def\glscustomtext{%
          \Glsxtrshortformat{#2}{#3}{\glsabbrvfont}%
        }%
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%
      \glspostlinkhook
    }
```

\GLSxtrshort

```
    \newrobustcmd*{\GLSxtrshort}{\@gls@hyp@opt\ns@GLSxtrshort}
    \glsmfublocker{\GLSxtrshort}
```

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

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

Read in the final optional argument:

```
    \def\@GLSxtrshort#1#2[#3]{%
      \def\glsxtrcurrentfield{short}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
      \@glsxtr@record{#1}{#2}{glslink}%
      \glsdoifexists{#2}%
      {%
        \glssetabbrvfmt{\glscategory{#2}}%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsxtrifwasglslike\@secondoftwo
        \let\glsxtrifwasfirstuse\@secondoftwo
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@thirdofthree
        \glsxtrsaveinsert{#2}{#3}%
        \def\glscustomtext{\GLSxtrshortformat{#2}{#3}{\glsabbrvfont}}%
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%
      \glspostlinkhook
    }
```

`\glsxtrsetlongfirstuse` Assigns `\glsxtrifwasfirstuse` for the long commands. The argument is the entry label. This now makes commands such as `\glsxtrlong` simulate first use.

```
\newcommand{\glsxtrsetlongfirstuse}[1]{%
  \let\glsxtrifwasfirstuse\@firstoftwo
}
```

`\glsxtrlong`

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

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

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

Read in the final optional argument:

```
\def\@glsxtrlong#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
\@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \glsxtrsetlongfirstuse{#2}%
    \glsxtrsaveinsert{#2}{#3}%
    \def\glscustomtext{%
      \glsxtrlongformat{#2}{#3}{\glslongfont}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

`\Glsxtrlong`

```
\newrobustcmd*{\Glsxtrlong}{\@gls@hyp@opt\ns@Glsxtrlong}
\glsmfuaddmap{\glsxtrlong}{\Glsxtrlong}
```

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

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

Read in the final optional argument:

```
\def\@Glsxtrlong#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
    \@glsxtr@record{#1}{#2}{glslink}%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasglslike\@secondoftwo
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@secondofthree
      \glsxtrsetlongfirstuse{#2}%
      \glsxtrsaveinsert{#2}{#3}%
      \def\glscustomtext{%
        \Glsxtrlongformat{#2}{#3}{\glslongfont}%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
  }
```

`\GLSxtrlong`

```
    \newrobustcmd*{\GLSxtrlong}{\@gls@hyp@opt\ns@GLSxtrlong}
    \glsmfublocker{\GLSxtrlong}
```

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

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

Read in the final optional argument:

```
    \def\@GLSxtrlong#1#2[#3]{%
      \def\glsxtrcurrentfield{long}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
    \@glsxtr@record{#1}{#2}{glslink}%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasglslike\@secondoftwo
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@thirdofthree
      \glsxtrsetlongfirstuse{#2}%
      \glsxtrsaveinsert{#2}{#3}%
```

```
    \def\glscustomtext{%
     \GLSxtrlongformat{#2}{#3}{\glslongfont}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

Plural short forms:

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

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

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

Read in the final optional argument:

```
\def\@glsxtrshortpl#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \glsxtrsaveinsert{#2}{#3}%
    \def\glscustomtext{%
      \glsxtrshortplformat{#2}{#3}{\glsabbrvfont}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

```
\newrobustcmd*{\Glsxtrshortpl}{\@gls@hyp@opt\ns@Glsxtrshortpl}
\glsmfuaddmap{\glsxtrshortpl}{\Glsxtrshortpl}
```

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

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

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

Read in the final optional argument:

```
\def\@Glsxtrshortpl#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
    \@glsxtr@record{#1}{#2}{glslink}%
    \glsdoifexists{#2}%
    {%
      \glssetabbrvfmt{\glscategory{#2}}%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasglslike\@secondoftwo
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@secondofthree
      \glsxtrsaveinsert{#2}{#3}%
      \def\glscustomtext{%
        \Glsxtrshortplformat{#2}{#3}{\glsabbrvfont}%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
  }
```

\GLSxtrshortpl

```
  \newrobustcmd*{\GLSxtrshortpl}{\@gls@hyp@opt\ns@GLSxtrshortpl}
  \glsmfublocker{\GLSxtrshortpl}
```

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

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

Read in the final optional argument:

```
\def\@GLSxtrshortpl#1#2[#3]{%
  \def\glsxtrcurrentfield{short}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
    \@glsxtr@record{#1}{#2}{glslink}%
    \glsdoifexists{#2}%
    {%
      \glssetabbrvfmt{\glscategory{#2}}%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasglslike\@secondoftwo
```

```
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \glsxtrsaveinsert{#2}{#3}%
    \def\glscustomtext{%
        \GLSxtrshortplformat{#2}{#3}{\glsabbrvfont}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
 }
```

Plural long forms:

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

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

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

Read in the final optional argument:

```
\def\@glsxtrlongpl#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
    \@glsxtr@record{#1}{#2}{glslink}%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasglslike\@secondoftwo
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@firstofthree
      \glsxtrsetlongfirstuse{#2}%
      \glsxtrsaveinsert{#2}{#3}%
      \def\glscustomtext{%
        \glsxtrlongplformat{#2}{#3}{\glslongfont}%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
```

```
\newrobustcmd*{\Glsxtrlongpl}{\@gls@hyp@opt\ns@Glsxtrlongpl}
\glsmfuaddmap{\glsxtrlongpl}{\Glsxtrlongpl}
```

347

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

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

Read in the final optional argument:

```
\def\@Glsxtrlongpl#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasglslike\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \glsxtrsetlongfirstuse{#2}%
    \glsxtrsaveinsert{#2}{#3}%
    \def\glscustomtext{%
      \Glsxtrlongplformat{#2}{#3}{\glslongfont}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
```

\GLSxtrlongpl

```
\newrobustcmd*{\GLSxtrlongpl}{\@gls@hyp@opt\ns@GLSxtrlongpl}
\glsmfublocker{\GLSxtrlongpl}
```

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

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

Read in the final optional argument:

```
\def\@GLSxtrlongpl#1#2[#3]{%
  \def\glsxtrcurrentfield{long}%
```

If the record option has been used, the information needs to be written to the aux file regardless of whether the entry exists (unless indexing has been switched off).

```
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

```
        \let\glsxtrifwasglslike\@secondoftwo
        \let\glsifplural\@firstoftwo
        \let\glscapscase\@thirdofthree
        \glsxtrsetlongfirstuse{#2}%
        \glsxtrsaveinsert{#2}{#3}%
        \def\glscustomtext{%
            \GLSxtrlongplformat{#2}{#3}{\glslongfont}%
        }%
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
  }
```

\glssetabbrvfmt Set the current format for the given category (or the abbreviation category if
unset).

```
\newcommand*{\glssetabbrvfmt}[1]{%
  \ifcsdef{@glsabbrv@current@#1}%
  {\glsxtr@applyabbrvfmt{\csname @glsabbrv@current@#1\endcsname}}%
  {\glsxtr@applyabbrvfmt{\@glsabbrv@current@abbreviation}}%
}
```

\glsuseabbrvfont Provide a way to use the abbreviation font for a given category for arbitrary
text.

```
\newrobustcmd*{\glsuseabbrvfont}[2]{{\glssetabbrvfmt{#2}\glsabbrvfont{#1}}}
```

\glsuselongfont Provide a way to use the long font for a given category for arbitrary text.

```
\newrobustcmd*{\glsuselongfont}[2]{{\glssetabbrvfmt{#2}\glslongfont{#1}}}
```

\glsxtrgenabbrvfmt Similar to \glsgenacfmt, but for abbreviations. The expansion is to ensure
that \glsinsert is expanded before being passed to \glsfmtfield etc in the
event that an inner command is being used (which typically signifies a complex
formatting command such as those provided by soul).

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

Subsequent use:

```
      \glsifplural
      {%
```

Subsequent plural form:

```
        \glscapscase
        {%
```

Subsequent plural form, don't adjust case:

```
          \expandafter\glsxtrsubsequentplfmt\expandafter\glslabel
            \expandafter{\glsinsert}%
        }%
        {%
```

349

Subsequent plural form, make first letter upper case:

```
\expandafter\Glsxtrsubsequentplfmt\expandafter\glslabel
  \expandafter{\glsinsert}%
}%
{%
```

Subsequent plural form, all caps:

```
\expandafter\GLSxtrsubsequentplfmt\expandafter\glslabel
  \expandafter{\glsinsert}%
}%
}%
{%
```

Subsequent singular form

```
\glscapscase
{%
```

Subsequent singular form, don't adjust case:

```
\expandafter\glsxtrsubsequentfmt\expandafter\glslabel
  \expandafter{\glsinsert}%
}%
{%
```

Subsequent singular form, make first letter upper case:

```
\expandafter\Glsxtrsubsequentfmt\expandafter
  \glslabel\expandafter{\glsinsert}%
}%
{%
```

Subsequent singular form, all caps:

```
\expandafter\GLSxtrsubsequentfmt\expandafter
  \glslabel\expandafter{\glsinsert}%
}%
}%
}%
{%
```

First use:

```
\glsifplural
{%
```

First use plural form:

```
\glscapscase
{%
```

First use plural form, don't adjust case:

```
\expandafter\glsxtrfullplformat\expandafter\glslabel
  \expandafter{\glsinsert}%
}%
{%
```

First use plural form, make first letter upper case:

```
\expandafter\Glsxtrfullplformat\expandafter\glslabel
```

```
                  \expandafter{\glsinsert}%
              }%
              {%
```

First use plural form, all caps:

```
                  \expandafter\GLSxtrfullplformat\expandafter\glslabel
                    \expandafter{\glsinsert}%
              }%
            }%
            {%
```

First use singular form

```
                \glscapscase
                {%
```

First use singular form, don't adjust case:

```
                  \expandafter\glsxtrfullformat\expandafter\glslabel
                    \expandafter{\glsinsert}%
              }%
              {%
```

First use singular form, make first letter upper case:

```
                  \expandafter\Glsxtrfullformat\expandafter\glslabel
                   \expandafter{\glsinsert}%
              }%
              {%
```

First use singular form, all caps:

```
                  \expandafter\GLSxtrfullformat\expandafter\glslabel
                   \expandafter{\glsinsert}%
              }%
            }%
          }%
        }%
        {%
```

Custom text provided in `\glsdisp`. (The insert is most likely to be empty at this point.) Any inner formatting can be supplied with the custom text.

```
          \glscustomtext
        }%
     }
```

`\glsxtrsubsequentfmt` Subsequent use format (singular no case change).

```
   \newcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsifattribute{#1}{markshortwords}{true}%
    {%
      \ifglsxtrinsertinside
        \glsabbrvfont{\glsaccessshort{#1}\glsxtrgenentrytextfmt{#2}}%
      \else
        \glsabbrvfont{\glsaccessshort{#1}}\glsxtrgenentrytextfmt{#2}%
      \fi
    }%
```

```
                        {%
                          \ifglsxtrinsertinside
                           \glsabbrvfont{\glsaccessfmtshort{#2}{\glsxtrgenentrytextfmt}{#1}}%
                          \else
                           \glsabbrvfont{\glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
                           \glsxtrgenentrytextfmt{#2}%
                          \fi
                        }%
                      }
                      \let\glsxtrdefaultsubsequentfmt\glsxtrsubsequentfmt
```

\glsxtrsubsequentplfmt Subsequent use format (plural no case change).

```
                      \newcommand*{\glsxtrsubsequentplfmt}[2]{%
                        \glsifattribute{#1}{markshortwords}{true}%
                        {%
                          \ifglsxtrinsertinside
                           \glsabbrvfont{\glsaccessshortpl{#1}\glsxtrgenentrytextfmt{#2}}%
                          \else
                           \glsabbrvfont{\glsaccessshortpl{#1}}\glsxtrgenentrytextfmt{#2}%
                          \fi
                        }%
                        {%
                          \ifglsxtrinsertinside
                           \glsabbrvfont{\glsaccessfmtshortpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
                          \else
                           \glsabbrvfont{\glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
                           \glsxtrgenentrytextfmt{#2}%
                          \fi
                        }%
                      }
                      \let\glsxtrdefaultsubsequentplfmt\glsxtrsubsequentplfmt
```

\Glsxtrsubsequentfmt Subsequent use format (singular, first letter uppercase).

```
                      \newcommand*{\Glsxtrsubsequentfmt}[2]{%
                        \glsifattribute{#1}{markshortwords}{true}%
                        {%
                          \ifglsxtrinsertinside
                           \glsabbrvfont{\Glsaccessshort{#1}\glsxtrgenentrytextfmt{#2}}%
                          \else
                           \glsabbrvfont{\Glsaccessshort{#1}}\glsxtrgenentrytextfmt{#2}%
                          \fi
                        }%
                        {%
                          \ifglsxtrinsertinside
                           \glsabbrvfont{\Glsaccessfmtshort{#2}{\glsxtrgenentrytextfmt}{#1}}%
                          \else
                           \glsabbrvfont{\Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
                           \glsxtrgenentrytextfmt{#2}%
                          \fi
                        }%
```

```
                }
                \let\Glsxtrdefaultsubsequentfmt\Glsxtrsubsequentfmt
                \glsmfuaddmap{\glsxtrsubsequentfmt}{\Glsxtrsubsequentfmt}
```

\Glsxtrsubsequentplfmt Subsequent use format (plural, first letter uppercase).
```
                \newcommand*{\Glsxtrsubsequentplfmt}[2]{%
                  \glsifattribute{#1}{markshortwords}{true}%
                  {%
                    \ifglsxtrinsertinside
                      \glsabbrvfont{\Glsaccessshortpl{#1}\glsxtrgenentrytextfmt{#2}}%
                    \else
                      \glsabbrvfont{\Glsaccessshortpl{#1}}\glsxtrgenentrytextfmt{#2}%
                    \fi
                  }%
                  {%
                    \ifglsxtrinsertinside
                     \glsabbrvfont{\Glsaccessfmtshortpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
                    \else
                     \glsabbrvfont{\Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
                     \glsxtrgenentrytextfmt{#2}%
                    \fi
                  }%
                }
                \let\Glsxtrdefaultsubsequentplfmt\Glsxtrsubsequentplfmt
                \glsmfuaddmap{\glsxtrsubsequentplfmt}{\Glsxtrsubsequentplfmt}
```

\GLSxtrsubsequentfmt Subsequent use format (singular, all caps).
```
                \newcommand*{\GLSxtrsubsequentfmt}[2]{%
                  \glsifattribute{#1}{markshortwords}{true}%
                  {%
                    \ifglsxtrinsertinside
                      \glsabbrvfont{\GLSaccessshort{#1}%
                        \glsuppercase{\glsxtrgenentrytextfmt{#2}}}%
                    \else
                      \glsabbrvfont{\GLSaccessshort{#1}}%
                        \glsuppercase{\glsxtrgenentrytextfmt{#2}}%
                    \fi
                  }%
                  {%
                    \ifglsxtrinsertinside
                     \glsabbrvfont{\GLSaccessfmtshort{#2}{\glsxtrgenentrytextfmt}{#1}}%
                    \else
                     \glsabbrvfont{\GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
                     \glsuppercase{\glsxtrgenentrytextfmt{#2}}%
                    \fi
                  }%
                }
                \glsmfublocker{\GLSxtrsubsequentfmt}
                \let\GLSxtrdefaultsubsequentfmt\GLSxtrsubsequentfmt
```

`\GLSxtrsubsequentplfmt` Subsequent use format (plural, all caps).

```
\newcommand*{\GLSxtrsubsequentplfmt}[2]{%
 \glsifattribute{#1}{markshortwords}{true}%
 {%
   \ifglsxtrinsertinside
     \glsabbrvfont{\GLSaccessshortpl{#1}%
      \glsuppercase{\glsxtrgenentrytextfmt{#2}}}%
   \else
     \glsabbrvfont{\GLSaccessshortpl{#1}}%
      \glsuppercase{\glsxtrgenentrytextfmt{#2}}%
   \fi
 }%
 {%
   \ifglsxtrinsertinside
    \glsabbrvfont{\GLSaccessfmtshortpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
   \else
    \glsabbrvfont{\GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \glsuppercase{\glsxtrgenentrytextfmt{#2}}%
   \fi
 }%
}
\glsmfublocker{\GLSxtrsubsequentplfmt}
\let\GLSxtrdefaultsubsequentplfmt\GLSxtrsubsequentplfmt
```

1.7.1 Abbreviation Styles Setup

`\setabbreviationstyle`

```
\newcommand*{\setabbreviationstyle}[2][abbreviation]{%
\ifcsundef{@glsabbrv@dispstyle@setup@#2}
{%
  \PackageError{glossaries-extra}{Undefined abbreviation style '#2'}{}%
}%
{%
```

Have abbreviations already been defined for this category?

```
\ifcsstring{@glsabbrv@current@#1}{#2}%
{%
```

Style already set.

```
}%
{%
  \def\@glsxtr@dostylewarn{}%
  \glsforeachincategory{#1}{\@gls@type}{\@gls@label}%
  {%
    \def\@glsxtr@dostylewarn{\GlossariesWarning{Abbreviation
      style has been switched \MessageBreak
      for category '#1', \MessageBreak
      but there have already been entries \MessageBreak
      defined for this category. Unwanted \MessageBreak
      side-effects may result}}%
```

```
      \@endfortrue
    }%
    \@glsxtr@dostylewarn
```

Set up the style for the given category.

```
      \csdef{@glsabbrv@current@#1}{#2}%
      \protected@edef\glscategorylabel{#1}%
      \glsxtr@applyabbrvstyle{#2}%
    }%
  }%
}
```

\glsxtr@applyabbrvstyle Apply the abbreviation style without existence check.

```
\newcommand*{\glsxtr@applyabbrvstyle}[1]{%
  \csuse{@glsabbrv@dispstyle@setup@#1}%
  \csuse{@glsabbrv@dispstyle@fmts@#1}%
}
```

\glsxtr@applyabbrvfmt Only apply the style formats.

```
\newcommand*{\glsxtr@applyabbrvfmt}[1]{%
  \csuse{@glsabbrv@dispstyle@fmts@#1}%
}
```

\glsxtrsetcomplexstyle Identify an entry as having a complex abbreviation style that doesn't work with \GLSfirst etc. The argument is the entry label. The second argument should be numeric: 1 (all caps doesn't work), 2 (all caps and insert don't work), 3 (insert doesn't work).

```
\newcommand*{\glsxtrsetcomplexstyle}[2]{%
  \csdef{@glsxtr@has@complexstyle@#1}{#2}%
}
```

tr@do@ifcomplexstyle@allcaps Do second argument if entry identified by first argument has a problem with all caps. Does nothing otherwise.

```
\newcommand*{\glsxtr@do@ifcomplexstyle@allcaps}[2]{%
  \ifcsdef{@glsxtr@has@complexstyle@#1}%
  {%
    \ifnum\csuse{@glsxtr@has@complexstyle@#1}<1
    \else
     \ifnum\csuse{@glsxtr@has@complexstyle@#1}<3
       #2%
     \fi
    \fi
  }%
  {}%
}
```

tr@do@ifcomplexstyle@insert Do second argument if entry identified by first argument has a problem with the insert argument. Does nothing otherwise.

```
\newcommand*{\glsxtr@do@ifcomplexstyle@insert}[2]{%
```

```
                          \ifcsdef{@glsxtr@has@complexstyle@#1}%
                          {%
                            \ifnum\csuse{@glsxtr@has@complexstyle@#1}<2
                            \else
                              #2%
                            \fi
                          }%
                          {}%
                        }
```

esAbbrStyleTooComplexWarning

```
                        \newcommand*{\GlossariesAbbrStyleTooComplexWarning}[2]{%
                          \GlossariesExtraWarning{Abbreviation style used by '#1' too complex #2}%
                        }
```

\glsxtr@check@complexstyle The first argument is the label the second is the insert.

```
                        \newcommand*{\glsxtr@check@complexstyle}[2]{%
                        \ifx\glscapscase\@thirdofthree
                        \glsxtr@do@ifcomplexstyle@allcaps{#1}%
                        {%
                          \glsxtrifwasfirstuse
                          {%
                            \glsifplural
                            {%
                              \GlossariesAbbrStyleTooComplexWarning{#1}{for \string\GLSfirstplural.
                                Use \string\GLSpl{#1} or \string\GLSxtrfullpl{#1} instead.
                                Switching off all-caps%
                              }%
                            }%
                            {%
                              \GlossariesAbbrStyleTooComplexWarning{#1}{for \string\GLSfirst.
                                Use \string\GLS{#1} or \string\GLSxtrfull{#1} instead.
                                Switching off all-caps%
                              }%
                            }%
                          }%
                          {%
                            \glsifplural
                            {%
                              \GlossariesAbbrStyleTooComplexWarning{#1}{for \string\GLSplural.
                                Use \string\GLSpl{#1} or \string\GLSxtrshortpl{#1} instead.
                                Switching off all-caps%
                              }%
                            }%
                            {%
                              \GlossariesAbbrStyleTooComplexWarning{#1}{for \string\GLStext.
                                Use \string\GLS{#1} or \string\GLSxtrshort{#1} instead.
                                Switching off all-caps%
                              }%
                            }%
                          }%
```

```
      }%
      \let\glscapscase\@firstofthree
    }%
    \fi
    \ifstrempty{#2}{}%
    {%
      \glsxtr@do@ifcomplexstyle@insert{#1}%
      {%
        \GlossariesAbbrStyleTooComplexWarning{#1}%
        {to support insert argument with commands like \string\glsfirst\space or
         \string\glstext. Unexpected results may occur. Use commands
         like \string\gls\space or \string\glsxtrshort\space instead}%
      }%
    }%
  }
```

\newabbreviationstyle This is different from \newacronymstyle. The first argument is the label, the
                       second argument sets the information required when defining the new abbrevia-
                       tion and the third argument sets the commands used to display the full format.

```
\newcommand*{\newabbreviationstyle}[3]{%
  \ifcsdef{@glsabbrv@dispstyle@setup@#1}
  {%
    \PackageError{glossaries-extra}{Abbreviation style '#1' already
     defined}{}%
  }%
  {%
    \csdef{@glsabbrv@dispstyle@setup@#1}{%
```

Initialise hook to do nothing. The style may change this.

```
      \renewcommand*{\GlsXtrPostNewAbbreviation}{}%
      #2}%
    \csdef{@glsabbrv@dispstyle@fmts@#1}{%
```

Assume in-line form is the same as first use. The style may change this.

```
      \renewcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}%
      \renewcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}%
      \renewcommand*{\GLSxtrinlinefullformat}{\GLSxtrfullformat}%
      \renewcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}%
      \renewcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}%
      \renewcommand*{\GLSxtrinlinefullplformat}{\GLSxtrfullplformat}%
```

In the event that some custom styles predate the introduction of the all caps
versions, provide default definitions:

```
      \renewcommand*{\GLSxtrfullformat}{\GLSxtr@fullformat@fallback}%
      \renewcommand*{\GLSxtrfullplformat}{\GLSxtr@fullplformat@fallback}%
```

Reset \glsxtrsubsequentfmt etc in case a style changes this.

```
      \let\glsxtrsubsequentfmt\glsxtrdefaultsubsequentfmt
      \let\glsxtrsubsequentplfmt\glsxtrdefaultsubsequentplfmt
      \let\Glsxtrsubsequentfmt\Glsxtrdefaultsubsequentfmt
```

```
                  \let\Glsxtrsubsequentplfmt\Glsxtrdefaultsubsequentplfmt
                  \let\GLSxtrsubsequentfmt\GLSxtrdefaultsubsequentfmt
                  \let\GLSxtrsubsequentplfmt\GLSxtrdefaultsubsequentplfmt
                  #3}%
                }%
              }
```

\renewabbreviationstyle

```
              \newcommand*{\renewabbreviationstyle}[3]{%
                \ifcsundef{@glsabbrv@dispstyle@setup@#1}
                {%
                  \PackageError{glossaries-extra}{Abbreviation style '#1' not defined}{}%
                }%
                {%
                  \csdef{@glsabbrv@dispstyle@setup@#1}{%
```

Initialise hook to do nothing. The style may change this.

```
                    \renewcommand*{\GlsXtrPostNewAbbreviation}{}%
                    #2}%
                  \csdef{@glsabbrv@dispstyle@fmts@#1}{%
```

Assume in-line form is the same as first use. The style may change this.

```
                    \renewcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}%
                    \renewcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}%
                    \renewcommand*{\GLSxtrinlinefullformat}{\GLSxtrfullformat}%
                    \renewcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}%
                    \renewcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}%
                    \renewcommand*{\GLSxtrinlinefullplformat}{\GLSxtrfullplformat}%
```

In the event that some custom styles predate the introduction of the all caps
versions, provide default definitions:

```
                    \renewcommand*{\GLSxtrfullformat}{\GLSxtr@fullformat@fallback}%
                    \renewcommand*{\GLSxtrfullplformat}{\GLSxtr@fullplformat@fallback}%
```

Reset \glsxtrsubsequentfmt etc in case a style changes this.

```
                    \let\glsxtrsubsequentfmt\glsxtrdefaultsubsequentfmt
                    \let\glsxtrsubsequentplfmt\glsxtrdefaultsubsequentplfmt
                    \let\Glsxtrsubsequentfmt\Glsxtrdefaultsubsequentfmt
                    \let\Glsxtrsubsequentplfmt\Glsxtrdefaultsubsequentplfmt
                    \let\GLSxtrsubsequentfmt\GLSxtrdefaultsubsequentfmt
                    \let\GLSxtrsubsequentplfmt\GLSxtrdefaultsubsequentplfmt
                    #3}%
                }%
              }
```

\letabbreviationstyle Define a synonym for an abbreviation style. The first argument is the new name.
The second argument is the original style's name.

```
              \newcommand*{\letabbreviationstyle}[2]{%
                \csletcs{@glsabbrv@dispstyle@setup@#1}{@glsabbrv@dispstyle@setup@#2}%
                \csletcs{@glsabbrv@dispstyle@fmts@#1}{@glsabbrv@dispstyle@fmts@#2}%
              }
```

@glsxtr@deprecated@abbrstyle

> \@glsxtr@deprecated@abbrstyle{⟨*old-name*⟩}{⟨*new-name*⟩}

Define a synonym for a deprecated abbreviation style.

```
\newcommand*{\@glsxtr@deprecated@abbrstyle}[2]{%
  \csdef{@glsabbrv@dispstyle@setup@#1}{%
    \GlsXtrWarnDeprecatedAbbrStyle{#1}{#2}%
    \csuse{@glsabbrv@dispstyle@setup@#2}%
  }%
  \csletcs{@glsabbrv@dispstyle@fmts@#1}{@glsabbrv@dispstyle@fmts@#2}%
}
```

\GlsXtrWarnDeprecatedAbbrStyle Generate warning for deprecated style use.

```
\newcommand*{\GlsXtrWarnDeprecatedAbbrStyle}[2]{%
  \GlossariesExtraWarning{Deprecated abbreviation style name '#1',
  use '#2' instead}%
}
```

\GlsXtrUseAbbrStyleSetup

```
\newcommand*{\GlsXtrUseAbbrStyleSetup}[1]{%
  \ifcsundef{@glsabbrv@dispstyle@setup@#1}%
  {%
    \PackageError{glossaries-extra}%
    {Unknown abbreviation style definitions '#1'}{}%
  }%
  {%
    \csname @glsabbrv@dispstyle@setup@#1\endcsname
  }%
}
```

\GlsXtrUseAbbrStyleFmts

```
\newcommand*{\GlsXtrUseAbbrStyleFmts}[1]{%
  \ifcsundef{@glsabbrv@dispstyle@fmts@#1}%
  {%
    \PackageError{glossaries-extra}%
    {Unknown abbreviation style formats '#1'}{}%
  }%
  {%
    \csname @glsabbrv@dispstyle@fmts@#1\endcsname
  }%
}
```

1.7.2 Predefined Styles

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 sentence case versions to work correctly, \glsxtrfullformat

359

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.

v1.49 has introduced innertextformat for formatting commands that need access to the actual text (which is normally too deeply embedded). The styles have been modified to allow for this. The all caps versions also now need to be implemented within the styles as again the text is now to deeply embedded for the case change to otherwise work.

\ifglsxtrinsertinside Switch to determine if the insert text should be inside or outside the font changing command. The default is outside.

```
\newif\ifglsxtrinsertinside
\glsxtrinsertinsidefalse
```

The abbreviation styles are now defined in the file `glossaries-extra-abbrstyles.def`, which needs to be input here:

```
\input{glossaries-extra-abbrstyles.def}
```

1.8   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 \glsxtrshort 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 unless mfirstuc 2.08+ is detected, so the label can be protected from case change with textcase's \NoCaseChange. This means that we don't have a problem provided the page style uses \MakeTextUppercase, but the default heading page style uses \MakeUppercase. (With newer versions of mfirstuc, exclusions are used to protect labels).

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. This only needs to be done with old versions of mfirstuc.

`\markright` Save original definition:

```
\let\@glsxtr@org@markright\markright
```

Redefine (grouping not added in case it interferes with the original code):

```
\renewcommand*{\markright}[1]{%
 \glsxtrmarkhook
 \@glsxtr@org@markright{\@glsxtrinmark#1\@glsxtrnotinmark}%
 \glsxtrrestoremarkhook
 }
```

`\markboth` Save original definition:

```
\let\@glsxtr@org@markboth\markboth
```

Redefine (grouping not added in case it interferes with the original code):

```
\renewcommand*{\markboth}[2]{%
 \glsxtrmarkhook
 \@glsxtr@org@markboth
   {\@glsxtrinmark#1\@glsxtrnotinmark}%
   {\@glsxtrinmark#2\@glsxtrnotinmark}%
 \glsxtrrestoremarkhook
 }
```

Also do this for `\@starttoc`

`\@starttoc` Save original definition:

```
\let\@glsxtr@org@@starttoc\@starttoc
```

Redefine:

```
\renewcommand*{\@starttoc}[1]{%
 \let\glsxtrifintoc\@firstoftwo
 \glsxtrmarkhook
 \@glsxtrinmark
 \@glsxtr@org@@starttoc{#1}%
 \@glsxtrnotinmark
 \glsxtrrestoremarkhook
 \let\glsxtrifintoc\@secondoftwo
 }
```

If this causes a problem provide a simple way of switching back to the original definitions:

`\glsxtrRevertMarks`

```
\newcommand*{\glsxtrRevertMarks}{%
 \let\markright\@glsxtr@org@markright
 \let\markboth\@glsxtr@org@markboth
 \let\@starttoc\@glsxtr@org@@starttoc
 }
```

`\glsxtrRevertTocMarks` Just restores `\@starttoc`.

```
\newcommand*{\glsxtrRevertTocMarks}{%
 \let\@starttoc\@glsxtr@org@@starttoc
 }
```

```
\newcommand*{\glsxtrifinmark}[2]{#2}
```

```
\newrobustcmd*{\@glsxtrinmark}{%
  \let\glsxtrifinmark\@firstoftwo
}
```

```
\newrobustcmd*{\@glsxtrnotinmark}{%
  \let\glsxtrifinmark\@secondoftwo
}
```

```
\newcommand*{\glsxtrtitleorpdforheading}[3]{%
 \glsxtrifinmark{#3}{\glstexorpdfstring{#1}{#2}}}
```

This will require \GetTitleStringSetup{expand} to work.

```
\ifdef\GetTitleStringDisableCommands
{\GetTitleStringDisableCommands{\let\glsxtrtitleorpdforheading\@thirdofthree
   \let\glsxtrifinmark\@firstoftwo}}
{}
```

Hook used in new definition of \markboth and \markright to make some changes to apply to the marks:

```
\newcommand*{\glsxtrmarkhook}{%
```

Save current definitions:

```
\@glsxtr@saveMakeUppercase
\let\@glsxtr@org@glsxtrtitleorpdforheading\glsxtrtitleorpdforheading
\let\@glsxtr@org@glsxtrtitleshort\glsxtrtitleshort
\let\@glsxtr@org@glsxtrtitleshortpl\glsxtrtitleshortpl
\let\@glsxtr@org@Glsxtrtitleshort\Glsxtrtitleshort
\let\@glsxtr@org@Glsxtrtitleshortpl\Glsxtrtitleshortpl
\let\@glsxtr@org@GLSxtrtitleshort\GLSxtrtitleshort
\let\@glsxtr@org@GLSxtrtitleshortpl\GLSxtrtitleshortpl
\let\@glsxtr@org@glsxtrtitlename\glsxtrtitlename
\let\@glsxtr@org@Glsxtrtitlename\Glsxtrtitlename

\let\@glsxtr@org@GLSxtrtitlename\GLSxtrtitlename
\let\@glsxtr@org@glsxtrtitletext\glsxtrtitletext
\let\@glsxtr@org@Glsxtrtitletext\Glsxtrtitletext
\let\@glsxtr@org@GLSxtrtitletext\GLSxtrtitletext
\let\@glsxtr@org@glsxtrtitleplural\glsxtrtitleplural
\let\@glsxtr@org@Glsxtrtitleplural\Glsxtrtitleplural
\let\@glsxtr@org@GLSxtrtitleplural\GLSxtrtitleplural
\let\@glsxtr@org@glsxtrtitlefirst\glsxtrtitlefirst
\let\@glsxtr@org@Glsxtrtitlefirst\Glsxtrtitlefirst
```

```
\let\@glsxtr@org@GLSxtrtitlefirst\GLSxtrtitlefirst
\let\@glsxtr@org@glsxtrtitlefirstplural\glsxtrtitlefirstplural
\let\@glsxtr@org@Glsxtrtitlefirstplural\Glsxtrtitlefirstplural
\let\@glsxtr@org@GLSxtrtitlefirstplural\GLSxtrtitlefirstplural
\let\@glsxtr@org@glsxtrtitlelong\glsxtrtitlelong
\let\@glsxtr@org@glsxtrtitlelongpl\glsxtrtitlelongpl
\let\@glsxtr@org@Glsxtrtitlelong\Glsxtrtitlelong
\let\@glsxtr@org@Glsxtrtitlelongpl\Glsxtrtitlelongpl
\let\@glsxtr@org@glsxtrtitlefull\glsxtrtitlefull
\let\@glsxtr@org@glsxtrtitlefullpl\glsxtrtitlefullpl
\let\@glsxtr@org@Glsxtrtitlefull\Glsxtrtitlefull
\let\@glsxtr@org@Glsxtrtitlefullpl\Glsxtrtitlefullpl
\let\@glsxtr@org@GLSxtrtitlefull\GLSxtrtitlefull
\let\@glsxtr@org@GLSxtrtitlefullpl\GLSxtrtitlefullpl
```

New definitions

```
\let\glsxtrifinmark\@firstoftwo
\@glsxtr@assignMakeUppercase
\let\glsxtrtitleorpdforheading\@thirdofthree
\let\glsxtrtitleshort\glsxtrheadshort
\let\glsxtrtitleshortpl\glsxtrheadshortpl
\let\Glsxtrtitleshort\Glsxtrheadshort
\let\Glsxtrtitleshortpl\Glsxtrheadshortpl
\let\GLSxtrtitleshort\GLSxtrheadshort
\let\GLSxtrtitleshortpl\GLSxtrheadshortpl
\let\glsxtrtitlename\glsxtrheadname
\let\Glsxtrtitlename\Glsxtrheadname
\let\GLSxtrtitlename\GLSxtrheadname
\let\glsxtrtitletext\glsxtrheadtext
\let\Glsxtrtitletext\Glsxtrheadtext
\let\GLSxtrtitletext\GLSxtrheadtext
\let\glsxtrtitleplural\glsxtrheadplural
\let\Glsxtrtitleplural\Glsxtrheadplural
\let\GLSxtrtitleplural\GLSxtrheadplural
\let\glsxtrtitlefirst\glsxtrheadfirst
\let\Glsxtrtitlefirst\Glsxtrheadfirst
\let\GLSxtrtitlefirst\GLSxtrheadfirst
\let\glsxtrtitlefirstplural\glsxtrheadfirstplural
\let\Glsxtrtitlefirstplural\Glsxtrheadfirstplural
\let\GLSxtrtitlefirstplural\GLSxtrheadfirstplural
\let\glsxtrtitlelong\glsxtrheadlong
\let\glsxtrtitlelongpl\glsxtrheadlongpl
\let\Glsxtrtitlelong\Glsxtrheadlong
\let\Glsxtrtitlelongpl\Glsxtrheadlongpl
\let\glsxtrtitlefull\glsxtrheadfull
\let\glsxtrtitlefullpl\glsxtrheadfullpl
\let\Glsxtrtitlefull\Glsxtrheadfull
\let\Glsxtrtitlefullpl\Glsxtrheadfullpl
\let\GLSxtrtitlefull\GLSxtrheadfull
\let\GLSxtrtitlefullpl\GLSxtrheadfullpl
```

```
}
```

\glsxtrrestoremarkhook 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.)

```
\newcommand*{\glsxtrrestoremarkhook}{%
  \let\glsxtrifinmark\@secondoftwo
  \@glsxtr@restoreMakeUppercase
  \let\glsxtrtitleorpdforheading\@glsxtr@org@glsxtrtitleorpdforheading
  \let\glsxtrtitleshort\@glsxtr@org@glsxtrtitleshort
  \let\glsxtrtitleshortpl\@glsxtr@org@glsxtrtitleshortpl
  \let\Glsxtrtitleshort\@glsxtr@org@Glsxtrtitleshort
  \let\Glsxtrtitleshortpl\@glsxtr@org@Glsxtrtitleshortpl
  \let\GLSxtrtitleshort\@glsxtr@org@GLSxtrtitleshort
  \let\GLSxtrtitleshortpl\@glsxtr@org@GLSxtrtitleshortpl
  \let\glsxtrtitlename\@glsxtr@org@glsxtrtitlename
  \let\Glsxtrtitlename\@glsxtr@org@Glsxtrtitlename
  \let\GLSxtrtitlename\@glsxtr@org@GLSxtrtitlename
  \let\glsxtrtitletext\@glsxtr@org@glsxtrtitletext
  \let\Glsxtrtitletext\@glsxtr@org@Glsxtrtitletext
  \let\GLSxtrtitletext\@glsxtr@org@GLSxtrtitletext
  \let\glsxtrtitleplural\@glsxtr@org@glsxtrtitleplural
  \let\Glsxtrtitleplural\@glsxtr@org@Glsxtrtitleplural
  \let\GLSxtrtitleplural\@glsxtr@org@GLSxtrtitleplural
  \let\glsxtrtitlefirst\@glsxtr@org@glsxtrtitlefirst
  \let\Glsxtrtitlefirst\@glsxtr@org@Glsxtrtitlefirst
  \let\GLSxtrtitlefirst\@glsxtr@org@GLSxtrtitlefirst
  \let\glsxtrtitlefirstplural\@glsxtr@org@glsxtrtitlefirstplural
  \let\Glsxtrtitlefirstplural\@glsxtr@org@Glsxtrtitlefirstplural
  \let\GLSxtrtitlefirstplural\@glsxtr@org@GLSxtrtitlefirstplural
  \let\glsxtrtitlelong\@glsxtr@org@glsxtrtitlelong
  \let\glsxtrtitlelongpl\@glsxtr@org@glsxtrtitlelongpl
  \let\Glsxtrtitlelong\@glsxtr@org@Glsxtrtitlelong
  \let\Glsxtrtitlelongpl\@glsxtr@org@Glsxtrtitlelongpl
  \let\glsxtrtitlefull\@glsxtr@org@glsxtrtitlefull
  \let\glsxtrtitlefullpl\@glsxtr@org@glsxtrtitlefullpl
  \let\Glsxtrtitlefull\@glsxtr@org@Glsxtrtitlefull
  \let\Glsxtrtitlefullpl\@glsxtr@org@Glsxtrtitlefullpl
  \let\GLSxtrtitlefull\@glsxtr@org@GLSxtrtitlefull
  \let\GLSxtrtitlefullpl\@glsxtr@org@GLSxtrtitlefullpl
}
```

Instead of using one document-wide conditional, use headuc attribute to
determine whether or not to use the all upper case form.

\glsxtrtitleopts Make it possible to change the default options within the title (but not the page
header or table of contents).

```
\newcommand*{\glsxtrtitleopts}{noindex,hyper=false}
```

\glsxtr@title@field

> \glsxtr@title@field{⟨*cs*⟩}{⟨*label*⟩}

Used by all the \glsxtrtitle⟨*field*⟩ commands for consistency.

```
\newcommand*{\glsxtr@title@field}[2]{%
  \expandafter#1\expandafter[\glsxtrtitleopts]{#2}[]%
}
```

\glsxtrheadshort   Command used to display short form in the page header.

```
\newcommand*{\glsxtrheadshort}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSxtrshort[noindex,hyper=false]{#1}[]%
   }%
   {%
     \glsxtrshort[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\glsxtrtitleshort   Command to display short form of abbreviation in section title.

```
\newrobustcmd*{\glsxtrtitleshort}[1]{%
  \glsxtr@title@field\glsxtrshort{#1}%
}
```

\glsxtrheadshortpl   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.

```
\newcommand*{\glsxtrheadshortpl}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSxtrshortpl[noindex,hyper=false]{#1}[]%
   }%
   {%
     \glsxtrshortpl[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\glsxtrtitleshortpl   Command to display plural short form of abbreviation in section title.

```
\newrobustcmd*{\glsxtrtitleshortpl}[1]{%
  \glsxtr@title@field\glsxtrshortpl{#1}%
}
```

\Glsxtrheadshort Command used to display short form in the page header with the first letter converted to upper case.

```
\newcommand*{\Glsxtrheadshort}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSxtrshort[noindex,hyper=false]{#1}[]%
   }%
   {%
     \Glsxtrshort[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\Glsxtrtitleshort Command to display short form of abbreviation in section title with the first letter converted to upper case.

```
\newrobustcmd*{\Glsxtrtitleshort}[1]{%
  \glsxtr@title@field\Glsxtrshort{#1}%
}
```

\GLSxtrheadshort There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadshort}[1]{%
 \protect\NoCaseChange
 {%
   \GLSxtrshort[noindex,hyper=false]{#1}[]%
 }%
}
```

\GLSxtrtitleshort Command to display short form of abbreviation in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitleshort}[1]{%
  \glsxtr@title@field\GLSxtrshort{#1}%
}
```

\GLSxtrheadshortpl There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadshortpl}[1]{%
 \protect\NoCaseChange
 {%
   \GLSxtrshortpl[noindex,hyper=false]{#1}[]%
 }%
}
```

\Glsxtrheadshortpl Command used to display plural short form in the page header with the first letter converted to upper case.

```
\newcommand*{\Glsxtrheadshortpl}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
```

```
                  \GLSxtrshortpl[noindex,hyper=false]{#1}[]%
               }%
               {%
                  \Glsxtrshortpl[noindex,hyper=false]{#1}[]%
               }%
             }%
           }
```

\Glsxtrtitleshortpl   Command to display plural short form of abbreviation in section title with the
                      first letter converted to upper case.

```
             \newrobustcmd*{\Glsxtrtitleshortpl}[1]{%
               \glsxtr@title@field\Glsxtrshortpl{#1}%
             }
```

\GLSxtrtitleshortpl   Command to display plural short form of abbreviation in section title in all
                      upper case.

```
             \newrobustcmd*{\GLSxtrtitleshortpl}[1]{%
               \glsxtr@title@field\GLSxtrshortpl{#1}%
             }
```

\glsxtrheadname   As above but for the name value.

```
             \newcommand*{\glsxtrheadname}[1]{%
              \protect\NoCaseChange
              {%
                \glsxtrifheaduc{#1}%
                {%
                  \GLSname[noindex,hyper=false]{#1}[]%
                }%
                {%
                  \glsname[noindex,hyper=false]{#1}[]%
                }%
              }%
             }
```

\glsxtrtitlename   Command to display name value in section title.

```
             \newrobustcmd*{\glsxtrtitlename}[1]{%
               \glsxtr@title@field\glsname{#1}%
             }
```

\Glsxtrheadname   First letter converted to upper case

```
             \newcommand*{\Glsxtrheadname}[1]{%
              \protect\NoCaseChange
              {%
                \glsxtrifheaduc{#1}%
                {%
                  \GLSname[noindex,hyper=false]{#1}[]%
                }%
                {%
                  \Glsname[noindex,hyper=false]{#1}[]%
```

```
    }%
   }%
  }
```

\Glsxtrtitlename Command to display name value in section title with the first letter changed to upper case.

```
\newrobustcmd*{\Glsxtrtitlename}[1]{%
  \glsxtr@title@field\Glsname{#1}%
}
```

\GLSxtrheadname There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadname}[1]{%
 \protect\NoCaseChange
 {%
   \GLSname[noindex,hyper=false]{#1}[]%
 }%
}
```

\GLSxtrtitlename Command to display name value in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlename}[1]{%
  \glsxtr@title@field\GLSname{#1}%
}
```

\glsxtrheadtext As above but for the text value.

```
\newcommand*{\glsxtrheadtext}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLStext[noindex,hyper=false]{#1}[]%
   }%
   {%
     \glstext[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\glsxtrtitletext Command to display text value in section title.

```
\newrobustcmd*{\glsxtrtitletext}[1]{%
  \glsxtr@title@field\glstext{#1}%
}
```

\Glsxtrheadtext First letter converted to upper case

```
\newcommand*{\Glsxtrheadtext}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLStext[noindex,hyper=false]{#1}[]%
```

```
        }%
        {%
          \Glstext[noindex,hyper=false]{#1}[]%
        }%
      }%
    }
```

\Glsxtrtitletext Command to display text value in section title with the first letter changed to upper case.

```
\newrobustcmd*{\Glsxtrtitletext}[1]{%
  \glsxtr@title@field\Glstext{#1}%
}
```

\GLSxtrheadtext There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadtext}[1]{%
 \protect\NoCaseChange
 {%
   \GLStext[noindex,hyper=false]{#1}[]%
 }%
}
```

\GLSxtrtitletext Command to display text value in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitletext}[1]{%
  \glsxtr@title@field\GLStext{#1}%
}
```

\glsxtrheadplural As above but for the plural value.

```
\newcommand*{\glsxtrheadplural}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSplural[noindex,hyper=false]{#1}[]%
   }%
   {%
     \glsplural[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\glsxtrtitleplural Command to display plural value in section title.

```
\newrobustcmd*{\glsxtrtitleplural}[1]{%
  \glsxtr@title@field\glsplural{#1}%
}
```

\Glsxtrheadplural Convert first letter to upper case.

```
\newcommand*{\Glsxtrheadplural}[1]{%
 \protect\NoCaseChange
 {%
```

```
      \glsxtrifheaduc{#1}%
      {%
        \GLSplural[noindex,hyper=false]{#1}[]%
      }%
      {%
        \Glsplural[noindex,hyper=false]{#1}[]%
      }%
    }%
  }
```

\Glsxtrtitleplural Command to display plural value in section title with the first letter changed to
upper case.

```
\newrobustcmd*{\Glsxtrtitleplural}[1]{%
  \glsxtr@title@field\Glsplural{#1}%
}
```

\GLSxtrheadplural There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadplural}[1]{%
 \protect\NoCaseChange
 {%
   \GLSplural[noindex,hyper=false]{#1}[]%
 }%
}
```

\GLSxtrtitleplural Command to display plural value in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitleplural}[1]{%
  \glsxtr@title@field\GLSplural{#1}%
}
```

\glsxtrheadfirst As above but for the first value.

```
\newcommand*{\glsxtrheadfirst}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSfirst[noindex,hyper=false]{#1}[]%
   }%
   {%
     \glsfirst[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\glsxtrtitlefirst Command to display first value in section title.

```
\newrobustcmd*{\glsxtrtitlefirst}[1]{%
  \glsxtr@title@field\glsfirst{#1}%
}
```

\Glsxtrheadfirst First letter converted to upper case

```
\newcommand*{\Glsxtrheadfirst}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSfirst[noindex,hyper=false]{#1}[]%
   }%
   {%
     \Glsfirst[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\Glsxtrtitlefirst Command to display first value in section title with the first letter changed to upper case.

```
\newrobustcmd*{\Glsxtrtitlefirst}[1]{%
  \glsxtr@title@field\Glsfirst{#1}%
}
```

\GLSxtrheadfirst There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadfirst}[1]{%
 \protect\NoCaseChange
 {%
   \GLSfirst[noindex,hyper=false]{#1}[]%
 }%
}
```

\GLSxtrtitlefirst Command to display first value in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlefirst}[1]{%
  \glsxtr@title@field\GLSfirst{#1}%
}
```

\glsxtrheadfirstplural As above but for the firstplural value.

```
\newcommand*{\glsxtrheadfirstplural}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSfirstplural[noindex,hyper=false]{#1}[]%
   }%
   {%
     \glsfirstplural[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\glsxtrtitlefirstplural Command to display firstplural value in section title.

```
\newrobustcmd*{\glsxtrtitlefirstplural}[1]{%
```

```
                       \glsxtr@title@field\glsfirstplural{#1}%
                      }
```

\Glsxtrheadfirstplural First letter converted to upper case

```
\newcommand*{\Glsxtrheadfirstplural}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSfirstplural[noindex,hyper=false]{#1}[]%
   }%
   {%
     \Glsfirstplural[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\Glsxtrtitlefirstplural Command to display first value in section title with the first letter changed to
                       upper case.

```
\newrobustcmd*{\Glsxtrtitlefirstplural}[1]{%
  \glsxtr@title@field\Glsfirstplural{#1}%
}
```

\GLSxtrheadfirstplural There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadfirstplural}[1]{%
 \protect\NoCaseChange
 {%
   \GLSfirstplural[noindex,hyper=false]{#1}[]%
 }%
}
```

\GLSxtrtitlefirstplural Command to display first value in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlefirstplural}[1]{%
  \glsxtr@title@field\GLSfirstplural{#1}%
}
```

\glsxtrheadlong Command used to display long form in the page header.

```
\newcommand*{\glsxtrheadlong}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSxtrlong[noindex,hyper=false]{#1}[]%
   }%
   {%
     \glsxtrlong[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

`\glsxtrtitlelong` Command to display long form of abbreviation in section title.

```
\newrobustcmd*{\glsxtrtitlelong}[1]{%
  \glsxtr@title@field\glsxtrlong{#1}%
}
```

`\glsxtrheadlongpl` 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.

```
\newcommand*{\glsxtrheadlongpl}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
      \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
   }%
   {%
      \glsxtrlongpl[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

`\glsxtrtitlelongpl` Command to display plural long form of abbreviation in section title.

```
\newrobustcmd*{\glsxtrtitlelongpl}[1]{%
  \glsxtr@title@field\glsxtrlongpl{#1}%
}
```

`\Glsxtrheadlong` Command used to display long form in the page header with the first letter converted to upper case.

```
\newcommand*{\Glsxtrheadlong}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSxtrlong[noindex,hyper=false]{#1}[]%
   }%
   {%
     \Glsxtrlong[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

`\Glsxtrtitlelong` Command to display long form of abbreviation in section title with the first letter converted to upper case.

```
\newrobustcmd*{\Glsxtrtitlelong}[1]{%
  \glsxtr@title@field\Glsxtrlong{#1}%
}
```

\GLSxtrtitlelong Command to display long form of abbreviation in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlelong}[1]{%
  \glsxtr@title@field\GLSxtrlong{#1}%
}
```

\GLSxtrheadlong There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadlong}[1]{%
 \protect\NoCaseChange
 {%
   \GLSxtrlong[noindex,hyper=false]{#1}[]%
 }%
}
```

\Glsxtrheadlongpl Command used to display plural long form in the page header with the first letter converted to upper case.

```
\newcommand*{\Glsxtrheadlongpl}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
   }%
   {%
     \Glsxtrlongpl[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\Glsxtrtitlelongpl Command to display plural long form of abbreviation in section title with the first letter converted to upper case.

```
\newrobustcmd*{\Glsxtrtitlelongpl}[1]{%
  \glsxtr@title@field\Glsxtrlongpl{#1}%
}
```

\GLSxtrtitlelongpl Command to display plural long form of abbreviation in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlelongpl}[1]{%
  \glsxtr@title@field\GLSxtrlongpl{#1}%
}
```

\GLSxtrheadlongpl There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadlongpl}[1]{%
 \protect\NoCaseChange
 {%
   \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
 }%
}
```

374

\glsxtrheadfull Command used to display full form in the page header.

```
\newcommand*{\glsxtrheadfull}[1]{%
\protect\NoCaseChange
{%
  \glsxtrifheaduc{#1}%
  {%
    \GLSxtrfull[noindex,hyper=false]{#1}[]%
  }%
  {%
    \glsxtrfull[noindex,hyper=false]{#1}[]%
  }%
}%
}
```

\glsxtrtitlefull Command to display full form of abbreviation in section title.

```
\newrobustcmd*{\glsxtrtitlefull}[1]{%
  \glsxtr@title@field\glsxtrfull{#1}%
}
```

\glsxtrheadfullpl 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.

```
\newcommand*{\glsxtrheadfullpl}[1]{%
\protect\NoCaseChange
{%
  \glsxtrifheaduc{#1}%
  {%
    \GLSxtrfullpl[noindex,hyper=false]{#1}[]%
  }%
  {%
    \glsxtrfullpl[noindex,hyper=false]{#1}[]%
  }%
}%
}
```

\glsxtrtitlefullpl Command to display plural full form of abbreviation in section title.

```
\newrobustcmd*{\glsxtrtitlefullpl}[1]{%
  \glsxtr@title@field\glsxtrfullpl{#1}%
}
```

\Glsxtrheadfull Command used to display full form in the page header with the first letter converted to upper case.

```
\newcommand*{\Glsxtrheadfull}[1]{%
\protect\NoCaseChange
{%
  \glsxtrifheaduc{#1}%
  {%
    \GLSxtrfull[noindex,hyper=false]{#1}[]%
```

375

```
      }%
      {%
        \Glsxtrfull[noindex,hyper=false]{#1}[]%
      }%
    }%
    }
```

\Glsxtrtitlefull Command to display full form of abbreviation in section title with the first letter
converted to upper case.

```
\newrobustcmd*{\Glsxtrtitlefull}[1]{%
  \glsxtr@title@field\Glsxtrfull{#1}%
}
```

\GLSxtrheadfull There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadfull}[1]{%
 \protect\NoCaseChange
 {%
   \GLSxtrfull[noindex,hyper=false]{#1}[]%
 }%
}
```

\GLSxtrtitlefull Command to display full form of abbreviation in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlefull}[1]{%
  \glsxtr@title@field\GLSxtrfull{#1}%
}
```

\Glsxtrheadfullpl Command used to display plural full form in the page header with the first letter
converted to upper case.

```
\newcommand*{\Glsxtrheadfullpl}[1]{%
 \protect\NoCaseChange
 {%
   \glsxtrifheaduc{#1}%
   {%
     \GLSxtrfullpl[noindex,hyper=false]{#1}[]%
   }%
   {%
     \Glsxtrfullpl[noindex,hyper=false]{#1}[]%
   }%
 }%
}
```

\Glsxtrtitlefullpl Command to display plural full form of abbreviation in section title with the
first letter converted to upper case.

```
\newrobustcmd*{\Glsxtrtitlefullpl}[1]{%
  \glsxtr@title@field\Glsxtrfullpl{#1}%
}
```

`\GLSxtrheadfullpl` There's no need to check for the headuc attribute.

```
\newcommand*{\GLSxtrheadfullpl}[1]{%
 \protect\NoCaseChange
 {%
   \GLSxtrfullpl[noindex,hyper=false]{#1}[]%
 }%
 }
```

`\GLSxtrtitlefullpl` Command to display plural full form of abbreviation in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlefullpl}[1]{%
   \glsxtr@title@field\GLSxtrfullpl{#1}%
 }
```

`\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.

```
\newcommand*{\glsfmtshort}[1]{%
  \glstexorpdfstring
    {\glsxtrtitleshort{#1}}%
    {\glsentryshort{#1}}%
 }
```

Similarly for the plural version.

`\glsfmtshortpl`

```
\newcommand*{\glsfmtshortpl}[1]{%
  \glstexorpdfstring
    {\glsxtrtitleshortpl{#1}}%
    {\glsentryshortpl{#1}}%
 }
```

Use the expandable `\MFUsentencecase` in the PDF bookmark.

`\Glsfmtshort` Singular form (first letter uppercase).

```
\newcommand*{\Glsfmtshort}[1]{%
  \glstexorpdfstring
    {\Glsxtrtitleshort{#1}}%
    {\MFUsentencecase{\glsentryshort{#1}}}%
 }
 \glsmfuaddmap{\glsfmtshort}{\Glsfmtshort}
```

`\Glsfmtshortpl` Plural form (first letter uppercase).

```
\newcommand*{\Glsfmtshortpl}[1]{%
  \glstexorpdfstring
  {\Glsxtrtitleshortpl{#1}}%
  {\MFUsentencecase{\glsentryshortpl{#1}}}%
 }
 \glsmfuaddmap{\glsfmtshortpl}{\Glsfmtshortpl}
```

Similarly for all-caps.

\GLSfmtshort

```
\newcommand*{\GLSfmtshort}[1]{%
  \glstexorpdfstring
    {\GLSxtrtitleshort{#1}}%
    {\GLSxtrusefield{#1}{short}}%
}
\glsmfublocker{\GLSfmtshort}
```

\GLSfmtshortpl

```
\newcommand*{\GLSfmtshortpl}[1]{%
  \glstexorpdfstring
    {\GLSxtrtitleshortpl{#1}}%
    {\GLSxtrusefield{#1}{shortpl}}%
}
\glsmfublocker{\GLSfmtshortpl}
```

\glsfmtname As above but for the name value.

```
\newcommand*{\glsfmtname}[1]{%
  \glstexorpdfstring
  {\glsxtrtitlename{#1}}%
  {\glsentryname{#1}}%
}
```

\Glsfmtname First letter converted to upper case.

```
\newcommand*{\Glsfmtname}[1]{%
  \glstexorpdfstring
  {\Glsxtrtitlename{#1}}%
  {\MFUsentencecase{\glsentryname{#1}}}%
}
\glsmfuaddmap{\glsfmtname}{\Glsfmtname}
```

\GLSfmtname All upper case.

```
\newcommand*{\GLSfmtname}[1]{%
  \glstexorpdfstring
  {\GLSxtrtitlename{#1}}%
  {\GLSxtrusefield{#1}{name}}%
}
\glsmfublocker{\GLSfmtname}
```

\glsfmttext As above but for the text value.

```
\newcommand*{\glsfmttext}[1]{%
  \glstexorpdfstring
  {\glsxtrtitletext{#1}}%
  {\glsentrytext{#1}}%
}
```

\Glsfmttext First letter converted to upper case.

```
\newcommand*{\Glsfmttext}[1]{%
  \glstexorpdfstring
```

378

```
                {\Glsxtrtitletext{#1}}%
                {\MFUsentencecase{\glsentrytext{#1}}}%
             }
             \glsmfuaddmap{\glsfmttext}{\Glsfmttext}
```

\GLSfmttext All upper case.

```
             \newcommand*{\GLSfmttext}[1]{%
               \glstexorpdfstring
               {\GLSxtrtitletext{#1}}%
               {\GLSxtrusefield{#1}{text}}%
             }
             \glsmfublocker{\GLSfmttext}
```

\glsfmtplural As above but for the plural value.

```
             \newcommand*{\glsfmtplural}[1]{%
               \glstexorpdfstring
               {\glsxtrtitleplural{#1}}%
               {\glsentryplural{#1}}%
             }
```

\Glsfmtplural First letter converted to upper case.

```
             \newcommand*{\Glsfmtplural}[1]{%
               \glstexorpdfstring
               {\Glsxtrtitleplural{#1}}%
               {\MFUsentencecase{\glsentryplural{#1}}}%
             }
             \glsmfuaddmap{\glsfmtplural}{\Glsfmtplural}
```

\GLSfmtplural All upper case.

```
             \newcommand*{\GLSfmtplural}[1]{%
               \glstexorpdfstring
               {\GLSxtrtitleplural{#1}}%
               {\GLSxtrusefield{#1}{plural}}%
             }
             \glsmfublocker{\GLSfmtplural}
```

\glsfmtfirst As above but for the first value.

```
             \newcommand*{\glsfmtfirst}[1]{%
               \glstexorpdfstring
               {\glsxtrtitlefirst{#1}}%
               {\glsentryfirst{#1}}%
             }
```

\Glsfmtfirst First letter converted to upper case.

```
             \newcommand*{\Glsfmtfirst}[1]{%
               \glstexorpdfstring
               {\Glsxtrtitlefirst{#1}}%
               {\MFUsentencecase{\glsentryfirst{#1}}}%
             }
             \glsmfuaddmap{\glsfmtfirst}{\Glsfmtfirst}
```

\GLSfmtfirst All upper case.

```
\newcommand*{\GLSfmtfirst}[1]{%
  \glstexorpdfstring
  {\GLSxtrtitlefirst{#1}}%
  {\GLSxtrusefield{#1}{first}}%
}
\glsmfublocker{\GLSfmtfirst}
```

\glsfmtfirstpl As above but for the firstplural value.

```
\newcommand*{\glsfmtfirstpl}[1]{%
  \glstexorpdfstring
  {\glsxtrtitlefirstplural{#1}}%
  {\glsxtrusefield{#1}{firstpl}}%
}
```

\Glsfmtfirstpl First letter converted to upper case.

```
\newcommand*{\Glsfmtfirstpl}[1]{%
  \glstexorpdfstring
  {\Glsxtrtitlefirstplural{#1}}%
  {\MFUsentencecase{\glsentryfirstplural{#1}}}%
}
```

\GLSfmtfirstpl All upper case.

```
\newcommand*{\GLSfmtfirstpl}[1]{%
  \glstexorpdfstring
  {\GLSxtrtitlefirstplural{#1}}%
  {\GLSxtrusefield{#1}{firstpl}}%
}
\glsmfublocker{\GLSfmtfirstpl}
```

\glsfmtlong As above but for the long value.

```
\newcommand*{\glsfmtlong}[1]{%
  \glstexorpdfstring
  {\glsxtrtitlelong{#1}}%
  {\glsentrylong{#1}}%
}
```

\Glsfmtlong First letter converted to upper case.

```
\newcommand*{\Glsfmtlong}[1]{%
  \glstexorpdfstring
  {\Glsxtrtitlelong{#1}}%
  {\glspdfsentencecase{\glsentrylong{#1}}}%
}
\glsmfuaddmap{\glsfmtlong}{\Glsfmtlong}
```

\GLSfmtlong All upper case.

```
\newcommand*{\GLSfmtlong}[1]{%
  \glstexorpdfstring
  {\GLSxtrtitlelong{#1}}%
```

```
                  {\GLSxtrusefield{#1}{long}}%
                }
                \glsmfublocker{\GLSfmtlong}
```

\glsfmtlongpl   As above but for the longplural value.
```
                \newcommand*{\glsfmtlongpl}[1]{%
                  \glstexorpdfstring
                  {\glsxtrtitlelongpl{#1}}%
                  {\glsentrylongpl{#1}}%
                }
```

\Glsfmtlongpl   First letter converted to upper case.
```
                \newcommand*{\Glsfmtlongpl}[1]{%
                  \glstexorpdfstring
                  {\Glsxtrtitlelongpl{#1}}%
                  {\glspdfsentencecase{\glsentrylongpl{#1}}}%
                }
                \glsmfuaddmap{\glsfmtlongpl}{\Glsfmtlongpl}
```

\GLSfmtlongpl   All upper case.
```
                \newcommand*{\GLSfmtlongpl}[1]{%
                  \glstexorpdfstring
                  {\GLSxtrtitlelongpl{#1}}%
                  {\GLSxtrusefield{#1}{longpl}}%
                }
                \glsmfublocker{\GLSfmtlongpl}
```

\glspdffmtfull   Can't use \glsxtrinlinefullformat in PDF bookmarks as it's not fully ex-
                 pandable. This command is for the PDF part of \texorpdfstring for the full
                 form.
```
                \newcommand*{\glspdffmtfull}[1]{\glsentrylong{#1} (\glsentryshort{#1})}%
```

\glspdffmtfullpl Likewise for plural.
```
                \newcommand*{\glspdffmtfullpl}[1]{\glsentrylongpl{#1} (\glsentryshortpl{#1})}%
```

\glsfmtfull   In-line full format.
```
                \newcommand*{\glsfmtfull}[1]{%
                  \glstexorpdfstring
                  {\glsxtrtitlefull{#1}}%
                  {\glspdffmtfull{#1}}%
                }
```

\Glsfmtfull   First letter converted to upper case.
```
                \newcommand*{\Glsfmtfull}[1]{%
                  \glstexorpdfstring
                  {\Glsxtrtitlefull{#1}}%
                  {\glspdfsentencecase{\glspdffmtfull{#1}{}}}%
                }
                \glsmfuaddmap{\glsfmtfull}{\Glsfmtfull}
```

`\GLSfmtfull` All upper case. This explicitly uses `\text_uppercase:n` in case an old version of glossaries or mfirstuc is present.

```
\ExplSyntaxOn
\newcommand*{\GLSfmtfull}[1]{%
  \glstexorpdfstring
  {\GLSxtrtitlefull{#1}}%
  {\text_uppercase:n{\glspdffmtfull{#1}}}%
}
\ExplSyntaxOff
\glsmfublocker{\GLSfmtfull}
```

`\glsfmtfullpl` In-line full plural format.

```
\newcommand*{\glsfmtfullpl}[1]{%
  \glstexorpdfstring
  {\glsxtrtitlefullpl{#1}}%
  {\glspdffmtfullpl{#1}}%
}
```

`\Glsfmtfullpl` First letter converted to upper case.

```
\newcommand*{\Glsfmtfullpl}[1]{%
  \glstexorpdfstring
  {\Glsxtrtitlefullpl{#1}}%
  {\glspdfsentencecase{\glspdffmtfullpl{#1}{}}}%
}
\glsmfuaddmap{\glsfmtfullpl}{\Glsfmtfullpl}
```

`\GLSfmtfullpl` All upper case. This explicitly uses `\text_uppercase:n` in case an old version of glossaries or mfirstuc is present.

```
\ExplSyntaxOn
\newcommand*{\GLSfmtfullpl}[1]{%
  \glstexorpdfstring
  {\GLSxtrtitlefullpl{#1}}%
  {\text_uppercase:n{\glspdffmtfullpl{#1}{}}}%
}
\ExplSyntaxOff
\glsmfublocker{\GLSfmtfullpl}
```

1.9 Prefixes

Provide support for glossaries-prefix.

`\pglsprefix`

$\boxed{\texttt{\textbackslash pglsprefix\{}\langle \mathit{entry\text{-}label}\rangle\texttt{\}\{}\langle \mathit{prefix\text{-}field}\rangle\texttt{\}}}$

A shortcut way of inserting the prefix and separator if they are required. If this needs to be redefined, use `\ifglsfieldvoid` for an expandable test.

```
\newcommand{\pglsprefix}[2]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@#2}{}%
```

```
  {\csuse{glo@\glsdetoklabel{#1}@#2}\glsprefixsep}%
}
```

\Pglsprefix{⟨*entry-label*⟩}{⟨*prefix-field*⟩}

\Pglsprefix

Similar to \pglsprefix but sentence case. The conditional is omitted as it will
have to already be checked.

```
\newcommand{\Pglsprefix}[2]{%
  \Glsxtrusefield{#1}{#2}\glsprefixsep
}
```

\PGLSprefix{⟨*entry-label*⟩}{⟨*prefix-field*⟩}

\PGLSprefix

As \pglsprefix but all caps.

```
\newcommand{\PGLSprefix}[2]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@#2}{}%
  {\glsuppercase{\csuse{glo@\glsdetoklabel{#1}@#2}\glsprefixsep}}%
}
```

Abbreviations. Short form uses prefix and prefixplural fields.

\pglsxtrshort No case-change.

```
\newrobustcmd*{\pglsxtrshort}{\@gls@hyp@opt\ns@pglsxtrshort}
\newcommand*{\ns@pglsxtrshort}[2][]{%
  \new@ifnextchar[{\@pglsxtrshort{#1}{#2}}{\@pglsxtrshort{#1}{#2}[]}%
}
\def\@pglsxtrshort#1#2[#3]{%
  \pglsprefix{#2}{prefix}%
  \@glsxtrshort{#1}{#2}[#3]%
}
```

\Pglsxtrshort Sentence case.

```
\newrobustcmd*{\Pglsxtrshort}{\@gls@hyp@opt\ns@Pglsxtrshort}
\newcommand*{\ns@Pglsxtrshort}[2][]{%
  \new@ifnextchar[{\@Pglsxtrshort{#1}{#2}}{\@Pglsxtrshort{#1}{#2}[]}%
}
\def\@Pglsxtrshort#1#2[#3]{%
  \ifglshasprefix{#2}%
  {%
    \Pglsprefix{#2}{prefix}%
    \@glsxtrshort{#1}{#2}[#3]%
  }%
  {\@Glsxtrshort{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglsxtrshort}{\Pglsxtrshort}
```

\PGLSxtrshort All-caps is also fairly simple.

```
\newrobustcmd*{\PGLSxtrshort}{\@gls@hyp@opt\ns@PGLSxtrshort}
\newcommand*{\ns@PGLSxtrshort}[2][]{%
  \new@ifnextchar[{\@PGLSxtrshort{#1}{#2}}{\@PGLSxtrshort{#1}{#2}[]}%
}
\def\@PGLSxtrshort#1#2[#3]{%
  \PGLSprefix{#2}{prefix}%
  \@GLSxtrshort{#1}{#2}[#3]%
}
\glsmfublocker{\PGLSxtrshort}
```

Short plural.

\pglsxtrshortpl

```
\newrobustcmd*{\pglsxtrshortpl}{\@gls@hyp@opt\ns@pglsxtrshortpl}
\newcommand*{\ns@pglsxtrshortpl}[2][]{%
  \new@ifnextchar[{\@pglsxtrshortpl{#1}{#2}}{\@pglsxtrshortpl{#1}{#2}[]}%
}
\def\@pglsxtrshortpl#1#2[#3]{%
  \pglsprefix{#2}{prefixplural}%
  \@glsxtrshortpl{#1}{#2}[#3]%
}
```

\Pglsxtrshortpl

```
\newrobustcmd*{\Pglsxtrshortpl}{\@gls@hyp@opt\ns@Pglsxtrshortpl}
\newcommand*{\ns@Pglsxtrshortpl}[2][]{%
  \new@ifnextchar[{\@Pglsxtrshortpl{#1}{#2}}{\@Pglsxtrshortpl{#1}{#2}[]}%
}
\def\@Pglsxtrshortpl#1#2[#3]{%
  \ifglshasprefixplural{#2}%
  {%
    \Pglsprefix{#2}{prefixplural}%
    \@glsxtrshortpl{#1}{#2}[#3]%
  }%
  {\@Glsxtrshortpl{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglsxtrshortpl}{\Pglsxtrshortpl}
```

\PGLSxtrshortpl All-caps is also fairly simple.

```
\newrobustcmd*{\PGLSxtrshortpl}{\@gls@hyp@opt\ns@PGLSxtrshortpl}
\newcommand*{\ns@PGLSxtrshortpl}[2][]{%
  \new@ifnextchar[{\@PGLSxtrshortpl{#1}{#2}}{\@PGLSxtrshortpl{#1}{#2}[]}%
}
\def\@PGLSxtrshortpl#1#2[#3]{%
  \PGLSprefix{#2}{prefixplural}%
  \@GLSxtrshortpl{#1}{#2}[#3]%
}
\glsmfublocker{\PGLSxtrshortpl}
```

Long form uses prefixfirst and prefixfirstplural fields.

`\pglsxtrlong` No case-change.

```
\newrobustcmd*{\pglsxtrlong}{\@gls@hyp@opt\ns@pglsxtrlong}
\newcommand*{\ns@pglsxtrlong}[2][]{%
  \new@ifnextchar[{\@pglsxtrlong{#1}{#2}}{\@pglsxtrlong{#1}{#2}[]}%
}
\def\@pglsxtrlong#1#2[#3]{%
  \pglsprefix{#2}{prefixfirst}%
  \@glsxtrlong{#1}{#2}[#3]%
}
```

`\Pglsxtrlong` Sentence case.

```
\newrobustcmd*{\Pglsxtrlong}{\@gls@hyp@opt\ns@Pglsxtrlong}
\newcommand*{\ns@Pglsxtrlong}[2][]{%
  \new@ifnextchar[{\@Pglsxtrlong{#1}{#2}}{\@Pglsxtrlong{#1}{#2}[]}%
}
\def\@Pglsxtrlong#1#2[#3]{%
  \ifglshasprefixfirst{#2}%
  {%
    \Pglsprefix{#2}{prefixfirst}%
    \@glsxtrlong{#1}{#2}[#3]%
  }%
  {\@Glsxtrlong{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglsxtrlong}{\Pglsxtrlong}
```

`\PGLSxtrlong` All-caps is also fairly simple.

```
\newrobustcmd*{\PGLSxtrlong}{\@gls@hyp@opt\ns@PGLSxtrlong}
\newcommand*{\ns@PGLSxtrlong}[2][]{%
  \new@ifnextchar[{\@PGLSxtrlong{#1}{#2}}{\@PGLSxtrlong{#1}{#2}[]}%
}
\def\@PGLSxtrlong#1#2[#3]{%
  \PGLSprefix{#2}{prefixfirst}%
  \@GLSxtrlong{#1}{#2}[#3]%
}
\glsmfublocker{\PGLSxtrlong}
```

Long plural.

`\pglsxtrlongpl`

```
\newrobustcmd*{\pglsxtrlongpl}{\@gls@hyp@opt\ns@pglsxtrlongpl}
\newcommand*{\ns@pglsxtrlongpl}[2][]{%
  \new@ifnextchar[{\@pglsxtrlongpl{#1}{#2}}{\@pglsxtrlongpl{#1}{#2}[]}%
}
\def\@pglsxtrlongpl#1#2[#3]{%
  \pglsprefix{#2}{prefixfirstplural}%
  \@glsxtrlongpl{#1}{#2}[#3]%
}
```

`\Pglsxtrlongpl`

```
\newrobustcmd*{\Pglsxtrlongpl}{\@gls@hyp@opt\ns@Pglsxtrlongpl}
\newcommand*{\ns@Pglsxtrlongpl}[2][]{%
  \new@ifnextchar[{\@Pglsxtrlongpl{#1}{#2}}{\@Pglsxtrlongpl{#1}{#2}[]}%
}
\def\@Pglsxtrlongpl#1#2[#3]{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Pglsprefix{#2}{prefixfirstplural}%
    \@glsxtrlongpl{#1}{#2}[#3]%
  }%
  {\@Glsxtrlongpl{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglsxtrlongpl}{\Pglsxtrlongpl}
```

\PGLSxtrlongpl All-caps is also fairly simple.

```
\newrobustcmd*{\PGLSxtrlongpl}{\@gls@hyp@opt\ns@PGLSxtrlongpl}
\newcommand*{\ns@PGLSxtrlongpl}[2][]{%
  \new@ifnextchar[{\@PGLSxtrlongpl{#1}{#2}}{\@PGLSxtrlongpl{#1}{#2}[]}%
}
\def\@PGLSxtrlongpl#1#2[#3]{%
  \PGLSprefix{#2}{prefixfirstplural}%
  \@GLSxtrlongpl{#1}{#2}[#3]%
}
\glsmfublocker{\PGLSxtrlongpl}
```

   Title commands (analogous to \glsfmtshort etc).

\pglsfmtshort

```
\newcommand*{\pglsfmtshort}[1]{%
  \pglsprefix{#1}{prefix}%
  \glsfmtshort{#1}%
}
```

\Pglsfmtshort

```
\newcommand*{\Pglsfmtshort}[1]{%
 \glstexorpdfstring
  {\Pglsxtrtitleshort{#1}}%
  {\glspdfsentencecase{%
    \pglsprefix{#1}{prefix}%
    \glsentryshort{#1}}%
  }%
}
\glsmfuaddmap{\pglsfmtshort}{\Pglsfmtshort}
```

\Pglsxtrtitleshort

```
\newrobustcmd*{\Pglsxtrtitleshort}[1]{%
  \glsxtr@title@field\Pglsxtrshort{#1}%
}
```

\PGLSfmtshort

```
\newcommand*{\PGLSfmtshort}[1]{%
  \PGLSprefix{#1}{prefix}%
  \GLSfmtshort{#1}%
}
\glsmfublocker{\PGLSfmtshort}
```

\pglsfmtshortpl

```
\newcommand*{\pglsfmtshortpl}[1]{%
  \pglsprefix{#1}{prefixplural}%
  \glsfmtshortpl{#1}%
}
```

\Pglsfmtshortpl

```
\newcommand*{\Pglsfmtshortpl}[1]{%
 \glstexorpdfstring
  {\Pglsxtrtitleshortpl{#1}}%
  {\glspdfsentencecase
    {%
      \pglsprefix{#1}{prefixplural}%
      \glsentryshortpl{#1}%
    }%
  }%
}
\glsmfuaddmap{\pglsfmtshortpl}{\Pglsfmtshortpl}
```

\Pglsxtrtitleshortpl

```
\newrobustcmd*{\Pglsxtrtitleshortpl}[1]{%
  \glsxtr@title@field\Pglsxtrshortpl{#1}%
}
```

\PGLSfmtshortpl

```
\newcommand*{\PGLSfmtshortpl}[1]{%
  \PGLSprefix{#1}{prefixplural}%
  \GLSfmtshortpl{#1}%
}
\glsmfublocker{\PGLSfmtshortpl}
```

\pglsfmtlong

```
\newcommand*{\pglsfmtlong}[1]{%
  \pglsprefix{#1}{prefixfirst}%
  \glsfmtlong{#1}%
}
```

\Pglsfmtlong

```
\newcommand*{\Pglsfmtlong}[1]{%
 \glstexorpdfstring
  {\Pglsxtrtitlelong{#1}}%
  {\glspdfsentencecase{%
```

```
                  \pglsprefix{#1}{prefixfirst}%
                  \glsentrylong{#1}}%
                }%
              }
              \glsmfuaddmap{\pglsfmtlong}{\Pglsfmtlong}

\Pglsxtrtitlelong

              \newrobustcmd*{\Pglsxtrtitlelong}[1]{%
                \glsxtr@title@field\Pglsxtrlong{#1}%
              }

    \PGLSfmtlong

              \newcommand*{\PGLSfmtlong}[1]{%
                \PGLSprefix{#1}{prefixfirst}%
                \GLSfmtlong{#1}%
              }
              \glsmfublocker{\PGLSfmtlong}

  \pglsfmtlongpl

              \newcommand*{\pglsfmtlongpl}[1]{%
                \pglsprefix{#1}{prefixfirstplural}%
                \glsfmtlongpl{#1}%
              }

  \Pglsfmtlongpl

              \newcommand*{\Pglsfmtlongpl}[1]{%
               \glstexorpdfstring
                {\Pglsxtrtitlelongpl{#1}}%
                {\glspdfsentencecase
                  {%
                    \pglsprefix{#1}{prefixfirstplural}%
                    \glsentrylongpl{#1}%
                  }%
                }%
              }
              \glsmfuaddmap{\pglsfmtlongpl}{\Pglsfmtlongpl}

\Pglsxtrtitlelongpl

              \newrobustcmd*{\Pglsxtrtitlelongpl}[1]{%
                \glsxtr@title@field\Pglsxtrlongpl{#1}%
              }

  \PGLSfmtlongpl

              \newcommand*{\PGLSfmtlongpl}[1]{%
               \PGLSprefix{#1}{prefixfirstplural}%
               \GLSfmtlongpl{#1}%
              }
              \glsmfublocker{\PGLSfmtlongpl}
```

1.10   Multi (Combined/Compound) Entries

(I'd rather call these combined or compound entries but `\cgls` is already taken.)

New to version 1.48, the commands here provide a way of referencing multiple entries as a single unit. For example, biological organisms are often referred to by their genus and species, such as *Clostridium botulinum* and *Clostridium perfringens* (where the genus is Clostridium). The genus is often abbreviated after first use, regardless of which species in the genus is being referenced. For example, "*Clostridium botulinum* and *C. perfringens*". This can't be supported by any abbreviation styles unless the genus and species names are defined separately. For example:

```
%\setabbreviationstyle{long-only-short-only}
%\newabbreviation{clostridium}{C.}{Clostridium}
%\newglossaryentry{botulinum}{name={botulinum},description={}}
%\newglossaryentry{perfringens}{name={perfringens},description={}}
%
```

This means that the entries then need to be referenced using a rather cumbersome method:

```
%\gls{clostridium} \gls{botulinum} and \gls{clostridum}
%\gls{perfringens}
%
```

This section provides a command that will provide a way of defining a label that represents a combination of entries (which must all be first defined). For example:

```
%\multiglossaryentry{cbot}{clostridium,botulinum}
%
```

This label can then be referenced using `\mgls`, which internally uses `\gls` for each component. The last component in the list is considered the "main" component (not to be confused with the main glossary). If this isn't the case, the label of the main component should be added in the optional argument before the label list. Note that the multi-label (`cbot` in this case) can't be referenced using commands like `\gls`.

First define the general set of options that should be applied to all multi-entries. These can be set with:

`\multiglossaryentrysetup`

```
\newcommand*{\multiglossaryentrysetup}[1]{\setkeys{glsxtrcombined}{#1}}
```

`\@gls@combined@indexmain` Numeric value: 0=false (don't index main component), 1=true (always index main component), 2=first (only index main component on first use). Default: 1 (true);

```
\newcommand*{\@gls@combined@indexmain}{1}
\define@choicekey{glsxtrcombined}{indexmain}%
 [\@gls@combined@indexmain@val\@gls@combined@indexmain]
 {false,true,first}[true]{}
```

`\@gls@combined@indexothers` Numeric value: 0=false (don't index other components), 1=true (always index other components), 2=first (only index other components on first use). Default: 2 (first);

```
\newcommand*{\@gls@combined@indexothers}{2}
\define@choicekey{glsxtrcombined}{indexothers}%
 [\@gls@combined@indexothers@val\@gls@combined@indexothers]
 {false,true,first}[true]{}
```

`\@gls@combined@hyper` Numeric value: 0=none (\mgls doesn't create a hyperlink), 1=allmain (all content hyperlinks to the main component), 2=mainonly (only the main component has a hyperlink), 3=individual (each component has a hyperlink to their own target). Default: 3.

```
\newcommand*{\@gls@combined@hyper}{3}
\define@choicekey{glsxtrcombined}{hyper}%
 [\@gls@combined@hyper@val\@gls@combined@hyper]
 {none,allmain,mainonly,individual,otheronly,notmainfirst,nototherfirst,notfirst}{}
```

`\@gls@combined@encapmain` Location encap value for main component (corresponding to format key in \gls).

```
\newcommand*{\@gls@combined@encapmain}{glsnumberformat}
\define@key{glsxtrcombined}{encapmain}{%
 \renewcommand*{\@gls@combined@encapmain}{#1}%
}
```

`\@gls@combined@encapothers` Location encap value for other components (corresponding to format key in \gls).

```
\newcommand*{\@gls@combined@encapothers}{glsnumberformat}
\define@key{glsxtrcombined}{encapothers}{%
 \renewcommand*{\@gls@combined@encapothers}{#1}%
}
```

`\@gls@combined@textformat` Encapsulate entire content with the command identified by the given control sequence name.

```
\newcommand*{\@gls@combined@textformat}{@firstofone}
\define@key{glsxtrcombined}{textformat}{%
 \renewcommand*{\@gls@combined@textformat}{#1}%
}
```

`\@gls@combined@category` Assign a category to the combined set.

```
\newcommand*{\@gls@combined@category}{}
\define@key{glsxtrcombined}{category}{%
 \renewcommand*{\@gls@combined@category}{#1}%
}
```

Pre-options family:

```
\define@key{glsxtrcombinedpreopts}{category}{%
 \renewcommand*{\@gls@combined@category}{#1}%
}
```

\@gls@combined@mglsopts   Default options to pass to \mgls.

```
\newcommand*{\@gls@combined@mglsopts}{}
\define@key{glsxtrcombined}{mglsopts}{%
 \renewcommand*{\@gls@combined@mglsopts}{#1}%
}
\define@key{glsxtrcombinedpreopts}{mglsopts}{%
 \@gls@combined@mglsopts@do
 {%
   \renewcommand*{\@gls@combined@mglsopts}{#1}%
 }%
}
```

\@gls@combined@mglsopts@do

```
\newcommand*{\@gls@combined@mglsopts@do}[1]{#1}
```

\mgls@disable@mglsopts

```
\newcommand*{\mgls@disable@mglsopts}{%
\let\@gls@combined@mglsopts@do\@gls@combined@mglsopts@do@not
}
```

\mgls@enable@mglsopts

```
\newcommand*{\mgls@enable@mglsopts}{%
\let\@gls@combined@mglsopts@do\@firstofone
}
```

\@gls@combined@mglsopts@do

```
\newcommand*{\@gls@combined@mglsopts@do@not}[1]{%
  \PackageError{glossaries-extra}{'mglsopts' key not permitted inside
    'setup' value}{}%
}
```

\@gls@combined@firstprefix   Prefix for multi-entry first use.

```
\newcommand*{\@gls@combined@firstprefix}{}
\define@key{glsxtrcombined}{firstprefix}{%
 \renewcommand*{\@gls@combined@firstprefix}{#1}%
}
```

\@gls@combined@usedprefix   Prefix for multi-entry subsequent first use.

```
\newcommand*{\@gls@combined@usedprefix}{}
\define@key{glsxtrcombined}{usedprefix}{%
 \renewcommand*{\@gls@combined@usedprefix}{#1}%
}
```

\@gls@combined@firstsuffix   Suffix for multi-entry first use.

```
\newcommand*{\@gls@combined@firstsuffix}{}
\define@key{glsxtrcombined}{firstsuffix}{%
 \renewcommand*{\@gls@combined@firstsuffix}{#1}%
}
```

`\@gls@combined@usedsuffix` Suffix for multi-entry subsequent first use.

```
\newcommand*{\@gls@combined@usedsuffix}{}
\define@key{glsxtrcombined}{usedsuffix}{%
 \renewcommand*{\@gls@combined@usedsuffix}{#1}%
}
```

`\@gls@combined@firstskipmain` Skip the main element on first use (multi-entry first use not element first use).

```
\define@boolkey{glsxtrcombined}{firstskipmain}[true]{}
\KV@glsxtrcombined@firstskipmainfalse
```

`\@gls@combined@firstskipothers` Skip the other elements on first use (multi-entry first use not element first use).

```
\define@boolkey{glsxtrcombined}{firstskipothers}[true]{}
\KV@glsxtrcombined@firstskipothersfalse
```

`\@gls@combined@usedskipmain` Skip the main element on subsequent use (multi-entry subsequent use not element subsequent use).

```
\define@boolkey{glsxtrcombined}{usedskipmain}[true]{}
\KV@glsxtrcombined@usedskipmainfalse
```

`\@gls@combined@usedskipothers` Skip the other elements on subsequent use (multi-entry subsequent use not element subsequent use).

```
\define@boolkey{glsxtrcombined}{usedskipothers}[true]{}
\KV@glsxtrcombined@usedskipothersfalse
```

`\@gls@combined@postlinks` Determine whether or not to use the individual element post-link hooks.

```
\newcommand*{\@gls@combined@postlinks@nr}{0}
\define@choicekey{glsxtrcombined}{postlinks}%
 [\@gls@combined@postlinks@val\@gls@combined@postlinks@nr]
 {none,all,notlast,mainnotlast,mainonly,othernotlast,otheronly}{}
```

`\@gls@combined@mpostlink` Determine whether or not to use the multi-entry post-link hook.

```
\newcommand*{\@gls@combined@mpostlink@nr}{1}
\define@choicekey{glsxtrcombined}{mpostlink}%
 [\@gls@combined@mpostlink@val\@gls@combined@mpostlink@nr]
 {false,true,firstonly,usedonly}[true]{}
```

`\@gls@combined@mpostlinkelement` Determine which element to use for the post-link hook.

```
\newcommand*{\@gls@combined@mpostlinkelement@nr}{0}
\define@choicekey{glsxtrcombined}{mpostlinkelement}%
 [\@gls@combined@mpostlinkelement@val\@gls@combined@mpostlinkelement@nr]
 {last,main,custom}{}
```

`\glsxtrifmulti`

```
\newcommand*{\glsxtrifmulti}[3]{\ifcsdef{@gls@combined@#1@main}{#2}{#3}}
```

`\glsxtrmultimain`

```
\newcommand*{\glsxtrmultimain}[1]{\csuse{@gls@combined@#1@main}}
```

392

`\glsxtrmultilist`

```
\newcommand*{\glsxtrmultilist}[1]{\csuse{@gls@combined@#1@list}}
```

`\glsxtrmultitotalelements` Total number of elements.

```
\newcommand*{\glsxtrmultitotalelements}[1]{\csuse{@gls@combined@#1@total}}
```

`\glsxtrmultimainindex` Index of main element (starting from 1). If the main element is the last element in the list then this should equal the total number of elements.

```
\newcommand*{\glsxtrmultimainindex}[1]{\csuse{@gls@combined@#1@mainindex}}
```

`\glsxtrmultilastotherindex` Index of the last non-main element.

```
\newcommand*{\glsxtrmultilastotherindex}[1]{\csuse{@gls@combined@#1@lastotherindex}}
```

`\ifmultiglossaryentryglobal` Make definitions global.

```
\newif\ifmultiglossaryentryglobal
\multiglossaryentryglobalfalse
```

`\mglselementindex` Count register to keep track of the current element index.

```
\newcount\mglselementindex
```

> $\verb|\multiglossaryentry[|\langle options\rangle\verb|]{|\langle multi\text{-}label\rangle\verb|}[|\langle main\ label\rangle\verb|]|$
> $\verb|{|\langle label$
> $list\rangle\verb|}|$

`\multiglossaryentry`

Defines the label ⟨*multi-label*⟩ that can be used in `\mgls`.

```
\newrobustcmd{\multiglossaryentry}[1][]{%
 \def\@gls@combined@current@opts{#1}%
 \ifnum\@glsxtr@docdefval=1\relax
  \let\@multi@glossentry@donext\@defmultiglossaryentry
 \else
  \let\@multi@glossentry@donext\@multiglossaryentry
 \fi
 \@multi@glossentry@donext
}
```

`\@multiglossaryentry`

```
\newcommand*{\@multiglossaryentry}[1]{%
 \def\@gls@combined@current@label{#1}%
 \@multi@glossaryentry
}
```

`\@multi@glossaryentry` Check for existence.

```
\newcommand*{\@multi@glossaryentry}[2][]{%
 \ifcsdef{@gls@combined@\@gls@combined@current@label @main}%
 {\PackageError{glossaries-extra}%
   {Multi-entry label '\@gls@combined@current@label' already defined}%
   {}%
```

393

```
      }%
      {%
        \@multi@glossary@entry{#1}{#2}%
      }%
    }
```

\@defmultiglossaryentry Used if document definitions are on.

```
\newcommand*{\@defmultiglossaryentry}[1]{%
  \def\@gls@combined@current@label{#1}%
  \@def@multi@glossaryentry
}
```

\@def@multi@glossaryentry Used if document definitions are on.

```
\newcommand*{\@def@multi@glossaryentry}[2][]{%
  \let\@def@multi@glossaryentry@do\@multi@glossary@entry
  \ifundef\@glsxtr@docdefs@multilist
  {%
    \gdef\@glsxtr@docdefs@multilist{}%
    \listxadd
     {\@glsxtr@docdefs@multilist}{\expandonce\@gls@combined@current@label}%
  }%
  {%
    \xifinlist{\@gls@combined@current@label}{\@glsxtr@docdefs@multilist}%
    {%
      \PackageError{glossaries-extra}%
      {Multi-entry label '\@gls@combined@current@label' already defined}%
      {}%
      \let\@def@multi@glossaryentry@do\@gobbletwo
    }%
    {%
      \listxadd
       {\@glsxtr@docdefs@multilist}{\expandonce\@gls@combined@current@label}%
    }%
  }%
  \@def@multi@glossaryentry@do{#1}{#2}%
}
```

\@multi@glossary@doifexists

```
\newcommand*{\@multi@glossary@doifexists}{\glsdoifexists}
```

> \providemultiglossaryentry[⟨*options*⟩]{⟨*multi-label*⟩}[⟨*main
> label*⟩]{⟨*label
> list*⟩}

\providemultiglossaryentry

Defines a multi-entry unless it has already been defined.

```
\newrobustcmd{\providemultiglossaryentry}[2][]{%
  \def\@gls@combined@current@opts{#1}%
```

```
                      \def\@gls@combined@current@label{#2}%
                      \ifcsdef{@gls@combined@\@gls@combined@current@label @main}%
                      {\def\@multi@glossentry@donext{\@provide@multi@glossaryentry@noop}}%
                      {%
                        \ifnum\@glsxtr@docdefval=1\relax
                         \def\@multi@glossentry@donext{\@def@multi@glossaryentry}%
                        \else
                         \def\@multi@glossentry@donext{\@multi@glossaryentry}%
                        \fi
                      }%
                      \@multi@glossentry@donext
                     }
```

de@multi@glossaryentry@noop Do nothing.

```
                     \newcommand*{\@provide@multi@glossaryentry@noop}[2][]{}
```

\@multi@glossaryentry@list List of all defined multi-entry sets.

```
                     \newcommand*{\@multi@glossaryentry@list}{}
```

\@multi@glossary@entry

```
                     \newcommand*{\@multi@glossary@entry}[2]{%
                       \protected@edef\@gls@combined@current@main{#1}%
```

Fully expand list.

```
                       \protected@edef\@gls@combined@currentlist{#2}%
```

Count items in list, check they are all defined, and find last item at the same time.

```
                     \mglselementindex=0\relax
                     \@for\@gls@tmp:=\@gls@combined@currentlist\do{%
                       \advance\mglselementindex by 1\relax
                       \@multi@glossary@doifexists{\@gls@tmp}{}%
                       \let\@gls@combined@finalitem\@gls@tmp
                       \ifdefvoid\@gls@combined@current@main
                       {}%
                       {%
                         \ifx\@gls@combined@current@main\@gls@tmp
                           \ifmultiglossaryentryglobal
                             \global\cslet{@gls@combined@\@gls@combined@current@label @main}%
                                     \@gls@combined@current@main
                             \csxdef{@gls@combined@\@gls@combined@current@label @mainindex}%
                                     {\the\mglselementindex}%
                           \else
                             \cslet{@gls@combined@\@gls@combined@current@label @main}%
                                     \@gls@combined@current@main
                             \csedef{@gls@combined@\@gls@combined@current@label @mainindex}%
                                     {\the\mglselementindex}%
                           \fi
                         \else
                           \ifmultiglossaryentryglobal
```

```
            \csxdef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
                    {\the\mglselementindex}%
          \else
            \csedef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
                    {\the\mglselementindex}%
          \fi
        \fi
      }%
    }%
    \ifmultiglossaryentryglobal
      \csxdef{@gls@combined@\@gls@combined@current@label @total}%
            {\the\mglselementindex}%
    \else
      \csedef{@gls@combined@\@gls@combined@current@label @total}%
            {\the\mglselementindex}%
    \fi
    \ifnum\mglselementindex<2\relax
        \PackageError{glossaries-extra}{At least 2 labels required in
          multi-entry element list (\number\mglselementindex\space found)}{}%
    \else
      \ifdefvoid\@gls@combined@current@main
      {}%
      {%
```

If `\@gls@combined@`⟨*label*⟩`@main` hasn't been set then it wasn't included in the list.

```
        \ifcsundef{@gls@combined@\@gls@combined@current@label @main}%
        {\PackageError{glossaries-extra}%
         {Main element '\@gls@combined@current@main' not found in list}%
         {The final element '\@gls@combined@finalitem' will be used instead}
```

Set to empty so that the default (final element) is used instead.

```
          \let\@gls@combined@current@main\@empty
        }%
        {}%
      }%
      \ifdefvoid\@gls@combined@current@main
      {%
```

Set main to final element.

```
        \ifmultiglossaryentryglobal
          \global\cslet{@gls@combined@\@gls@combined@current@label @main}%
                \@gls@combined@finalitem
          \global\csletcs{@gls@combined@\@gls@combined@current@label @mainindex}%
            {@gls@combined@\@gls@combined@current@label @total}%
          \csxdef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
                {\the\numexpr\mglselementindex-1 }%
        \else
          \cslet{@gls@combined@\@gls@combined@current@label @main}%
                \@gls@combined@finalitem
          \csletcs{@gls@combined@\@gls@combined@current@label @mainindex}%
```

```
              {@gls@combined@\@gls@combined@current@label @total}%
          \csedef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
                  {\the\numexpr\mglselementindex-1 }%
        \fi
      }%
      {}%
      \ifmultiglossaryentryglobal
```
Globally define element list.
```
          \global\cslet{@gls@combined@\@gls@combined@current@label @list}%
              \@gls@combined@currentlist
```
Globally define options.
```
          \protected@csxdef{@gls@combined@\@gls@combined@current@label @options}%
           {\@gls@combined@current@opts}%
```
Global conditional definition.
```
          \expandafter\@ifdefinable
           \csname if@gls@combined@\@gls@combined@current@label @flag\endcsname
           {\expandafter\global\expandafter
             \newif\csname if@gls@combined@\@gls@combined@current@label @flag\endcsname}%
           \expandafter\global
             \csname @gls@combined@\@gls@combined@current@label @flagfalse\endcsname
        \else
```
Locally define element list.
```
          \cslet{@gls@combined@\@gls@combined@current@label @list}%
              \@gls@combined@currentlist
```
Locally define options.
```
          \protected@csedef{@gls@combined@\@gls@combined@current@label @options}%
           {\@gls@combined@current@opts}%
```
Local conditional definition.
```
          \newboolean{@gls@combined@\@gls@combined@current@label @flag}%
          \csname @gls@combined@\@gls@combined@current@label @flagfalse\endcsname
        \fi
      \fi
      \writemultiglossentry
       {\@gls@combined@current@opts}{\@gls@combined@current@label}%
       {\csuse{@gls@combined@\@gls@combined@current@label @main}}{#2}%
```
Append label to list.
```
    \ifmultiglossaryentryglobal
      \ifdefempty\@multi@glossaryentry@list
      {\let\@multi@glossaryentry@list\@gls@combined@current@label}%
      {%
        \eappto\@multi@glossaryentry@list{,\expandonce\@gls@combined@current@label}%
      }%
    \else
      \ifdefempty\@multi@glossaryentry@list
      {\global\let\@multi@glossaryentry@list\@gls@combined@current@label}%
```

```
        {%
          \xappto\@multi@glossaryentry@list{,\expandonce\@gls@combined@current@label}%
        }%
      \fi
    }
```

> $\backslash$@glsxtr@multientry$\{\langle options \rangle\}\{\langle multilabel \rangle\}\{\langle main \rangle\}\{\langle list \rangle\}$

\@glsxtr@multientry

Information for aux file. Useful for `bib2gls` and also for `docdef`.

```
\newcommand*{\@glsxtr@multientry}[4]{%
 \ifnum\@glsxtr@docdefval=1\relax
  \bgroup
    \def\@gls@combined@current@opts{#1}%
    \def\@gls@combined@current@label{#2}%
    \let\@multi@glossary@doifexists\@secondoftwo
    \let\writemultiglossentry\@gobblefour
    \multiglossaryentryglobaltrue
    \@multi@glossary@entry{#3}{#4}%
  \egroup
 \fi
}
```

\writemultiglossentry This can be redefined to do nothing if the information isn't required.

```
\newcommand*{\writemultiglossentry}[4]{%
  \protected@write\@auxout{}{\string\@glsxtr@multientry{#1}{#2}{#3}{#4}}%
}
```

\ifmglsused Determines whether or not the multi-entry set has been referenced by commands like \mgls or \mglsname.

```
\newcommand*{\ifmglsused}[3]{%
 \ifbool{@gls@combined@#1@flag}{#2}{#3}%
}
```

\mglsunset Unset the flag.

```
\newcommand*{\mglsunset}[1]{%
 \gls@ifnotmeasuring
 {%
    \glsxtrifmulti{#1}{\@mglsunset{#1}}%
    {%
      \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
      {You need to define '#1' with \string\multiglossaryentry}%
    }%
 }%
}
```

\@mglsunset

```
\newcommand*{\@mglsunset}[1]{%
```

```
              \expandafter\global\csname @gls@combined@#1@flagtrue\endcsname
              }
```

\mglsreset Unset the flag.

```
              \newcommand*{\mglsreset}[1]{%
              \gls@ifnotmeasuring
              {%
                 \glsxtrifmulti{#1}{\@mglsreset{#1}}%
                 {%
                   \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
                   {You need to define '#1' with \string\multiglossaryentry}%
                 }%
              }%
              }
```

\@mglsreset

```
              \newcommand*{\@mglsreset}[1]{%
              \expandafter\global\csname @gls@combined@#1@flagfalse\endcsname
              }
```

\mglslocalunset Unset the flag.

```
              \newcommand*{\mglslocalunset}[1]{%
              \gls@ifnotmeasuring
              {%
                 \glsxtrifmulti{#1}{\@mglslocalunset{#1}}%
                 {%
                   \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
                   {You need to define '#1' with \string\multiglossaryentry}%
                 }%
              }%
              }
```

\@mglslocalunset

```
              \newcommand*{\@mglslocalunset}[1]{%
              \csname @gls@combined@#1@flagtrue\endcsname
              }
```

\mglslocalreset Unset the flag.

```
              \newcommand*{\mglslocalreset}[1]{%
              \gls@ifnotmeasuring
              {%
                 \glsxtrifmulti{#1}{\@mglslocalreset{#1}}%
                 {%
                   \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
                   {You need to define '#1' with \string\multiglossaryentry}%
                 }%
              }%
              }
```

`\@mglslocalreset`

```
\newcommand*{\@mglslocalreset}[1]{%
 \csname @gls@combined@#1@flagfalse\endcsname
}
```

`\mglsunsetall` Unset all.

```
\newcommand*{\mglsunsetall}{%
  \@for\@mgls@thislabel:=\@multi@glossaryentry@list\do{\mglsunset\@mgls@thislabel}%
}%
```

`\mglsresetall` Reset all.

```
\newcommand*{\mglsresetall}{%
  \@for\@mgls@thislabel:=\@multi@glossaryentry@list\do{\mglsreset\@mgls@thislabel}%
}%
```

$\boxed{\text{\mglsSetName\{⟨\textit{multi-label}⟩\}\{⟨\textit{new main}⟩\}}}$

`\mglsSetMain`

Allow the main label to be changed (local).

```
\newrobustcmd{\mglsSetMain}[2]{%
  \ifcsundef{@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label '#1' not defined}{}}%
  {%
    \protected@edef\@gls@combined@current@main{#2}%
    \letcs\@gls@combined@currentlist{@gls@combined@#1@list}%
```

Check that the given label is in the list of elements and update main and last other element index.

```
    \mglselementindex=0\relax
    \count@=0\relax
    \@for\@gls@tmp:=\@gls@combined@currentlist\do{%
      \advance\mglselementindex by 1\relax
      \ifx\@gls@combined@current@main\@gls@tmp
        \count@=\mglselementindex\relax
        \let\@gls@combined@finalitem\@gls@tmp
        \ifmultiglossaryentryglobal
          \global\cslet{@gls@combined@#1@main}\@gls@combined@current@main
          \csxdef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \else
          \cslet{@gls@combined@#1@main}\@gls@combined@current@main
          \csedef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \fi
      \else
        \ifmultiglossaryentryglobal
          \csxdef{@gls@combined@#1@lastotherindex}{\the\mglselementindex}%
        \else
          \csedef{@gls@combined@#1@lastotherindex}{\the\mglselementindex}%
        \fi
      \fi
```

```
    }%
    \ifnum\count@=0\relax
      \PackageError{glossaries-extra}{Label '#2' is not in '#1' set
       (\@gls@combined@currentlist)}{}%
```

Default to final item.

```
      \ifmultiglossaryentryglobal
        \global\cslet{@gls@combined@#1@main}\@gls@combined@finalitem
        \csxdef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \csxdef{@gls@combined@#1@lastotherindex}{%
          \number\numexpr\mglselementindex-1 }%
      \else
        \cslet{@gls@combined@#1@main}\@gls@combined@finalitem
        \csedef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \csedef{@gls@combined@#1@lastotherindex}{%
          \number\numexpr\mglselementindex-1 }%
      \fi
    \fi
  }%
}
```

\mglsSetOptions

$\boxed{\texttt{\textbackslash mglsSetOptions\{}\langle\textit{multi-label}\rangle\texttt{\}\{}\langle\textit{new options}\rangle\texttt{\}}}$

Allow the options to be changed (local). No expansion is applied.

```
\newrobustcmd{\mglsSetOptions}[2]{%
  \ifcsundef{@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label '#1' not defined}{}}%
  {%
    \csdef{@gls@combined@#1@options}{#2}%
  }%
}
```

\mglsAddOptions

$\boxed{\texttt{\textbackslash mglsAddOptions\{}\langle\textit{multi-label}\rangle\texttt{\}\{}\langle\textit{extra options}\rangle\texttt{\}}}$

Allow the options to be changed (local). No expansion is applied.

```
\newrobustcmd{\mglsAddOptions}[2]{%
  \ifcsundef{@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label '#1' not defined}{}}%
  {%
    \ifcsempty{@gls@combined@#1@options}%
    {\csdef{@gls@combined@#1@options}{#2}}%
    {\csappto{@gls@combined@#1@options}{,#2}}%
  }%
}
```

Options for \mgls:

\@mgls@all Options to apply to all elements.

```
\newcommand*{\@mgls@all}{}
\define@key{mgls}{all}{\renewcommand*{\@mgls@all}{#1}}
```

\@mgls@main Options to apply to the main element only.

```
\newcommand*{\@mgls@main}{}
\define@key{mgls}{main}{\renewcommand*{\@mgls@main}{#1}}
```

\@mgls@others Options to apply to the other (no main) elements.

```
\newcommand*{\@mgls@others}{}
\define@key{mgls}{others}{\renewcommand*{\@mgls@others}{#1}}
```

\@mgls@setup Options to apply to \multiglossaryentrysetup.

```
\newcommand*{\@mgls@setup}{}
\define@key{mgls}{setup}{%
 \@mgls@setup@do{\renewcommand*{\@mgls@setup}{#1}}%
}
```

\@mgls@setup@do

```
\newcommand*{\@mgls@setup@do}[1]{#1}
```

\@mgls@setup@do@not

```
\newcommand*{\@mgls@setup@do@not}[1]{%
  \PackageError{glossaries-extra}{'setup' key not permitted inside
   'mglsopts' value}{}%
}
```

\mgls@disable@setup

```
\newcommand*{\mgls@disable@setup}{%
  \let\@mgls@setup@do\@mgls@setup@do@not
}
```

\mgls@enable@setup

```
\newcommand*{\mgls@enable@setup}{%
  \let\@mgls@setup@do\@firstofone
}
```

\@mgls@unsetaction

```
\newcommand\@mgls@unsetaction{0}
\define@choicekey{mgls}{multiunset}[\@mgls@unsetaction@val\@mgls@unsetaction]%
 {global,local,none}{}
```

\ifKV@mgls@presetlocal

```
\define@boolkey{mgls}{presetlocal}[true]{}
\KV@mgls@presetlocalfalse
```

`\@mgls@hyper`

```
\newcommand*{\@mgls@hyper}{}
\define@choicekey{mgls}{hyper}[\@mgls@hyper@val\@mgls@hyper@nr]{true,false}[true]%
{%
  \renewcommand*{\@mgls@hyper}{hyper=#1}%
  \ifnum\@mgls@hyper@nr=1\relax
   \let\@mgls@hyperlink\@secondoftwo
  \else
   \let\@mgls@hyperlink\@@mgls@hyperlink
  \fi
}
```

`\@@mgls@hyperlink`

```
\newcommand*{\@@mgls@hyperlink}[2]{%
  \ifx\@glslink\glsdonohyperlink
   #2%
  \else
   \glsxtr@org@dohyperlink{\glolinkprefix#1}{#2}%
  \fi
}
```

`\@mgls@hyperlink`

```
\let\@mgls@hyperlink\@@mgls@hyperlink
```

`\mglsforelements`

> $\boxed{\texttt{\textbackslash mglsforelements\{}\langle \mathit{multi\text{-}label}\rangle\texttt{\}\{}\langle \mathit{cs}\rangle\texttt{\}\{}\langle \mathit{body}\rangle\texttt{\}}}$

```
\newcommand*{\mglsforelements}[3]{%
  \expandafter\@for\expandafter#2\expandafter:\expandafter
    =\csname @gls@combined@#1@list\endcsname\do{#3}%
}
```

`\mglsforotherelements`

> $\boxed{\texttt{\textbackslash mglsforotherelements\{}\langle \mathit{multi\text{-}label}\rangle\texttt{\}\{}\langle \mathit{cs}\rangle\texttt{\}\{}\langle \mathit{body}\rangle\texttt{\}}}$

```
\newcommand*{\mglsforotherelements}[3]{%
  \expandafter\@for\expandafter#2\expandafter:\expandafter
    =\csname @gls@combined@#1@list\endcsname\do
  {\expandafter\ifdefequal\csname @gls@combined@#1@main\endcsname{#2}{}{#3}}%
}
```

`\mglsunsetothers`

```
\newcommand*{\mglsunsetothers}[1]{%
  \mglsforotherelements{#1}{\@gls@tmp}{\glsunset{\@gls@tmp}}%
}
```

`\mglslocalunsetothers`

```
\newcommand*{\mglslocalunsetothers}[1]{%
```

```
                    \mglsforotherelements{#1}{\@gls@tmp}{\glslocalunset{\@gls@tmp}}%
                    }
```

\mglselementreset

```
                    \newcommand*{\mglselementreset}[1]{%
                     \ifKV@mgls@presetlocal
                      \glslocalreset{#1}%
                     \else
                      \glsreset{#1}%
                     \fi
                    }
```

\mglselementunset

```
                    \newcommand*{\mglselementunset}[1]{%
                     \ifKV@mgls@presetlocal
                      \glslocalunset{#1}%
                     \else
                      \glsunset{#1}%
                     \fi
                    }
```

\@mgls@resetall

```
                    \newcommand*{\@mgls@resetall}{}
                    \define@choicekey{mgls}{resetall}%
                    [\@mgls@resetall@val\@mgls@resetall@nr]{false,true}[true]%
                    {%
                     \ifcase\@mgls@resetall@nr\relax
                       \renewcommand*{\@mgls@resetall}{}%
                     \or
                       \renewcommand*{\@mgls@resetall}{%
                         \@for\@gls@resetlabel:=\mglscurrentlist\do{\mglselementreset\@gls@resetlabel}}%
                       \renewcommand*{\@mgls@unsetall}{}%
                     \fi
                    }
```

\@mgls@resetmain

```
                    \newcommand*{\@mgls@resetmain}{}
                    \define@choicekey{mgls}{resetmain}
                    [\@mgls@resetmain@val\@mgls@resetmain@nr]{false,true}[true]%
                    {%
                     \ifcase\@mgls@resetmain@nr\relax
                       \renewcommand*{\@mgls@resetmain}{}%
                     \or
                       \renewcommand*{\@mgls@resetmain}{\mglselementreset\mglscurrentmainlabel}%
                       \renewcommand*{\@mgls@unsetmain}{}%
                     \fi
                    }
```

\@mgls@resetothers

```
\newcommand*{\@mgls@resetothers}{}
\define@choicekey{mgls}{resetothers}
[\@mgls@resetothers@val\@mgls@resetothers@nr]{false,true}[true]%
{%
 \ifcase\@mgls@resetothers@nr\relax
   \renewcommand*{\@mgls@resetothers}{}%
 \or
   \renewcommand*{\@mgls@resetothers}{%
     \@for\@gls@resetlabel:=\mglscurrentlist\do{%
       \ifx\@gls@resetlabel\mglscurrentmainlabel
       \else
         \mglselementreset\@gls@resetlabel
       \fi
     }%
   }%
   \renewcommand*{\@mgls@unsetothers}{}%
 \fi
}
```

\@mgls@unsetall

```
\newcommand*{\@mgls@unsetall}{}
\define@choicekey{mgls}{unsetall}%
[\@mgls@unsetall@val\@mgls@unsetall@nr]{false,true}[true]%
{%
 \ifcase\@mgls@unsetall@nr\relax
   \renewcommand*{\@mgls@unsetall}{}%
 \or
   \renewcommand*{\@mgls@unsetall}{%
     \@for\@gls@unsetlabel:=\mglscurrentlist\do{\mglselementunset\@gls@unsetlabel}}%
   \renewcommand*{\@mgls@resetall}{}%
 \fi
}
```

\@mgls@unsetmain

```
\newcommand*{\@mgls@unsetmain}{}
\define@choicekey{mgls}{unsetmain}
[\@mgls@unsetmain@val\@mgls@unsetmain@nr]{false,true}[true]%
{%
 \ifcase\@mgls@unsetmain@nr\relax
   \renewcommand*{\@mgls@unsetmain}{}%
 \or
   \renewcommand*{\@mgls@unsetmain}{\mglselementunset\mglscurrentmainlabel}%
   \renewcommand*{\@mgls@resetmain}{}%
 \fi
}
```

\@mgls@unsetothers

```
\newcommand*{\@mgls@unsetothers}{}
\define@choicekey{mgls}{unsetothers}
```

```
                     [\@mgls@unsetothers@val\@mgls@unsetothers@nr]{false,true}[true]%
                     {%
                      \ifcase\@mgls@unsetothers@nr\relax
                         \renewcommand*{\@mgls@unsetothers}{}%
                      \or
                         \renewcommand*{\@mgls@unsetothers}{%
                           \@for\@gls@unsetlabel:=\mglscurrentlist\do{%
                             \ifx\@gls@unsetlabel\mglscurrentmainlabel
                             \else
                                \mglselementunset\@gls@unsetlabel
                             \fi
                         }%
                      }%
                      \renewcommand*{\@mgls@resetothers}{}%
                      \fi
                     }
```

\glsxtr@setup@docurrent Set up the commands to determine whether or not to do the current element.

```
\newcommand{\glsxtr@setup@docurrent}{%
```

\mglscurrentlabel expands to the label of the current element. Should this
element be skipped?

```
\ifx\mglscurrentlabel\mglscurrentmainlabel
```

Main element. Should it be skipped?

```
\mglsisfirstuse
{%
  \ifKV@glsxtrcombined@firstskipmain
    \let\@mgls@do@current@element\@gobble
  \else
    \let\@mgls@do@current@element\@firstofone
  \fi
}%
{%
  \ifKV@glsxtrcombined@usedskipmain
    \let\@mgls@do@current@element\@gobble
  \else
    \let\@mgls@do@current@element\@firstofone
  \fi
}%
\else
```

Other element. Should it be skipped?

```
\mglsisfirstuse
{%
  \ifKV@glsxtrcombined@firstskipothers
    \let\@mgls@do@current@element\@gobble
  \else
    \let\@mgls@do@current@element\@firstofone
  \fi
}%
```

```
        {%
          \ifKV@glsxtrcombined@usedskipothers
            \let\@mgls@do@current@element\@gobble
          \else
            \let\@mgls@do@current@element\@firstofone
          \fi
        }%
      \fi
    }
```

glsxtr@mgls@checklastelement If the last element is skipped, \mglsiflast needs adjusting. The first argument should be either "first" or "used". The second argument is the multi-element label.

```
  \newcommand*{\glsxtr@mgls@checklastelement}[2]{%
    \ifbool{KV@glsxtrcombined@#1skipmain}%
    {%
      \ifbool{KV@glsxtrcombined@#1skipothers}%
      {%
```

This condition has already been checked for.

```
      }%
      {%
```

Main skipped. The last item will be the last other element.

```
        \ifnum\mglselementindex=\glsxtrmultilastotherindex{#2}\relax
          \let\mglsiflast\@firstoftwo
        \else
          \let\mglsiflast\@secondoftwo
        \fi
      }%
    }%
    {%
```

Main not skipped.

```
      \ifbool{KV@glsxtrcombined@#1skipothers}%
      {%
```

Others skipped. The main element is the only item.

```
        \ifnum\mglselementindex=\glsxtrmultimainindex{#2}\relax
          \let\mglsiflast\@firstoftwo
        \else
          \let\mglsiflast\@secondoftwo
        \fi
      }%
      {%
```

None skipped. This isn't the last element.

```
        \let\mglsiflast\@secondoftwo
      }%
    }%
  }
```

`\glsxtrmglsWarnAllSkipped` Warning if all elements are skipped. The first argument is the warning message, the second argument is the inserted content (final optional argument), the third command is the encapsulation command (which may be a hyperlink).

```
\newcommand{\glsxtrmglsWarnAllSkipped}[3]{%
  \GlossariesExtraWarning{#1}%
  #3{#2}%
}
```

`\glsxtr@mgls@applyopts`

```
\newcommand*{\glsxtr@mgls@applyopts}[1]{%
  \edef\@mgls@dooptions{\noexpand\setkeys*{mgls}{\expandonce#1}}%
  \@mgls@dooptions
```

Append any unknown options to all.

```
  \ifdefvoid\XKV@rm{}{\eappto\@mgls@all{,\expandonce\XKV@rm}}%
```

If setup key has been used, check for pre-option keys:

```
  \ifdefvoid\@mgls@setup
  {}%
  {%
    \edef\@mgls@dooptions{%
        \noexpand\setkeys*{glsxtrcombinedpreopts}{\expandonce\@mgls@setup}}%
    \mgls@disable@mglsopts
    \@mgls@dooptions
    \mgls@enable@mglsopts
```

Save remaining setup options.

```
    \ifx\@mgls@setupoptions\@empty
      \let\@mgls@setupoptions\XKV@rm
    \else
      \eappto\@mgls@setupoptions{,\expandonce\XKV@rm}%
    \fi
  }%
```

Apply gls unset/reset options.

```
  \@mgls@resetall
  \@mgls@unsetall
  \@mgls@resetmain
  \@mgls@unsetmain
  \@mgls@resetothers
  \@mgls@unsetothers
```

Disable.

```
  \let\@mgls@resetall\@empty
  \let\@mgls@resetmain\@empty
  \let\@mgls@resetothers\@empty
  \let\@mgls@unsetall\@empty
  \let\@mgls@unsetmain\@empty
  \let\@mgls@unsetothers\@empty
```

First use flags.

```
                \ifmglsused\mglscurrentmultilabel
                {\let\mglsisfirstuse\@secondoftwo}%
                {\let\mglsisfirstuse\@firstoftwo}%
              }
```

\@firstofthree

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

\@secondofthree

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

\@thirdofthree

```
                \providecommand{\@thirdofthree}[3]{#3}
```

The main internal command for referencing multi-entries:

\glsxtr@mgls@inner{⟨*options*⟩}{⟨*label*⟩}{⟨*insert*⟩}{⟨*first
cs*⟩}{⟨*not first cs*⟩}{⟨*main first cs*⟩}{⟨*main other cs*⟩}

\glsxtr@mgls@inner

```
              \newcommand*{\glsxtr@mgls@inner}[7]{%
              \let\mglslastmainlabel\@empty
              \let\mglsiflastmainwasfirstuse\@firstoftwo
              \let\mglsiflastmainwasplural\@secondoftwo
              \let\mglsiflastmaincapscase\@firstofthree
              \let\mglsiflastmainskipped\@firstoftwo
              \bgroup
              \ifcsundef{@gls@combined@#2@main}%
              {%
                \glsxtrundefaction{Multi entry '#2' hasn't been defined}%
                {You need to define '#2' with \string\multiglossaryentry}%
                \gdef\@mgls@post@hookdefs{%
                  \protected@edef\mglslastmultilabel{#2}%
                  \let\mglswasfirstuse\@firstoftwo
                  \let\mglslastcategory\@empty
                  \let\mglsiflastelementskipped\@firstoftwo
                  \let\mglsiflastelementwasfirstuse\@firstoftwo
                  \let\mglsiflastelementwasplural\@secondoftwo
                  \let\mglsiflastelementcapscase\@firstofthree
                  \let\mglslastelementlabel\@empty
                  \let\mgls@do@postlinkhook\relax
                }%
              }%
              {%
```

Initialise hooks in case component entries haven't been defined (which may
happen with bib2gls).

```
                \let\glsxtrifwasfirstuse\@firstoftwo
                \let\glsifplural\@secondoftwo
                \let\glscapscase\@firstofthree
```

409

Save information for hooks.

```
\protected@edef\mglscurrentmultilabel{#2}%
\letcs\mglscurrentmainlabel{@gls@combined@#2@main}%
\letcs\mglscurrentlist{@gls@combined@#2@list}%
\letcs\mglscurrentoptions{@gls@combined@#2@options}%
```

Initialise (may be changed if `multiunset` is present):

```
\ifmglsused\mglscurrentmultilabel
{\let\mglsisfirstuse\@secondoftwo}%
{\let\mglsisfirstuse\@firstoftwo}%
```

Only obtain pre-option keys:

```
\edef\@mgls@dooptions{%
    \noexpand\setkeys*{glsxtrcombinedpreopts}{\expandonce\mglscurrentoptions}}%
\@mgls@dooptions
```

Save remaining setup options.

```
\let\@mgls@setupoptions\XKV@rm
```

Apply `\mgls` options.

```
\mgls@disable@setup
\ifdefvoid\@gls@combined@mglsopts
{}%
{\glsxtr@mgls@applyopts\@gls@combined@mglsopts}%
\mgls@enable@setup
```

Apply options provided in #1.

```
\ifstrempty{#1}{}{\def\@mgls@options{#1}\glsxtr@mgls@applyopts\@mgls@options}%
```

Check for attribute settings.

```
\ifx\@gls@combined@category\empty
```

No category

```
\else
```

Attribute options:

```
\glshascategoryattribute{\@gls@combined@category}{multioptions}%
{%
  \letcs\@mgls@attroptions{@glsxtr@categoryattr@@\@gls@combined@category
      @multioptions}%
```

Only obtain pre-option keys:

```
\let\@gls@combined@mglsopts\@empty
\edef\@mgls@dooptions{%
  \noexpand\setkeys*{glsxtrcombinedpreopts}{\expandonce\@mgls@attroptions}}%
\@mgls@dooptions
```

Append remaining options:

```
\eappto\@mgls@setupoptions{,\expandonce\XKV@rm}%
\ifx\@gls@combined@mglsopts\@empty
\else
```

mgls options found:

```
        \let\@mgls@setup\@empty
        \mgls@disable@setup
        \glsxtr@mgls@applyopts\@gls@combined@mglsopts
        \mgls@enable@setup
      \fi
  }%
  {}%
\fi
```

Apply setup options.

```
\edef\@mgls@dooptions{%
    \noexpand\setkeys{glsxtrcombined}{\expandonce\@mgls@setupoptions}}%
\@mgls@dooptions
```

Provide local user-level access to category:

```
\let\mglscurrentcategory\@gls@combined@category
```

Should the entire content be a hyperlink?

```
\ifnum\@gls@combined@hyper=1\relax
  \def\@mgls@combinedlink{\@mgls@hyperlink{\mglscurrentmainlabel}}%
\else
  \def\@mgls@combinedlink{\@firstofone}%
\fi
```

Entire content encapsulator.

```
\def\@gls@combined@encapsulator##1{%
    \@mgls@combinedlink{\csuse{\@gls@combined@textformat}{##1}}}%
```

Initialise.

```
\let\@mgls@do@current@element\@firstofone
```

Check if all elements are being skipped.

```
\mglsisfirstuse
{%
  \ifKV@glsxtrcombined@firstskipmain
    \ifKV@glsxtrcombined@firstskipothers
```

Just do the warning and insert. This will ignore the loop.

```
      \let\@gls@org@combined@encapsulator\@gls@combined@encapsulator
      \def\@gls@combined@encapsulator##1{%
        \glsxtrmglsWarnAllSkipped{All elements skipped for
          first use of multi-entry '#2'}{#3}%
          {\@gls@org@combined@encapsulator}%
      }%
      \let\@mgls@do@current@element\@gobble
    \fi
  \fi
}%
{%
  \ifKV@glsxtrcombined@usedskipmain
    \ifKV@glsxtrcombined@usedskipothers
```

Just do the warning and insert. This will ignore the loop.

```
          \let\@gls@org@combined@encapsulator\@gls@combined@encapsulator
          \def\@gls@combined@encapsulator##1{%
            \glsxtrmglsWarnAllSkipped{All elements skipped for
              subsequent use of multi-entry '#2'}{#3}%
              {\@gls@org@combined@encapsulator}%
          }%
          \let\@mgls@do@current@element\@gobble
        \fi
      \fi
    }%
```

Determine prefix and suffix.

```
    \mglsisfirstuse
    {%
      \let\mglscurrentprefix\@gls@combined@firstprefix
      \let\mglscurrentsuffix\@gls@combined@firstsuffix
    }%
    {%
      \let\mglscurrentprefix\@gls@combined@usedprefix
      \let\mglscurrentsuffix\@gls@combined@usedsuffix
    }%
```

Set up post-link hook used after current scope.

```
    \xdef\@mgls@post@hookdefs{%
     \noexpand\def\noexpand\mglslastmultilabel{\expandonce\mglscurrentmultilabel}%
     \noexpand\def\noexpand\mglslastcategory{\mglscurrentcategory}%
    }%
    \ifx\@mgls@do@current@element\@gobble
      \gappto\@mgls@post@hookdefs{%
        \let\mglsiflastelementskipped\@firstoftwo
        \let\mglslastelementlabel\@empty
        \let\mglsiflastelementwasfirstuse\@firstoftwo
        \let\mglsiflastelementwasplural\@secondoftwo
        \let\mglsiflastelementcapscase\@firstofthree
      }%
    \fi
    \mglsisfirstuse
    {%
      \gappto\@mgls@post@hookdefs{\let\mglswasfirstuse\@firstoftwo}%
```

Determine if the multi-entry post-link hook should be applied.

```
      \ifcase\@gls@combined@mpostlink@nr\relax
```

mpostlink=false.

```
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
      \or
```

mpostlink=true.

```
        \ifcase\@gls@combined@mpostlinkelement@nr\relax
          \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
```

```
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
      \fi
    \or
```
mpostlink=firstonly.
```
      \ifcase\@gls@combined@mpostlinkelement@nr\relax
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
      \fi
    \or
```
mpostlink=usedonly.
```
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
    \fi
  }%
  {%
    \gappto\@mgls@post@hookdefs{\let\mglswasfirstuse\@secondoftwo}%
```
Determine if the multi-entry post-link hook should be applied.
```
    \ifcase\@gls@combined@mpostlink@nr\relax
```
mpostlink=false.
```
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
    \or
```
mpostlink=true.
```
      \ifcase\@gls@combined@mpostlinkelement@nr\relax
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
      \fi
    \or
```
mpostlink=firstonly.
```
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
    \or
```
mpostlink=usedonly.
```
      \ifcase\@gls@combined@mpostlinkelement@nr\relax
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
      \or
        \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
```

```
      \fi
    \fi
  }%
```
Save current post-link hook.

```
    \let\mgls@org@postlinkhook\glspostlinkhook
```
Prefix.

```
    \mglsprefix
```
Initialise last element label (for `\mglssuffix`).

```
    \let\mglslastelementlabel\@empty
    \@gls@combined@encapsulator
    {%
```
Save previous label.

```
    \def\@mgls@previouslabel{}%
    \mglselementindex=0\relax
    \@for\mglscurrentlabel:=\mglscurrentlist\do{%
     \advance\mglselementindex by 1\relax
     \glsxtr@setup@docurrent
```
Is this the last element?

```
    \ifx\@xfor@nextelement\@nnil
      \let\mglsiflast\@firstoftwo
    \else
      \let\mglsiflast\@secondoftwo
```
Are any elements being skipped?

```
    \mglsisfirstuse
    {%
     \glsxtr@mgls@checklastelement{first}{#2}%
    }%
    {%
     \glsxtr@mgls@checklastelement{used}{#2}%
    }%
    \fi
```
Should the element post-link hook be used?

```
    \ifcase\@gls@combined@postlinks@nr\relax
```
`postlinks=none`

```
    \let\glspostlinkhook\relax
    \or
```
`postlinks=all`

```
    \let\glspostlinkhook\mgls@org@postlinkhook
    \or
```
`postlinks=notlast`

```
    \mglsiflast
    {%
     \let\glspostlinkhook\relax
    }%
```

```
        {%
          \let\glspostlinkhook\mgls@org@postlinkhook
        }%
      \or
postlinks=mainnotlast
        \ifx\mglscurrentlabel\mglscurrentmainlabel
          \mglsiflast
          {%
            \let\glspostlinkhook\relax
          }%
          {%
            \let\glspostlinkhook\mgls@org@postlinkhook
          }%
        \else
          \let\glspostlinkhook\relax
        \fi
      \or
postlinks=mainonly
        \ifx\mglscurrentlabel\mglscurrentmainlabel
          \let\glspostlinkhook\mgls@org@postlinkhook
        \else
          \let\glspostlinkhook\relax
        \fi
      \or
postlinks=othernotlast
        \ifx\mglscurrentlabel\mglscurrentmainlabel
          \let\glspostlinkhook\relax
        \else
          \mglsiflast
          {%
            \let\glspostlinkhook\relax
          }%
          {%
            \let\glspostlinkhook\mgls@org@postlinkhook
          }%
        \fi
      \or
postlinks=otheronly
        \ifx\mglscurrentlabel\mglscurrentmainlabel
          \let\glspostlinkhook\relax
        \else
          \let\glspostlinkhook\mgls@org@postlinkhook
        \fi
      \fi
```

Save the last element for the multi-entry post-link hook.

```
      \mglsiflast
      {%
```

```
            \xappto\@mgls@post@hookdefs{%
              \noexpand\def\noexpand\mglslastelementlabel
                {\expandonce\mglscurrentlabel}}%
        }%
        {}%
```
Do current element:
```
        \@mgls@do@current@element
        {%
```
Pre element hook.
```
            \mglselementprehook
```
Is this the first use of the current element?
```
            \GlsXtrIfUnusedOrUndefined{\mglscurrentlabel}%
            {\let\@mgls@current@iffirstuse\@firstoftwo}%
            {\let\@mgls@current@iffirstuse\@secondoftwo}%
            \ifx\mglscurrentlabel\mglscurrentmainlabel
```
Main element. Location encap option:
```
              \edef\@mgls@current@options{format=\@gls@combined@encapmain}%
```
Indexing option:
```
            \ifcase\@gls@combined@indexmain
             \appto\@mgls@current@options{,noindex}%
            \or
             \appto\@mgls@current@options{,noindex=false}%
            \or
             \@mgls@current@iffirstuse
             {\appto\@mgls@current@options{,noindex=false}}%
             {\appto\@mgls@current@options{,noindex}}%
            \fi
```
Hyperlink option:
```
            \ifcase\@gls@combined@hyper\relax
              \appto\@mgls@current@options{,hyper=false}% none
            \or
              \appto\@mgls@current@options{,hyper=false}% allmain
            \or
              \eappto\@mgls@current@options{,\@mgls@hyper}% mainonly
            \or
              \eappto\@mgls@current@options{,\@mgls@hyper}% individual
            \or
              \appto\@mgls@current@options{,hyper=false}% otheronly
            \or
             \mglsisfirstuse
             {%
              \appto\@mgls@current@options{,hyper=false}% notmainfirst
             }%
             {%
              \eappto\@mgls@current@options{,\@mgls@hyper}% notmainfirst
             }%
```

```
\or
 \eappto\@mgls@current@options{,\@mgls@hyper}% nototherfirst
\or
 \mglsisfirstuse
 {%
 \appto\@mgls@current@options{,hyper=false}% notfirst
 }%
 {%
 \eappto\@mgls@current@options{,\@mgls@hyper}% notfirst
 }%
\fi
```

Append all and then main:

```
\eappto\@mgls@current@options{,\@mgls@all,\@mgls@main}%
\else
```

Other element. Location encap option:

```
\edef\@mgls@current@options{format=\@gls@combined@encapothers}%
```

Indexing option:

```
\ifcase\@gls@combined@indexothers\relax
 \appto\@mgls@current@options{,noindex}%
\or
 \appto\@mgls@current@options{,noindex=false}%
\or
 \@mgls@current@iffirstuse
 {\appto\@mgls@current@options{,noindex=false}}%
 {\appto\@mgls@current@options{,noindex}}%
\fi
```

Hyperlink option:

```
\ifcase\@gls@combined@hyper\relax
 \appto\@mgls@current@options{,hyper=false}% none
\or
 \appto\@mgls@current@options{,hyper=false}% allmain
\or
 \appto\@mgls@current@options{,hyper=false}% mainonly
\or
 \eappto\@mgls@current@options{,\@mgls@hyper}% individual
\or
 \eappto\@mgls@current@options{,\@mgls@hyper}% otheronly
\or
 \eappto\@mgls@current@options{,\@mgls@hyper}% notmainfirst
\or
 \mglsisfirstuse
 {%
 \appto\@mgls@current@options{,hyper=false}% nototherfirst
 }%
 {%
 \eappto\@mgls@current@options{,\@mgls@hyper}% nototherfirst
 }%
```

```
       \or
       \mglsisfirstuse
       {%
        \appto\@mgls@current@options{,hyper=false}% notfirst
       }%
       {%
        \eappto\@mgls@current@options{,\@mgls@hyper}% notfirst
       }%
       \fi
```
Append all and then others:
```
       \eappto\@mgls@current@options{,\@mgls@all,\@mgls@others}%
    \fi
```
Is this the first element?
```
       \ifx\@mgls@previouslabel\empty
       \ifx\mglscurrentlabel\mglscurrentmainlabel
         \let\@mgls@cs#6\relax
       \else
         \let\@mgls@cs#4\relax
       \fi
       \else
```
Not the first element so add separator.
```
       \@mgls@previous@iffirstuse
       {%
         \@mgls@current@iffirstuse
         {\glscombinedfirstsepfirst{\@mgls@previouslabel}{\mglscurrentlabel}}%
         {\glscombinedfirstsep{\@mgls@previouslabel}{\mglscurrentlabel}}%
       }%
       {%
         \@mgls@current@iffirstuse
         {\glscombinedsepfirst{\@mgls@previouslabel}{\mglscurrentlabel}}%
         {\glscombinedsep{\@mgls@previouslabel}{\mglscurrentlabel}}%
       }%
       \ifx\mglscurrentlabel\mglscurrentmainlabel
         \let\@mgls@cs#7\relax
       \else
         \let\@mgls@cs#5\relax
       \fi
       \fi
```
Is this the last element?
```
       \mglsiflast
       {\expandafter\@mgls@cs\expandafter{\@mgls@current@options}{\mglscurrentlabel}[#3]}%
       {\expandafter\@mgls@cs\expandafter{\@mgls@current@options}{\mglscurrentlabel}[]}%
```
Is this the main element? If so, save information for post-link hook.
```
       \ifx\mglscurrentlabel\mglscurrentmainlabel
       \xappto\@mgls@post@hookdefs{%
         \noexpand\def\noexpand\mglslastmainlabel
           {\expandonce\mglscurrentmainlabel}%
```

```
      }%
      \glsxtrifwasfirstuse
      {%
       \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasfirstuse\@firstoftwo}%
      }%
      {%
       \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasfirstuse\@secondoftwo}%
      }%
      \glsifplural
      {%
       \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasplural\@firstoftwo}%
      }%
      {%
       \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasplural\@secondoftwo}%
      }%
      \glscapscase
      {%
         \gappto\@mgls@post@hookdefs{%
           \let\mglsiflastmaincapscase\@firstofthree
         }%
      }%
      {%
         \gappto\@mgls@post@hookdefs{%
           \let\mglsiflastmaincapscase\@secondofthree
         }%
      }%
      {%
         \gappto\@mgls@post@hookdefs{%
           \let\mglsiflastmaincapscase\@thirdofthree
         }%
      }%
      \fi
      \let\@mgls@previouslabel\mglscurrentlabel
      \let\@mgls@previous@iffirstuse\@mgls@current@iffirstuse
     }%
```

Post element hook.

```
      \mglselementposthook
   }%
   \ifx\mglslastmainlabel\@empty
    \gappto\@mgls@post@hookdefs{\let\mglsiflastmainskipped\@firstoftwo}%
   \else
    \gappto\@mgls@post@hookdefs{\let\mglsiflastmainskipped\@secondoftwo}%
   \fi
```

Encapsulator may introduce grouping so check here.

```
   \ifx\@mgls@do@current@element\@gobble
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementskipped\@firstoftwo}%
   \else
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementskipped\@secondoftwo}%
   \fi
```

```
    \glsxtrifwasfirstuse
    {%
     \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@firstoftwo}%
    }%
    {%
     \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@secondoftwo}%
    }%
    \glsifplural
    {%
     \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasplural\@firstoftwo}%
    }%
    {%
     \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasplural\@secondoftwo}%
    }%
    \glscapscase
    {%
       \gappto\@mgls@post@hookdefs{%
         \let\mglsiflastelementcapscase\@firstofthree
       }%
    }%
    {%
       \gappto\@mgls@post@hookdefs{%
         \let\mglsiflastelementcapscase\@secondofthree
       }%
    }%
    {%
       \gappto\@mgls@post@hookdefs{%
         \let\mglsiflastelementcapscase\@thirdofthree
       }%
    }%
    }%
```

Suffix needs post-link hook commands.

```
    \@mgls@post@hookdefs
    \mglssuffix
```

Unset multi-entry first use flag after current scope.

```
    \ifcase\@mgls@unsetaction\relax
      \xappto\@mgls@post@hookdefs{%
        \noexpand\mglsunset{\expandonce\mglscurrentmultilabel}}%
    \or
      \xappto\@mgls@post@hookdefs{%
        \noexpand\mglslocalunset{\expandonce\mglscurrentmultilabel}}%
    \fi
  }%
  \glsxtrmglswrite{#2}%
  \egroup
  \@mgls@post@hookdefs
  \mgls@do@postlinkhook
}
```

\mglscustompostlinkhook

```
\newcommand*{\mglscustompostlinkhook}{}
```

\mglslastelementpostlinkhook

```
\newcommand*{\mglslastelementpostlinkhook}{%
 \let\glsxtrifwasfirstuse\mglsiflastelementwasfirstuse
 \let\glsifplural\mglsiflastelementwasplural
 \let\glscapscase\mglsiflastelementcapscase
 \let\glslabel\mglslastelementlabel
 \glspostlinkhook
}
```

\mglslastmainpostlinkhook

```
\newcommand*{\mglslastmainpostlinkhook}{%
 \let\glsxtrifwasfirstuse\mglsiflastmainwasfirstuse
 \let\glsifplural\mglsiflastmainwasplural
 \let\glscapscase\mglsiflastmaincapscase
 \let\glslabel\mglslastmainlabel
 \glspostlinkhook
}
```

\mglsdefcategoryprefix

```
\newcommand*{\mglsdefcategoryprefix}[2]{%
 \csdef{mglsprefix@#1}{#2}%
}
```

\mglshascategoryprefix

```
\newcommand*{\mglshascategoryprefix}[3]{%
 \ifcsdef{mglsprefix@#1}{#2}{#3}%
}
```

\mglsusecategoryprefix

```
\newcommand*{\mglsusecategoryprefix}[1]{%
 \csuse{mglsprefix@#1}%
}
```

\mglsprefix

```
\newcommand*{\mglsprefix}{%
 \ifdefempty\mglscurrentcategory
 {\mglscurrentprefix}%
 {%
   \mglshascategoryprefix{\mglscurrentcategory}%
   {\mglsusecategoryprefix{\mglscurrentcategory}}%
   {\mglscurrentprefix}%
 }%
}
```

\mglsdefcategorysuffix

```
\newcommand*{\mglsdefcategorysuffix}[2]{%
 \csdef{mglssuffix@#1}{#2}%
}
```

\mglshascategorysuffix

```
\newcommand*{\mglshascategorysuffix}[3]{%
 \ifcsdef{mglssuffix@#1}{#2}{#3}%
}
```

\mglsusecategorysuffix

```
\newcommand*{\mglsusecategorysuffix}[1]{%
 \csuse{mglssuffix@#1}%
}
```

\mglssuffix

```
\newcommand*{\mglssuffix}{%
 \ifdefempty\mglscurrentcategory
 {\ifdefempty{\mglscurrentsuffix}{}{\space(\mglscurrentsuffix)}}%
 {%
   \mglshascategorysuffix{\mglscurrentcategory}%
   {\mglsusecategorysuffix{\mglscurrentcategory}}%
   {\ifdefempty{\mglscurrentsuffix}{}{\space(\mglscurrentsuffix)}}%
 }%
}
```

\mglselementprehook

```
\newcommand*{\mglselementprehook}{}
```

\mglselementposthook

```
\newcommand*{\mglselementposthook}{}
```

Separators.

\glscombinedsep Separator between two elements that have been marked as used. This takes the two element labels as arguments.

```
\newcommand*{\glscombinedsep}[2]{%
 \glshasattribute{#1}{combinedsep}%
 {\glsgetattribute{#1}{combinedsep}}%
 { }%
}
```

\glscombinedfirstsepfirst Separator following and preceding a first use.

```
\newcommand*{\glscombinedfirstsepfirst}[2]{%
 \glshasattribute{#1}{combinedfirstsepfirst}%
 {\glsgetattribute{#1}{combinedfirstsepfirst}}%
 {\glscombinedsep{#1}{#2}}%
}
```

\glscombinedfirstsep Separator following a first use.

```
\newcommand*{\glscombinedfirstsep}[2]{%
  \glshasattribute{#1}{combinedfirstsep}%
  {\glsgetattribute{#1}{combinedfirstsep}}%
  {\glscombinedsep{#1}{#2}}%
}
```

\glscombinedsepfirst Separator preceding a first use.

```
\newcommand*{\glscombinedsepfirst}[2]{%
  \glshasattribute{#1}{combinedsepfirst}%
  {\glsgetattribute{#1}{combinedsepfirst}}%
  {\glscombinedsep{#1}{#2}}%
}
```

\glssetcombinedsepabbrvnbsp Provide shortcut for using non-breakable space following an abbreviation that has already been used.

```
\newcommand*{\glssetcombinedsepabbrvnbsp}{%
 \renewcommand*{\glscombinedsep}[2]{%
 \glshasattribute{##1}{combinedsep}%
 {\glsgetattribute{##1}{combinedsep}}%
 {\ifglshasshort{##1}{~}{ }}%
 }%
 \renewcommand*{\glscombinedsepfirst}[2]{%
 \glshasattribute{##1}{combinedsepfirst}%
 {\glsgetattribute{##1}{combinedsepfirst}}%
 {\ifglshasshort{##1}{~}{ }}%
 }%
 \renewcommand*{\glscombinedfirstsep}[2]{%
 \glshasattribute{##1}{combinedfirstsep}%
 {\glsgetattribute{##1}{combinedfirstsep}}%
 { }%
 }%
 \renewcommand*{\glscombinedfirstsepfirst}[2]{%
 \glshasattribute{##1}{combinedfirstsepfirst}%
 {\glsgetattribute{##1}{combinedfirstsepfirst}}%
 { }%
 }%
 }
```

\glssetcombinedsepabbrvnone Provide shortcut for using nothing if either on next use are abbreviations (otherwise use space).

```
\newcommand*{\glssetcombinedsepabbrvnone}{%
 \renewcommand*{\glscombinedsep}[2]{%
   \glshasattribute{##1}{combinedsep}%
   {\glsgetattribute{##1}{combinedsep}}%
   {\ifglshasshort{##1}{}{\ifglshasshort{##2}{}{ }}}%
 }%
 \renewcommand*{\glscombinedsepfirst}[2]{%
   \glshasattribute{##1}{combinedsepfirst}%
```

```
    {\glsgetattribute{##1}{combinedsepfirst}}%
    {\ifglshasshort{##1}{}{ }}%
 }%
 \renewcommand*{\glscombinedfirstsep}[2]{%
  \glshasattribute{##1}{combinedfirstsep}%
  {\glsgetattribute{##1}{combinedfirstsep}}%
  {\ifglshasshort{##2}{}{ }}%
 }%
 \renewcommand*{\glscombinedfirstsepfirst}[2]{%
  \glshasattribute{##1}{combinedfirstsepfirst}%
  {\glsgetattribute{##1}{combinedfirstsepfirst}}%
  { }%
 }%
}
```

\glssetcombinedsepnarrow Measures both.

```
\newcommand*{\glssetcombinedsepnarrow}[2]{%
\renewcommand*{\glscombinedsep}[2]{%
 \glshasattribute{##1}{combinedsep}%
 {\glsgetattribute{##1}{combinedsep}}%
 {%
  \ifglshasshort{##1}%
  {\glsmeasurewidth{\dimen@}{\glsentryshort{##1}}}%
  {\glsmeasurewidth{\dimen@}{\glsentrytext{##1}}}%
  \ifdim\dimen@<#1\relax
   #2%
  \else
    \ifglshasshort{##2}%
    {\glsmeasurewidth{\dimen@}{\glsentryshort{##2}}}%
    {\glsmeasurewidth{\dimen@}{\glsentrytext{##2}}}%
    \ifdim\dimen@<#1\relax
     #2%
    \else
     \space
    \fi
  \fi
 }%
}%
\renewcommand*{\glscombinedsepfirst}[2]{%
 \glshasattribute{##1}{combinedsepfirst}%
 {\glsgetattribute{##1}{combinedsepfirst}}%
 {%
  \ifglshasshort{##1}%
  {\glsmeasurewidth{\dimen@}{\glsentryshort{##1}}}%
  {\glsmeasurewidth{\dimen@}{\glsentrytext{##1}}}%
  \ifdim\dimen@<#1\relax
   #2%
  \else
    \ifhaslong{##2}%
    {\glsmeasurewidth{\dimen@}{\glsentrylong{##2}}}%
```

```
     {\glsmeasurewidth{\dimen@}{\glsentryfirst{##2}}}%
     \ifdim\dimen@<#1\relax
      #2%
     \else
      \space
     \fi
  \fi
 }%
}%
\renewcommand*{\glscombinedfirstsep}[2]{%
 \glshasattribute{##1}{combinedfirstsep}%
 {\glsgetattribute{##1}{combinedfirstsep}}%
 {%
   \ifhaslong{##1}%
   {\glsmeasurewidth{\dimen@}{\glsentrylong{##1}}}%
   {\glsmeasurewidth{\dimen@}{\glsentryfirst{##1}}}%
   \ifdim\dimen@<#1\relax
    #2%
   \else
     \ifglshasshort{##2}%
     {\glsmeasurewidth{\dimen@}{\glsentryshort{##2}}}%
     {\glsmeasurewidth{\dimen@}{\glsentrytext{##2}}}%
     \ifdim\dimen@<#1\relax
      #2%
     \else
      \space
     \fi
   \fi
 }%
}%
\renewcommand*{\glscombinedfirstsepfirst}[2]{%
 \glshasattribute{##1}{combinedfirstsepfirst}%
 {\glsgetattribute{##1}{combinedfirstsepfirst}}%
 {%
   \ifhaslong{##1}%
   {\glsmeasurewidth{\dimen@}{\glsentrylong{##1}}}%
   {\glsmeasurewidth{\dimen@}{\glsentryfirst{##1}}}%
   \ifdim\dimen@<#1\relax
    #2%
   \else
     \ifhaslong{##2}%
     {\glsmeasurewidth{\dimen@}{\glsentrylong{##2}}}%
     {\glsmeasurewidth{\dimen@}{\glsentryfirst{##2}}}%
     \ifdim\dimen@<#1\relax
      #2%
     \else
      \space
     \fi
   \fi
 }%
}%
```

```
   }%
 }
```

\@glsxtr@mglswrite Write information to the aux file for `bib2gls` to pick up, but only need to do it once per label since it only indicates which multi-entry has been referenced without any additional information.

```
\newcommand{\glsxtrmglswrite}[1]{%
 \ifx\@glsxtr@record@setting\@glsxtr@record@setting@off
 \else
  \protected@edef\@glsxtr@mglslabel{#1}%
  \ifdef\@glsxtr@mglsreflist
  {%
    \expandafter\DTLifinlist\expandafter{\@glsxtr@mglslabel}%
    {\@glsxtr@mglsreflist}{}%
    {%
      \xappto\@glsxtr@mglsreflist{,\expandonce\@glsxtr@mglslabel}%
      \if@mgls@writeseparaterefs
        \protected@write\@auxout{}{\string\@glsxtr@mglsrefs{#1}}%
      \fi
    }%
  }%
  {%
    \global\let\@glsxtr@mglsreflist\@glsxtr@mglslabel
    \if@mgls@writeseparaterefs
      \protected@write\@auxout{}{\string\@glsxtr@mglsrefs{#1}}%
    \else
```

Bug fix #262: `\immediate\protected@write` doesn't work in end document hook when `tikz` loaded. No real need for `\protected@write` as `\@glsxtr@mglsreflist` is just a comma-separated list of labels, but use `\expandonce` in case labels contain UTF-8 characters.

```
      \AtEndDocument{\immediate\write\@auxout
        {\string\@glsxtr@mglsrefs{\expandonce{\@glsxtr@mglsreflist}}}}%
    \fi
    \@mgls@disable@writeseparateref@cond
  }%
 \fi
 }
```

\@glsxtr@mglsrefs

```
\newcommand{\@glsxtr@mglsrefs}[1]{}
```

\if@mgls@writeseparaterefs If this conditional is changed, it must be done before the first instance of any `\mgls`-like command.

```
\newif\if@mgls@writeseparaterefs \@mgls@writeseparaterefsfalse
```

\mglsWriteSeparateRefsTrue

```
\newcommand{\mglsWriteSeparateRefsTrue}{\global\@mgls@writeseparaterefstrue}
```

```
\newcommand{\mglsWriteSeparateRefsFalse}{\global\@mgls@writeseparaterefsfalse}
```

```
\newcommand*{\@mgls@disable@writeseparateref@cond}{%
  \gdef\mglsWriteSeparateRefsTrue{\PackageError{glossaries-extra}%
   {Too late to use \string\mglsWriteSeparateRefsTrue}%
   {\string\mglsWriteSeparateRefsTrue\space can only be used before
    the first instance of any \string\mgls-like command}}%
  \gdef\mglsWriteSeparateRefsFalse{\PackageError{glossaries-extra}%
   {Too late to use \string\mglsWriteSeparateRefsFalse}%
   {\string\mglsWriteSeparateRefsFalse\space can only be used before
    the first instance of any \string\mgls-like command}}%
}
```

```
\newcommand{\glsxtr@newmgls}[6][]{%
  \edef\@glsxr@newmgls@do{%
  \noexpand\newrobustcmd*{\expandonce{\csname #2\endcsname}}%
   {\noexpand\@gls@hyp@opt\expandonce{\csname ns@glsxtr@#2\endcsname}}%
  \noexpand\newcommand*{\expandonce{\csname ns@glsxtr@#2\endcsname}}[2][]{%
   \noexpand\new@ifnextchar[%
    {\expandonce{\csname glsxtr@#2\endcsname}{####1}{####2}}%
    {\expandonce{\csname glsxtr@#2\endcsname}{####1}{####2}[]}%
  }%
  \noexpand\def\expandonce{\csname glsxtr@#2\endcsname}####1####2[####3]{%
    \noexpand\@glsxtr@mgls@linkdefs{\unexpanded{#1}}%
    \noexpand\def\noexpand\glsxtrcurrentmglscsname{#2}%
    \noexpand\glsxtr@mgls@inner{####1}{####2}{####3}%
     {\noexpand#3}{\noexpand#4}{\noexpand#5}{\noexpand#6}%
  }%
 }%
 \@glsxr@newmgls@do
 \ifx\@glsxtr@record@setting\@glsxtr@record@setting@off
 \else
```

Provide a way for `bib2gls` to recognise the command (this will make it easier
to add extra commands without having to modify `bib2gls`).

```
 \ifdef\@glsxtr@mglslikelist
 {\xappto\@glsxtr@mglslikelist{,#2}}%
 {%
   \gdef\@glsxtr@mglslikelist{#2}%
   \AtEndDocument{\immediate\protected@write\@auxout{}%
     {\string\@glsxtr@mglslike{\@glsxtr@mglslikelist}}}%
 }%
 \fi
}
```

```
\newcommand{\@glsxtr@mgls@linkdefs}[1]{%
 \ifstrempty{#1}%
  {%
    \let\glsxtrifwasglslike\@firstoftwo
    \def\glsxtrcurrentfield{}%
  }%
  {%
    \let\glsxtrifwasglslike\@secondoftwo
    \def\glsxtrcurrentfield{#1}%
  }%
}
```

\@glsxtr@mglslike

```
\newcommand*{\@glsxtr@mglslike}[1]{}
```

\GlsXtrMglsOrGls{⟨*mgls cs*⟩}{⟨*gls cs*⟩}⟨*modifier*⟩[⟨*options*⟩]
{⟨*label*⟩}[⟨*insert*⟩]

\GlsXtrMglsOrGls

```
\newcommand*{\GlsXtrMglsOrGls}[2]{%
  \def\@glsxtr@mgls@or@gls@mcs{#1}%
  \def\@glsxtr@mgls@or@gls@gcs{#2}%
  \@ifstar{\s@GlsXtrMglsOrGls}%
  {%
   \@ifnextchar+{\@firstoftwo{\p@GlsXtrMglsOrGls}}%
   {%
     \ifdefempty\@gls@alt@hyp@opt@char\@GlsXtrMglsOrGls\alt@GlsXtrMglsOrGls
   }%
  }%
}
```

\alt@GlsXtrMglsOrGls

```
\newcommand*{\alt@GlsXtrMglsOrGls}{
 \expandafter\@ifnextchar\@gls@alt@hyp@opt@char
 {\@firstoftwo{\@alt@GlsXtrMglsOrGls}}{\@GlsXtrMglsOrGls}%
}
```

\@GlsXtrMglsOrGls

```
\newcommand*{\@GlsXtrMglsOrGls}[2][]{%
 \glsxtrifmulti{#2}%
 {\@glsxtr@mgls@or@gls@mcs[#1]{#2}}%
 {\@glsxtr@mgls@or@gls@gcs[#1]{#2}}%
}
```

\s@GlsXtrMglsOrGls

```
\newcommand*{\s@GlsXtrMglsOrGls}[2][]{%
 \glsxtrifmulti{#2}%
```

```
                    {\@glsxtr@mgls@or@gls@mcs*[#1]{#2}}%
                    {\@glsxtr@mgls@or@gls@gcs*[#1]{#2}}%
                }
```

\p@GlsXtrMglsOrGls

```
                \newcommand*{\p@GlsXtrMglsOrGls}[2][]{%
                    \glsxtrifmulti{#2}%
                    {\@glsxtr@mgls@or@gls@mcs+[#1]{#2}}%
                    {\@glsxtr@mgls@or@gls@gcs+[#1]{#2}}%
                }
```

\@alt@GlsXtrMglsOrGls

```
                \newcommand*{\@alt@GlsXtrMglsOrGls}[2][]{%
                    \glsxtrifmulti{#2}%
                    {\expandafter\@glsxtr@mgls@or@gls@mcs\@gls@alt@hyp@opt@char[#1]{#2}}%
                    {\expandafter\@glsxtr@mgls@or@gls@gcs\@gls@alt@hyp@opt@char[#1]{#2}}%
                }
```

> \mgls[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

\mgls

Use \gls for all elements.

```
                \glsxtr@newmgls{mgls}{\@gls@}{\@gls@}{\@gls@}{\@gls@}%
```

> \mglspl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

\mglspl

Use \glspl for all elements.

```
                \glsxtr@newmgls{mglspl}{\@glspl@}{\@glspl@}{\@glspl@}{\@glspl@}%
```

> \mglsmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

\mglsmainpl

Only use \glspl for the main element, otherwise use \gls.

```
                \glsxtr@newmgls{mglsmainpl}{\@gls@}{\@gls@}{\@glspl@}{\@glspl@}%
```

> \Mgls[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

\Mgls

Use \Gls for first element and \gls for others.

```
                \glsxtr@newmgls{Mgls}{\@Gls@}{\@gls@}{\@Gls@}{\@gls@}%
                \glsmfuaddmap{\mgls}{\Mgls}
```

> \Mglspl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

\Mglspl

429

Use \Glspl for first element and \glspl for others.

```
\glsxtr@newmgls{Mglspl}{\@Glspl@}{\@glspl@}{\@Glspl@}{\@glspl@}%
\glsmfuaddmap{\mglspl}{\Mglspl}
```

\Mglsmainpl

> \Mglsmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Upper case the first element, no case change for others. Use plural for the main element only.

```
\glsxtr@newmgls{Mglsmainpl}{\@Gls@}{\@gls@}{\@Glspl@}{\@glspl@}%
\glsmfuaddmap{\mglsmainpl}{\Mglsmainpl}
```

\MGls

> \MGls[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \Gls for all elements.

```
\glsxtr@newmgls{MGls}{\@Gls@}{\@Gls@}{\@Gls@}{\@Gls@}%
\glsmfublocker{\MGls}
```

\MGlspl

> \MGlspl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \Glspl for all elements.

```
\glsxtr@newmgls{MGlspl}{\@Glspl@}{\@Glspl@}{\@Glspl@}{\@Glspl@}%
\glsmfublocker{\MGlspl}
```

\MGlsmainpl

> \MGlsmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Start all elements with upper case. Only use plural for main element.

```
\glsxtr@newmgls{MGlsmainpl}{\@Gls@}{\@Gls@}{\@Glspl@}{\@Glspl@}%
\glsmfublocker{\MGlsmainpl}
```

\MGLS

> \MGLS[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \GLS for all elements.

```
\glsxtr@newmgls{MGLS}{\@GLS@}{\@GLS@}{\@GLS@}{\@GLS@}%
\glsmfublocker{\MGLS}
```

\MGLSpl

> \MGLSpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \GLSpl for all elements.

```
\glsxtr@newmgls{MGLSpl}{\@GLSpl@}{\@GLSpl@}{\@GLSpl@}{\@GLSpl@}%
\glsmfublocker{\MGLSpl}
```

\MGLSmainpl

> \MGLSmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Upper case all elements. Only use plural for main element.

```
\glsxtr@newmgls{MGLSmainpl}{\@GLS@}{\@GLS@}{\@GLSpl@}{\@GLSpl@}%
\glsmfublocker{\MGLSmainpl}
```

\@glslongortext@

```
\def\@glslongortext#1#2[#3]{%
  \ifglshaslong{#2}{\@glsxtrlong{#1}{#2}[#3]}{\@glstext@{#1}{#2}[#3]}%
}
```

\@glsshortortext@

```
\def\@glsshortortext#1#2[#3]{%
  \ifglshasshort{#2}{\@glsxtrshort{#1}{#2}[#3]}{\@glstext@{#1}{#2}[#3]}%
}
```

\@glsfullorfirst@

```
\def\@glsfullorfirst#1#2[#3]{%
  \ifglshasshort{#2}{\@glsxtr@full{#1}{#2}[#3]}{\@glsfirst@{#1}{#2}[#3]}%
}
```

\@Glslongortext@

```
\def\@Glslongortext#1#2[#3]{%
  \ifglshaslong{#2}{\@Glsxtrlong{#1}{#2}[#3]}{\@Glstext@{#1}{#2}[#3]}%
}
```

\@Glsshortortext@

```
\def\@Glsshortortext#1#2[#3]{%
  \ifglshasshort{#2}{\@Glsxtrshort{#1}{#2}[#3]}{\@Glstext@{#1}{#2}[#3]}%
}
```

\@Glsfullorfirst@

```
\def\@Glsfullorfirst#1#2[#3]{%
  \ifglshasshort{#2}{\@Glsxtr@full{#1}{#2}[#3]}{\@Glsfirst@{#1}{#2}[#3]}%
}
```

\mglsshort

> \mglsshort[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use short or text for all elements.

```
\glsxtr@newmgls[short]{mglsshort}%
{\@glsshortortext}{\@glsshortortext}{\@glsshortortext}{\@glsshortortext}%
```

\mglslong

> \mglslong[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use long or text for all elements.

```
\glsxtr@newmgls[long]{mglslong}%
{\@glslongortext}{\@glslongortext}{\@glslongortext}{\@glslongortext}%
```

\mglsfull

$\boxed{\texttt{\textbackslash mglsfull[}\langle options\rangle\texttt{]\{}\langle label\rangle\texttt{\}[}\langle insert\rangle\texttt{]}}$

Use full or first for all elements.

```
\glsxtr@newmgls[first]{mglsfull}%
{\@glsfullorfirst}{\@glsfullorfirst}{\@glsfullorfirst}{\@glsfullorfirst}%
```

\Mglsshort

$\boxed{\texttt{\textbackslash Mglsshort[}\langle options\rangle\texttt{]\{}\langle label\rangle\texttt{\}[}\langle insert\rangle\texttt{]}}$

Use short or text for all elements with initial cap on first element.

```
\glsxtr@newmgls[short]{Mglsshort}%
{\@Glsshortortext}{\@glsshortortext}{\@Glsshortortext}{\@glsshortortext}%
\glsmfuaddmap{\mglsshort}{\Mglsshort}
```

\Mglslong

$\boxed{\texttt{\textbackslash Mglslong[}\langle options\rangle\texttt{]\{}\langle label\rangle\texttt{\}[}\langle insert\rangle\texttt{]}}$

Use long or text for all elements with initial cap on first element.

```
\glsxtr@newmgls[long]{Mglslong}%
{\@Glslongortext}{\@glslongortext}{\@Glslongortext}{\@glslongortext}%
\glsmfuaddmap{\mglslong}{\Mglslong}
```

\Mglsfull

$\boxed{\texttt{\textbackslash Mglsfull[}\langle options\rangle\texttt{]\{}\langle label\rangle\texttt{\}[}\langle insert\rangle\texttt{]}}$

Use full or first for all elements with initial cap on first element.

```
\glsxtr@newmgls[first]{Mglsfull}%
{\@Glsfullorfirst}{\@glsfullorfirst}{\@Glsfullorfirst}{\@glsfullorfirst}%
\glsmfuaddmap{\mglsfull}{\Mglsfull}
```

\mglsname

$\boxed{\texttt{\textbackslash mglsname[}\langle options\rangle\texttt{]\{}\langle label\rangle\texttt{\}[}\langle insert\rangle\texttt{]}}$

Use name for all elements.

```
\glsxtr@newmgls[name]{mglsname}%
{\@glsname@}{\@glsname@}{\@glsname@}{\@glsname@}%
```

\Mglsname

$\boxed{\texttt{\textbackslash Mglsname[}\langle options\rangle\texttt{]\{}\langle label\rangle\texttt{\}[}\langle insert\rangle\texttt{]}}$

Use name for all elements with initial cap on first element.

```
\glsxtr@newmgls[name]{Mglsname}%
 {\@Glsname@}{\@glsname@}{\@Glsname@}{\@glsname@}%
\glsmfuaddmap{\mglsname}{\Mglsname}
```

\MGlsname

> **\MGlsname[⟨options⟩]{⟨label⟩}[⟨insert⟩]**

Use name for all elements with initial cap on all elements.

```
\glsxtr@newmgls[name]{MGlsname}%
 {\@Glsname@}{\@Glsname@}{\@Glsname@}{\@Glsname@}%
\glsmfublocker{\MGlsname}
```

\@glssymbolorgls

```
\def\@glssymbolorgls#1#2[#3]{%
  \ifglshassymbol{#2}{\@glssymbol@{#1}{#2}[#3]}{\@gls@{#1}{#2}[#3]}%
}
```

\@glssymbolorGls

```
\def\@glssymbolorGls#1#2[#3]{%
  \ifglshassymbol{#2}{\@glssymbol@{#1}{#2}[#3]}{\@Gls@{#1}{#2}[#3]}%
}
```

\mglssymbol

> **\mglssymbol[⟨options⟩]{⟨label⟩}[⟨insert⟩]**

Use \glssymbol if the symbol key is set otherwise use \gls.

```
\glsxtr@newmgls[symbol]{mglssymbol}%
 {\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
```

\Mglssymbol

> **\Mglssymbol[⟨options⟩]{⟨label⟩}[⟨insert⟩]**

As above but initial the first element if it's not a symbol.

```
\glsxtr@newmgls[symbol]{Mglssymbol}%
 {\@glssymbolorGls}{\@glssymbolorgls}{\@glssymbolorGls}{\@glssymbolorgls}%
\glsmfuaddmap{\mglssymbol}{\Mglssymbol}
```

\MGlssymbol

> **\MGlssymbol[⟨options⟩]{⟨label⟩}[⟨insert⟩]**

As above but initial each element if it's not a symbol.

```
\glsxtr@newmgls[symbol]{MGlssymbol}%
 {\@glssymbolorGls}{\@glssymbolorGls}{\@glssymbolorGls}{\@glssymbolorGls}%
\glsmfublocker{\MGlssymbol}
```

433

`\mglsfield`

```
\newcommand{\mglsfield}{useri}
```

`\@glsfieldorgls`

```
\def\@glsfieldorgls#1#2[#3]{%
  \glsxtrifhasfield{\mglsfield}{#2}%
  {\@glsdisp[#1]{#2}{\glscurrentfieldvalue#3}}%
  {\@gls@{#1}{#2}[#3]}%
}
```

`\@Glsfieldorgls`

```
\def\@Glsfieldorgls#1#2[#3]{%
  \glsxtrifhasfield{\mglsfield}{#2}%
  {\@glsdisp[#1]{#2}{%
    \expandafter\glssentencecase\expandafter{\glscurrentfieldvalue#3}}}%
  {\@Gls@{#1}{#2}[#3]}%
}
```

`\mglsusefield[⟨options⟩]{⟨label⟩}[⟨insert⟩]`

`\mglsusefield`

Use the field given by `\mglsfield`.

```
\glsxtr@newmgls[\mglsfield]{mglsusefield}%
 {\@glsfieldorgls}{\@glsfieldorgls}{\@glsfieldorgls}{\@glsfieldorgls}%
```

`\Mglsusefield[⟨options⟩]{⟨label⟩}[⟨insert⟩]`

`\Mglsusefield`

As above but use initial cap for first element only.

```
\glsxtr@newmgls[\mglsfield]{Mglsusefield}%
 {\@Glsfieldorgls}{\@glsfieldorgls}{\@Glsfieldorgls}{\@glsfieldorgls}%
\glsmfuaddmap{\mglsusefield}{\Mglsusefield}
```

`\MGlsusefield[⟨options⟩]{⟨label⟩}[⟨insert⟩]`

`\MGlsusefield`

As above but use initial cap for all elements.

```
\glsxtr@newmgls[\mglsfield]{MGlsusefield}%
 {\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}%
\glsmfublocker{\MGlsusefield}
```

Use commands provided by glossaries-prefix if it has been loaded.

`\mpglsWarning`

```
\newcommand*{\mpglsWarning}{%
  \GlossariesExtraWarning{glossaries-prefix.sty is required for
   \string\mpgls\space family of commands}%
}
```

434

\@pglsorgls

```
\def\@pglsorgls#1#2[#3]{%
  \ifdef\@pgls@{\@pgls@{#1}{#2}[#3]}{\mpglsWarning\@gls@{#1}{#2}[#3]}%
}
```

\@pglsorglspl

```
\def\@pglsorglspl#1#2[#3]{%
  \ifdef\@pglspl@{\@pglspl@{#1}{#2}[#3]}{\mpglsWarning\@glspl@{#1}{#2}[#3]}%
}
```

\@Pglsorgls

```
\def\@Pglsorgls#1#2[#3]{%
  \ifdef\@Pgls@{\@Pgls@{#1}{#2}[#3]}{\mpglsWarning\@Gls@{#1}{#2}[#3]}%
}
```

\@pglsorglspl

```
\def\@pglsorglspl#1#2[#3]{%
  \ifdef\@pglspl@{\@pglspl@{#1}{#2}[#3]}{\mpglsWarning\@glspl@{#1}{#2}[#3]}%
}
```

\@Pglsorglspl

```
\def\@Pglsorglspl#1#2[#3]{%
  \ifdef\@Pglspl@{\@Pglspl@{#1}{#2}[#3]}{\mpglsWarning\@Glspl@{#1}{#2}[#3]}%
}
```

\@PGLSorgls

```
\def\@PGLSorgls#1#2[#3]{%
  \ifdef\@PGLS@{\@PGLS@{#1}{#2}[#3]}{\mpglsWarning\@GLS@{#1}{#2}[#3]}%
}
```

\@PGLSorglspl

```
\def\@PGLSorglspl#1#2[#3]{%
  \ifdef\@PGLSpl@{\@PGLSpl@{#1}{#2}[#3]}{\mpglsWarning\@GLSpl@{#1}{#2}[#3]}%
}
```

\mpgls

> \mpgls[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \pgls for the first element and \gls for the remainder.

`\glsxtr@newmgls{mpgls}{\@pglsorgls@}{\@gls@}{\@pglsorgls@}{\@gls@}%`

\mpglspl

> \mpglspl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \pglspl for the first element and \glspl for the remainder.

`\glsxtr@newmgls{mpglspl}{\@pglsorglspl@}{\@glspl@}{\@pglsorglspl@}{\@glspl@}%`

435

\mpglsmainpl

> \mpglsmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Only use plural for main element and only use prefixing command for first element.

```
\glsxtr@newmgls{mpglsmainpl}{\@pglsorgls@}{\@gls@}{\@pglsorglspl@}{\@glspl@}%
```

\Mpgls

> \Mpgls[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \Pgls for the first element and \gls for the remainder.

```
\glsxtr@newmgls{Mpgls}{\@Pglsorgls@}{\@gls@}{\@Pglsorgls@}{\@gls@}%
\glsmfuaddmap{\mpgls}{\Mpgls}
```

\Mpglspl

> \Mpglspl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \Pglspl for the first element and \glspl for the remainder.

```
\glsxtr@newmgls{Mpglspl}{\@Pglsorglspl@}{\@glspl@}{\@Pglsorglspl@}{\@glspl@}%
\glsmfuaddmap{\mpglspl}{\Mpglspl}
```

\Mpglsmainpl

> \Mpglsmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Only use plural for main element and only use first letter uppercase prefixing command for first element.

```
\glsxtr@newmgls{Mpglsmainpl}{\@Pglsorgls@}{\@gls@}{\@Pglsorglspl@}{\@glspl@}%
\glsmfuaddmap{\mpglsmainpl}{\Mpglsmainpl}
```

\MPGls

> \MPGls[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \Pgls for the first element and \Gls for the remainder.

```
\glsxtr@newmgls{MPGls}{\@Pglsorgls@}{\@Gls@}{\@Pglsorgls@}{\@Gls@}%
\glsmfublocker{\MPGls}
```

\MPGlspl

> \MPGlspl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \Pglspl for the first element and \Glspl for the remainder.

```
\glsxtr@newmgls{MPGlspl}{\@Pglsorglspl@}{\@Glspl@}{\@Pglsorglspl@}{\@Glspl@}%
\glsmfublocker{\MPGlspl}
```

\MPGlsmainpl

> \MPGlsmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Only use plural for main element and first letter uppercase all elements.

```
\glsxtr@newmgls{MPGlsmainpl}{\@Pglsorgls@}{\@Gls@}{\@Pglsorglspl@}{\@Glspl@}%
\glsmfublocker{\MPGlsmainpl}
```

\MPGLS

> \MPGLS[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \PGLS for the first element and \GLS for the remainder.

```
\glsxtr@newmgls{MPGLS}{\@PGLSorgls@}{\@GLS@}{\@PGLSorgls@}{\@GLS@}%
\glsmfublocker{\MPGLS}
```

\MPGLSpl

> \MPGLSpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Use \PGLSpl for the first element and \GLSpl for the remainder.

```
\glsxtr@newmgls{MPGLSpl}{\@PGLSorglspl@}{\@GLSpl@}{\@PGLSorglspl@}{\@GLSpl@}%
\glsmfublocker{\MPGLSpl}
```

\MPGLSmainpl

> \MPGLSmainpl[⟨*options*⟩]{⟨*label*⟩}[⟨*insert*⟩]

Only use plural for main element and uppercase all elements.

```
\glsxtr@newmgls{MPGLSmainpl}{\@PGLSorgls@}{\@GLS@}{\@PGLSorglspl@}{\@GLSpl@}%
\glsmfublocker{\MPGLSmainpl}
```

Not currently implementing any other variations.

1.11 Multi-Lingual Support

Add the facility to load language modules, if they are installed, but none are provided with this package.

\glsxtrcontinuedname Provide for use in \printunsrttable.

```
\providecommand{\glsxtrcontinuedname}{continued}
```

\RequireGlossariesExtraLang

```
\newcommand*{\RequireGlossariesExtraLang}[1]{%
  \@ifundefined{ver@glossariesxtr-#1.ldf}{\input{glossariesxtr-#1.ldf}}{}%
}
```

\ProvidesGlossariesExtraLang

```
\newcommand*{\ProvidesGlossariesExtraLang}[1]{%
  \ProvidesFile{glossariesxtr-#1.ldf}%
}
```

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. However, with bib2gls it might be useful to provide custom rules for a particular locale.)

\glsxtr@loaddialect The dialect label should be stored in \this@dialect before using this command.

```
\newcommand{\glsxtr@loaddialect}{%
  \IfTrackedLanguageFileExists{\this@dialect}%
  {glossariesxtr-}% prefix
  {.ldf}%
  {%
    \RequireGlossariesExtraLang{\CurrentTrackedTag}%
  }%
  {}% not found
```

If glossaries-extra-bib2gls has been loaded, \@glsxtrdialecthook will check for the associated script, otherwise it will do nothing.

```
  \@glsxtrdialecthook
}
```

```
\@ifpackageloaded{tracklang} {%
  \AnyTrackedLanguages
  {%
    \ForEachTrackedDialect{\this@dialect}{\glsxtr@loaddialect}%
  }%
  {}%
} {}
```

The style needs to be set at the end to ensure that \setglossarystyle has been redefined and extra style commands have been defined. Load glossaries-extra-stylemods if required.

```
\@glsxtr@redefstyles
```

and set the style:

```
\@glsxtr@do@style
```

2   Predefined Abbreviation Styles (glossaries-extra-abbrstyles.def)

```
\ProvidesFile{glossaries-extra-abbrstyles.def}[2025/05/14 v1.7 (NLCT)]
```

This file contains the predefined abbreviation styles. Some helper commands first.

\glsxtrlongshortname

```
\newcommand*{\glsxtrlongshortname}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
}
```

Provide convenient wrappers for common formats.

\glsxtrlongformat

> \glsxtrlongformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}

```
\newcommand*{\glsxtrlongformat}[3]{%
```
Don't add inner formatting if markwords attribute set as the inner formatting
is implemented within \glsxtrword and \glsxtrwordsep.
```
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccesslong{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\glsaccesslong{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccessfmtlong{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsxtrgenentrytextfmt{#2}%
    \fi
  }%
}%
```

\glsxtrlongplformat

> \glsxtrlongplformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}

```
\newcommand*{\glsxtrlongplformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccesslongpl{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\glsaccesslongpl{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccessfmtlongpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsxtrgenentrytextfmt{#2}%
    \fi
  }%
```

```
}%
```

$\boxed{\text{\Glsxtrlongformat}\{\langle label\rangle\}\{\langle insert\rangle\}\{\langle longfmtcs\rangle\}}$

```
\newcommand*{\Glsxtrlongformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccesslong{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\Glsaccesslong{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccessfmtlong{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsxtrgenentrytextfmt{#2}%
    \fi
  }%
}%
```

$\boxed{\text{\Glsxtrlongplformat}\{\langle label\rangle\}\{\langle insert\rangle\}\{\langle longfmtcs\rangle\}}$

```
\newcommand*{\Glsxtrlongplformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccesslongpl{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\Glsaccesslongpl{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccessfmtlongpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsxtrgenentrytextfmt{#2}%
    \fi
  }%
}%
```

\GLSxtrlongformat

```
\newcommand*{\GLSxtrlongformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccesslong{#1}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\GLSaccesslong{#1}}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccessfmtlong{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
      \mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\GLSxtrlongplformat

```
\newcommand*{\GLSxtrlongplformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccesslongpl{#1}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\GLSaccesslongpl{#1}}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccessfmtlongpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
      \mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\glsxtrlongformatgrp

Add grouping around insert.

```
\newcommand*{\glsxtrlongformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccesslong{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\glsaccesslong{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {#3{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\glsxtrlongformatplgrp{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}

\glsxtrlongplformatgrp

Add grouping around insert.

```
\newcommand*{\glsxtrlongplformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccesslongpl{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\glsaccesslongpl{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {#3{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\Glsxtrlongformatgrp{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}

\Glsxtrlongformatgrp

Add grouping around insert.

```
\newcommand*{\Glsxtrlongformatgrp}[3]{%
```

```
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccesslong{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\Glsaccesslong{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {#3{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\Glsxtrlongplformatgrp

> **\Glsxtrlongformatplgrp{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}**

Add grouping around insert.

```
\newcommand*{\Glsxtrlongplformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccesslongpl{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\Glsaccesslongpl{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {#3{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\GLSxtrlongformatgrp

> **\GLSxtrlongformatgrp{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}**

Add grouping around insert.

```
\newcommand*{\GLSxtrlongformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
```

443

```
    \ifglsxtrinsertinside
      #3{\GLSaccesslong{#1}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      #3{\GLSaccesslong{#1}}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    #3{\GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {\mfirstucMakeUppercase{#3{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \fi
  }%
}%
```

> $\backslash$**GLSxtrlongformatplgrp**{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}

Add grouping around insert.

```
\newcommand*{\GLSxtrlongplformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccesslongpl{#1}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      #3{\GLSaccesslongpl{#1}}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    #3{\GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {\mfirstucMakeUppercase{#3{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \fi
  }%
}%
```

> $\backslash$**glsxtrshortformat**{⟨*label*⟩}{⟨*insert*⟩}{⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtrshortformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccessshort{#1}\glsxtrgenentrytextfmt{#2}}%
```

444

```
      \else
        #3{\glsaccessshort{#1}}\glsxtrgenentrytextfmt{#2}%
      \fi
    }%
    {%
      \ifglsxtrinsertinside
        #3{\glsaccessfmtshort{#2}{\glsxtrgenentrytextfmt}{#1}}%
      \else
        #3{\glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
        \glsxtrgenentrytextfmt{#2}%
      \fi
    }%
  }%
```

> \glsxtrshortplformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtrshortplformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccessshortpl{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\glsaccessshortpl{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccessfmtshortpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsxtrgenentrytextfmt{#2}%
    \fi
  }%
}%
```

> \Glsxtrshortformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*shortfmtcs*⟩}

```
\newcommand*{\Glsxtrshortformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccessshort{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\Glsaccessshort{#1}}\glsxtrgenentrytextfmt{#2}%
```

```
        \fi
    }%
    {%
      \ifglsxtrinsertinside
        #3{\Glsaccessfmtshort{#2}{\glsxtrgenentrytextfmt}{#1}}%
      \else
        #3{\Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
        \glsxtrgenentrytextfmt{#2}%
      \fi
    }%
}%
```

```
\newcommand*{\Glsxtrshortplformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccessshortpl{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\Glsaccessshortpl{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccessfmtshortpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsxtrgenentrytextfmt{#2}%
    \fi
  }%
}%
```

```
\newcommand*{\GLSxtrshortformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccessshort{#1}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\GLSaccessshort{#1}}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
```

```
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccessfmtshort{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
      \mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

```
\newcommand*{\GLSxtrshortplformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccessshortpl{#1}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\GLSaccessshortpl{#1}}\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccessfmtshortpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
      \mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

Add grouping around insert.

```
\newcommand*{\glsxtrshortformatgrp}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccessshort{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\glsaccessshort{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
```

```
      \ifglsxtrinsertinside
        {#3{\glsxtrgenentrytextfmt{#2}}}%
      \else
        {\glsxtrgenentrytextfmt{#2}}%
      \fi
    }%
  }%
```

$\backslash\mathtt{glsxtrshortplformatgrp}\{\langle label\rangle\}\{\langle insert\rangle\}\{\langle shortfmtcs\rangle\}$

Add grouping around insert.

```
\newcommand*{\glsxtrshortplformatgrp}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\glsaccessshortpl{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\glsaccessshortpl{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {#3{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

$\backslash\mathtt{Glsxtrshortformatgrp}\{\langle label\rangle\}\{\langle insert\rangle\}\{\langle shortfmtcs\rangle\}$

Add grouping around insert.

```
\newcommand*{\Glsxtrshortformatgrp}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccessshort{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\Glsaccessshort{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {#3{\glsxtrgenentrytextfmt{#2}}}%
```

```
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

> **\Glsxtrshortplformatgrp**{⟨*label*⟩}{⟨*insert*⟩}{⟨*shortfmtcs*⟩}

Add grouping around insert.

```
\newcommand*{\Glsxtrshortplformatgrp}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\Glsaccessshortpl{#1}{\glsxtrgenentrytextfmt{#2}}}%
    \else
      #3{\Glsaccessshortpl{#1}}{\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    #3{\Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {#3{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

> **\GLSxtrshortformatgrp**{⟨*label*⟩}{⟨*insert*⟩}{⟨*shortfmtcs*⟩}

Add grouping around insert.

```
\newcommand*{\GLSxtrshortformatgrp}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglsxtrinsertinside
      #3{\GLSaccessshort{#1}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      #3{\GLSaccessshort{#1}}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    #3{\GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{#1}}%
    \ifglsxtrinsertinside
      {\mfirstucMakeUppercase{#3{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
```

```
        \fi
    }%
}%
```

┌───┐
│ \GLSxtrshortplformatgrp{⟨label⟩}{⟨insert⟩}{⟨shortfmtcs⟩}   │
└───┘

\GLSxtrshortplformatgrp

Add grouping around insert.

```
\newcommand*{\GLSxtrshortplformatgrp}[3]{%
 \glsifattribute{#1}{markshortwords}{true}%
 {%
   \ifglsxtrinsertinside
     #3{\GLSaccessshortpl{#1}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
   \else
     #3{\GLSaccessshortpl{#1}}{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
   \fi
 }%
 {%
   #3{\GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{#1}}%
   \ifglsxtrinsertinside
     {\mfirstucMakeUppercase{#3{\glsxtrgenentrytextfmt{#2}}}}%
   \else
     {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
   \fi
 }%
}%
```

┌───┐
│ \glsxtrlongshortformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}   │
│ {⟨shortfmtcs⟩}   │
└───┘

\glsxtrlongshortformat

```
\newcommand*{\glsxtrlongshortformat}[4]{%
 \glsxtrlongformat{#1}{#2}{#3}%
 \glsxtrfullsep{#1}%
 \glsxtrparen{\glsxtrshortformat{#1}{}{#4}}%
}%
```

┌───┐
│ \glsxtrlongshortplformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}   │
│ {⟨shortfmtcs⟩}   │
└───┘

\glsxtrlongshortplformat

```
\newcommand*{\glsxtrlongshortplformat}[4]{%
 \glsxtrlongplformat{#1}{#2}{#3}%
 \glsxtrfullsep{#1}%
 \glsxtrparen{\glsxtrshortplformat{#1}{}{#4}}%
}%
```

450

\Glsxtrlongshortformat

> \Glsxtrlongshortformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
> {⟨*shortfmtcs*⟩}

```
\newcommand*{\Glsxtrlongshortformat}[4]{%
  \Glsxtrlongformat{#1}{#2}{#3}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrshortformat{#1}{}{#4}}%
}%
```

\Glsxtrlongshortplformat

> \Glsxtrlongshortplformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
> {⟨*shortfmtcs*⟩}

```
\newcommand*{\Glsxtrlongshortplformat}[4]{%
  \Glsxtrlongplformat{#1}{#2}{#3}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrshortplformat{#1}{}{#4}}%
}%
```

\GLSxtrlongshortformat

> \GLSxtrlongshortformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
> {⟨*shortfmtcs*⟩}

```
\newcommand*{\GLSxtrlongshortformat}[4]{%
  \GLSxtrlongformat{#1}{#2}{#3}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\GLSxtrshortformat{#1}{}{#4}}%
}%
```

\GLSxtrlongshortplformat

> \GLSxtrlongshortplformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
> {⟨*shortfmtcs*⟩}

```
\newcommand*{\GLSxtrlongshortplformat}[4]{%
  \GLSxtrlongplformat{#1}{#2}{#3}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\GLSxtrshortplformat{#1}{}{#4}}%
}%
```

\glsxtrshortlongformat

> \glsxtrshortlongformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
> {⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtrshortlongformat}[4]{%
  \glsxtrshortformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrlongformat{#1}{}{#3}}%
}%
```

\glsxtrshortlongplformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
{⟨shortfmtcs⟩}

\glsxtrshortlongplformat

```
\newcommand*{\glsxtrshortlongplformat}[4]{%
  \glsxtrshortplformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrlongplformat{#1}{}{#3}}%
}%
```

\Glsxtrshortlongformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
{⟨shortfmtcs⟩}

\Glsxtrshortlongformat

```
\newcommand*{\Glsxtrshortlongformat}[4]{%
  \Glsxtrshortformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrlongformat{#1}{}{#3}}%
}%
```

\Glsxtrshortlongplformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
{⟨shortfmtcs⟩}

\Glsxtrshortlongplformat

```
\newcommand*{\Glsxtrshortlongplformat}[4]{%
  \Glsxtrshortplformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrlongplformat{#1}{}{#3}}%
}%
```

\GLSxtrshortlongformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
{⟨shortfmtcs⟩}

\GLSxtrshortlongformat

```
\newcommand*{\GLSxtrshortlongformat}[4]{%
```

452

```
\GLSxtrshortformat{#1}{#2}{#4}%
\glsxtrfullsep{#1}%
\glsxtrparen{\GLSxtrlongformat{#1}{}{#3}}%
}%
```

\GLSxtrshortlongplformat

$\boxed{\begin{array}{l}\texttt{\textbackslash GLSxtrshortlongplformat\{}\langle\mathit{label}\rangle\texttt{\}\{}\langle\mathit{insert}\rangle\texttt{\}\{}\langle\mathit{longfmtcs}\rangle\texttt{\}}\\\texttt{\{}\langle\mathit{shortfmtcs}\rangle\texttt{\}}\end{array}}$

```
\newcommand*{\GLSxtrshortlongplformat}[4]{%
\GLSxtrshortplformat{#1}{#2}{#4}%
\glsxtrfullsep{#1}%
\glsxtrparen{\GLSxtrlongplformat{#1}{}{#3}}%
}%
```

\glsxtrfootnotelongformat

$\boxed{\texttt{\textbackslash glsxtrfootnotelongformat\{}\langle\mathit{label}\rangle\texttt{\}\{}\langle\mathit{longfmtcs}\rangle\texttt{\}}}$

```
\newcommand*{\glsxtrfootnotelongformat}[2]{%
\glsxtrlongformat{#1}{}{#2}%
}%
```

\glsxtrfootnotelongplformat

$\boxed{\texttt{\textbackslash glsxtrfootnotelongplformat\{}\langle\mathit{label}\rangle\texttt{\}\{}\langle\mathit{longfmtcs}\rangle\texttt{\}}}$

```
\newcommand*{\glsxtrfootnotelongplformat}[2]{%
\glsxtrlongplformat{#1}{}{#2}%
}%
```

\glsxtrpostfootnotelongformat

$\boxed{\texttt{\textbackslash glsxtrpostfootnotelongformat\{}\langle\mathit{label}\rangle\texttt{\}\{}\langle\mathit{longfmtcs}\rangle\texttt{\}}}$

```
\newcommand*{\glsxtrpostfootnotelongformat}{%
\glsxtrfootnotelongformat
}%
```

\glsxtrpostusershortformat

$\boxed{\texttt{\textbackslash glsxtruserpostshortformat\{}\langle\mathit{label}\rangle\texttt{\}\{}\langle\mathit{shortfmtcs}\rangle\texttt{\}}}$

```
\newcommand*{\glsxtrpostusershortformat}[2]{%
\glsxtrifallcaps
```

```
  {\GLSxtrusershortformat{#1}{#2}}%
  {\glsxtrusershortformat{#1}{#2}}%
}%
```

\glsxtrusershortformat

> \glsxtrusershortformat{⟨*label*⟩}{⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtrusershortformat}[2]{%
  \glsxtruserparen{\glsxtrshortformat{#1}{}{#2}}{#1}%
}%
```

\glsxtrusershortplformat

> \glsxtrusershortplformat{⟨*label*⟩}{⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtrusershortplformat}[2]{%
  \glsxtruserparen{\glsxtrshortplformat{#1}{}{#2}}{#1}%
}%
```

\GLSxtrusershortformat

> \GLSxtrusershortformat{⟨*label*⟩}{⟨*shortfmtcs*⟩}

```
\newcommand*{\GLSxtrusershortformat}[2]{%
  \GLSxtruserparen{\GLSxtrshortformat{#1}{}{#2}}{#1}%
}%
```

\GLSxtrusershortplformat

> \GLSxtrusershortplformat{⟨*label*⟩}{⟨*shortfmtcs*⟩}

```
\newcommand*{\GLSxtrusershortplformat}[2]{%
  \GLSxtruserparen{\GLSxtrshortplformat{#1}{}{#2}}{#1}%
}%
```

\glsxtrpostuserlongformat

> \glsxtruserpostlongformat{⟨*label*⟩}{⟨*longfmtcs*⟩}

```
\newcommand*{\glsxtrpostuserlongformat}[2]{%
  \glsxtrifallcaps
  {\GLSxtruserlongformat{#1}{#2}}%
  {\glsxtruserlongformat{#1}{#2}}%
}%
```

\glsxtruserlongformat

> \glsxtruserlongformat{⟨*label*⟩}{⟨*longfmtcs*⟩}

```
\newcommand*{\glsxtruserlongformat}[2]{%
  \glsxtruserparen{\glsxtrlongformat{#1}{}{#2}}{#1}%
}%
```

\GLSxtruserlongformat

> \GLSxtruserlongformat{⟨*label*⟩}{⟨*longfmtcs*⟩}

```
\newcommand*{\GLSxtruserlongformat}[2]{%
  \GLSxtruserparen{\GLSxtrlongformat{#1}{}{#2}}{#1}%
}%
```

\glsxtruserlongplformat

> \glsxtruserlongplformat{⟨*label*⟩}{⟨*longfmtcs*⟩}

```
\newcommand*{\glsxtruserlongplformat}[2]{%
  \glsxtruserparen{\glsxtrlongplformat{#1}{}{#2}}{#1}%
}%
```

\GLSxtruserlongplformat

> \GLSxtruserlongplformat{⟨*label*⟩}{⟨*longfmtcs*⟩}

```
\newcommand*{\GLSxtruserlongplformat}[2]{%
  \GLSxtruserparen{\GLSxtrlongplformat{#1}{}{#2}}{#1}%
}%
```

\glsxtruserlongshortformat

> \glsxtruserlongshortformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
> {⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtruserlongshortformat}[4]{%
  \glsxtrlongformat{#1}{#2}{#3}%
  \glsxtrusershortformat{#1}{#4}%
}%
```

\glsxtruserlongshortplformat

> \glsxtruserlongshortplformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
> {⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtruserlongshortplformat}[4]{%
  \glsxtrlongplformat{#1}{#2}{#3}%
  \glsxtrusershortplformat{#1}{#4}%
}%
```

\Glsxtruserlongshortformat

> \Glsxtruserlongshortformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
> {⟨shortfmtcs⟩}

```
\newcommand*{\Glsxtruserlongshortformat}[4]{%
  \Glsxtrlongformat{#1}{#2}{#3}%
  \glsxtrusershortformat{#1}{#4}%
}%
```

\Glsxtruserlongshortplformat

> \Glsxtruserlongshortplformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
> {⟨shortfmtcs⟩}

```
\newcommand*{\Glsxtruserlongshortplformat}[4]{%
  \Glsxtrlongplformat{#1}{#2}{#3}%
  \glsxtrusershortplformat{#1}{#4}%
}%
```

\GLSxtruserlongshortformat

> \GLSxtruserlongshortformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
> {⟨shortfmtcs⟩}

```
\newcommand*{\GLSxtruserlongshortformat}[4]{%
  \GLSxtrlongformat{#1}{#2}{#3}%
  \GLSxtrusershortformat{#1}{#4}%
}%
```

\GLSxtruserlongshortplformat

> \GLSxtruserlongshortplformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}
> {⟨shortfmtcs⟩}

```
\newcommand*{\GLSxtruserlongshortplformat}[4]{%
  \GLSxtrlongplformat{#1}{#2}{#3}%
  \GLSxtrusershortplformat{#1}{#4}%
}%
```

\glsxtrusershortlongformat

\glsxtrusershortlongformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
{⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtrusershortlongformat}[4]{%
  \glsxtrshortformat{#1}{#2}{#3}%
  \glsxtruserlongformat{#1}{#4}%
}%
```

\glsxtrusershortlongplformat

\glsxtrusershortlongplformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
{⟨*shortfmtcs*⟩}

```
\newcommand*{\glsxtrusershortlongplformat}[4]{%
  \glsxtrshortplformat{#1}{#2}{#3}%
  \glsxtruserlongplformat{#1}{#4}%
}%
```

\Glsxtrusershortlongformat

\Glsxtrusershortlongformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
{⟨*shortfmtcs*⟩}

```
\newcommand*{\Glsxtrusershortlongformat}[4]{%
  \Glsxtrshortformat{#1}{#2}{#3}%
  \glsxtruserlongformat{#1}{#4}%
}%
```

\Glsxtrusershortlongplformat

\Glsxtrusershortlongplformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
{⟨*shortfmtcs*⟩}

```
\newcommand*{\Glsxtrusershortlongplformat}[4]{%
  \Glsxtrshortplformat{#1}{#2}{#3}%
  \glsxtruserlongplformat{#1}{#4}%
}%
```

\GLSxtrusershortlongformat

\GLSxtrusershortlongformat{⟨*label*⟩}{⟨*insert*⟩}{⟨*longfmtcs*⟩}
{⟨*shortfmtcs*⟩}

```
\newcommand*{\GLSxtrusershortlongformat}[4]{%
```

```
\GLSxtrshortformat{#1}{#2}{#3}%
\GLSxtruserlongformat{#1}{#4}%
}%
```

> `\GLSxtrusershortlongplformat{⟨label⟩}{⟨insert⟩}{⟨longfmtcs⟩}`
> `{⟨shortfmtcs⟩}`

```
\newcommand*{\GLSxtrusershortlongplformat}[4]{%
  \GLSxtrshortplformat{#1}{#2}{#3}%
  \GLSxtruserlongplformat{#1}{#4}%
}%
```

2.1 Predefined Styles (Default Font)

long-short

```
\newabbreviationstyle{long-short}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortname},
  sort={\the\glsshorttok},
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%

  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  description={\the\glslongtok}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

458

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glsxtrrevert}[1]{\glsxtrdefaultrevert{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
}
```

Set this as the default style for general abbreviations:

```
\setabbreviationstyle{long-short}
```

\glsxtrlongshortdescsort

```
\newcommand*{\glsxtrlongshortdescsort}{%
 \expandonce\glsxtrorglong\space (\expandonce\glsxtrorgshort)%
}
```

\glsxtrlongshortdescname

```
\newcommand*{\glsxtrlongshortdescname}{%
  \glsxplongfont{\the\glslongtok}{\glscategorylabel}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}%
}
```

459

long-short-desc User supplies description. The long form is included in the name.

```
\newabbreviationstyle{long-short-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortdescname},
  sort={\glsxtrlongshortdescsort},%
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
```

The text key should only have the short form.

```
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%

  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-short}%
}
```

\glsxtrshortlongname

```
\newcommand*{\glsxtrshortlongname}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
}
```

short-long Short form followed by long form in parenthesis on first use.

```
\newabbreviationstyle{short-long}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortlongname},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%

  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
 }%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
```

```
        \GLSxtrshortlongformat{##1}{##2}%
          {\glsfirstlongfont}{\glsfirstabbrvfont}%
      }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
          {\glsfirstlongfont}{\glsfirstabbrvfont}%
      }%
  }
```

\glsxtrshortlongdescsort

```
\newcommand*{\glsxtrshortlongdescsort}{\expandonce\glsxtrorgshort}
```

\glsxtrshortlongdescname

```
\newcommand*{\glsxtrshortlongdescname}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\glsxplongfont{\the\glslongtok}{\glscategorylabel}}}%
}
```

short-long-desc User supplies description. The long form is included in the name.

```
\newabbreviationstyle{short-long-desc}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},
    sort={\glsxtrshortlongdescsort},
    first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}}},%
    firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}}},%

    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%

    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
  }%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
```

```
      }%
      {%
        \GlsXtrUseAbbrStyleFmts{short-long}%
      }
```

\glsfirstlongfootnotefont Only used by the "footnote" styles.

```
\newcommand*{\glsfirstlongfootnotefont}[1]{\glslongfootnotefont{#1}}%
```

\glslongfootnotefont Only used by the "footnote" styles.

```
\newcommand*{\glslongfootnotefont}[1]{\glslongdefaultfont{#1}}%
```

\glsxtrabbrvfootnote

> $\boxed{\text{\textbackslash glsxtrabbrvfootnote\{}\langle label\rangle\text{\}\{}\langle long\rangle\text{\}}}$

Command used by footnote abbreviation styles. The default definition ignores the first argument. The second argument $\langle long\rangle$ 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).

```
\newcommand*{\glsxtrabbrvfootnote}[2]{\footnote{#2}}
```

\glsxtrpostabbrvfootnote Used by post-footnote style to include formatting.

```
\newrobustcmd*{\glsxtrpostabbrvfootnote}[2]{%
  \glsxtrabbrvfootnote{#1}%
  {#2\glsxtrpostfootnotelongformat{#1}{\glsfirstlongfootnotefont}}%
}
```

\xpglsxtrpostabbrvfootnote Perform all the appropriate expansions to ensure \glslabel and \glsxtrassignlinktextfmt are expanded as they may be lost by the time the footnote occurs.

```
\newcommand*{\xpglsxtrpostabbrvfootnote}{%
  \expandafter\expandafter\expandafter
  \glsxtrpostabbrvfootnote
  \expandafter\expandafter\expandafter
    {\expandafter\glslabel\expandafter}\expandafter
    {\glsxtrassignlinktextfmt}%
}
```

\glsxtrfootnotename

```
\newcommand*{\glsxtrfootnotename}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
}
```

footnote Short form followed by long form in footnote on first use.

```
\newabbreviationstyle{footnote}%
{%
```

Set accessibility attributes if enabled. (Add firstshortaccess since long form is hidden in a footnote on first use.) The inner formatting isn't be applied to the footnote text because the innertextformat key value may have gone out of scope

by that the time the footnote text is processed. (Neither is the outer formatting applied.)

```
\glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%

  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%

  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
  }%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form followed by the long form as a footnote.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%
```

```
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}}%
}%
```
The first use full form and the inline full form use the short (long) style.
```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
}
```

short-footnote

```
\letabbreviationstyle{short-footnote}{footnote}
```

`\glsxtrfootnotedescname`

```
\newcommand*{\glsxtrfootnotedescname}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\glsxplongfont{\the\glslongtok}{\glscategorylabel}}}%
}
```

`\glsxtrfootnotedescsort`

```
\newcommand*{\glsxtrfootnotedescsort}{\the\glsshorttok}
```

short-footnote-desc Like short-footnote but with user supplied description.

```
\newabbreviationstyle{short-footnote-desc}%
{%
```

Set accessibility attributes if enabled

```
\glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotedescname},
  sort={\glsxtrfootnotedescsort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
     {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
     {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the
regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
 }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{footnote}%
}
```

footnote-desc Synonym.

```
\letabbreviationstyle{footnote-desc}{short-footnote-desc}
```

postfootnote Similar to footnote but the footnote is placed afterwards, outside the link. This
avoids nested links and can also move the footnote marker after any following

466

punctuation mark. Pre v1.07 included `\footnote` in the first keys, which was incorrect as it caused duplicate footnotes.

```
\newabbreviationstyle{postfootnote}%
{%
```

Set accessibility attributes if enabled. (Add firstshortaccess since long form is hidden in a footnote on first use.)

```
\glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%

  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

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.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
```

The footnote needs to be suppressed in the inline form. Previously this was done by redefining glsxtrsetupfulldefs but that interferes with other styles. Instead, this now uses `\glsxtrifwasglslike`

```
        \glsxtrifwasglslikeandfirstuse
        {%
```

Ensure `\glslabel` and `\glsxtrassignlinktextfmt` are expanded as they may be lost by the time the footnote occurs.

```
        \glsxtrdopostpunc{\expandafter\expandafter\expandafter
          \glsxtrpostabbrvfootnote
          \expandafter\expandafter\expandafter
            {\expandafter\glslabel\expandafter}\expandafter
            {\glsxtrassignlinktextfmt}}%
      }{}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
```

```
    \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form. The long form is deferred.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
    }%
```

The first use full form and the inline full form use the short (long) style.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
    }%
 }
```

short-postfootnote

```
\letabbreviationstyle{short-postfootnote}{postfootnote}
```

short-postfootnote-desc Like short-postfootnote but with user supplied description.

```
\newabbreviationstyle{short-postfootnote-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotedescname},
  sort={\glsxtrfootnotedescsort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%

  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

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. Previously this was done by
redefining glsxtrsetupfulldefs but that interferes with other styles. Instead, this
now uses \glsxtrifwasglslike

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasglslikeandfirstuse
    {%
```

Ensure \glslabel and \glsxtrassignlinktextfmt are expanded as they may
be lost by the time the footnote occurs.

```
      \glsxtrdopostpunc{\expandafter\expandafter\expandafter
        \glsxtrpostabbrvfootnote
        \expandafter\expandafter\expandafter
          {\expandafter\glslabel\expandafter}\expandafter
          {\glsxtrassignlinktextfmt}}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{postfootnote}%
}
```

469

postfootnote-desc

```
\letabbreviationstyle{postfootnote-desc}{short-postfootnote-desc}
```

\glsxtrshortnolongname

```
\newcommand*{\glsxtrshortnolongname}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
}
```

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.

```
\newabbreviationstyle{short}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortnolongname},
    sort={\the\glsshorttok},
    first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
    firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short form followed by the long form in paren-
theses.

```
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsxtrshortlongformat{##1}{##2}%
      {\glsfirstlongfont}{\glsfirstabbrvfont}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsxtrshortlongplformat{##1}{##2}%
      {\glsfirstlongfont}{\glsfirstabbrvfont}%
  }%
```

470

```
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
     {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
     {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
     {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
     {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
```

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.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
}
```

Set this as the default style for acronyms:

```
\setabbreviationstyle[acronym]{short}
```

short-nolong

```
\letabbreviationstyle{short-nolong}{short}
```

short-nolong-noreg Like short-nolong but doesn't set the regular attribute.

```
\newabbreviationstyle{short-nolong-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{short-nolong}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong}%
}
```

\glsxtrshortdescname

```
\newcommand*{\glsxtrshortdescname}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\glsxplongfont{\the\glslongtok}{\glscategorylabel}}}%
}
```

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.

```
\newabbreviationstyle{short-desc}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortdescname},
    sort={\the\glsshorttok},
    first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
    firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short format followed by the long form in parentheses.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
```

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.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
}
```

short-nolong-desc

```
\letabbreviationstyle{short-nolong-desc}{short-desc}
```

short-nolong-desc-noreg Like short-nolong-desc but doesn't set the regular attribute.

```
\newabbreviationstyle{short-nolong-desc-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{short-nolong-desc}%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong-desc}%
}
```

nolong-short Similar to short-nolong but the full form shows the long form followed by the short form in parentheses.

```
\newabbreviationstyle{nolong-short}%
{%
  \GlsXtrUseAbbrStyleSetup{short-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong}%
```

The inline full form displays the long form followed by the short form in parentheses.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
```

```
        \GLSxtrlongshortplformat{##1}{##2}%
          {\glsfirstlongfont}{\glsfirstabbrvfont}%
      }%
    }
```

nolong-short-noreg Like nolong-short but doesn't set the regular attribute.

```
\newabbreviationstyle{nolong-short-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{nolong-short}%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{nolong-short}%
}
```

\glsxtrlongnoshortdescname

```
\newcommand*{\glsxtrlongnoshortdescname}{%
  \glsxplongfont{\the\glslongtok}{\glscategorylabel}%
}
```

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. The accessibility attributes don't need setting here.

```
\newabbreviationstyle{long-desc}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortdescname},
    sort={\the\glslongtok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
    firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
    text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
    plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
```

```
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrlongformat{##1}{##2}{\glslongfont}%
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrlongplformat{##1}{##2}{\glslongfont}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glslongfont}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \Glsxtrlongplformat{##1}{##2}{\glslongfont}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \GLSxtrlongformat{##1}{##2}{\glslongfont}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
  \GLSxtrlongplformat{##1}{##2}{\glslongfont}%
}%
```

The inline full form displays the long format followed by the short form in parentheses.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
   {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlongshortplformat{##1}{##2}%
   {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
   {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
   {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
   {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlongshortplformat{##1}{##2}%
   {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
```

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.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlongformat{##1}{##2}{\glsfirstlongfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlongplformat{##1}{##2}{\glsfirstlongfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glsfirstlongfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrlongplformat{##1}{##2}{\glsfirstlongfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrlongformat{##1}{##2}{\glsfirstlongfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrlongplformat{##1}{##2}{\glsfirstlongfont}%
}%
}
```

long-noshort-desc Provide a synonym that matches similar styles.

```
\letabbreviationstyle{long-noshort-desc}{long-desc}
```

long-noshort-desc-noreg Like long-noshort-desc but doesn't set the regular attribute.

```
\newabbreviationstyle{long-noshort-desc-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-noshort-desc}%
}
```

\glsxtrlongnoshortname

```
\newcommand*{\glsxtrlongnoshortname}{%
  \glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
}
```

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.

```
\newabbreviationstyle{long}%
{%
```

Set accessibility attributes if enabled.

```
    \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
    firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
    text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
    plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-desc}%
}
```

long-noshort Provide a synonym that matches similar styles.

```
  \letabbreviationstyle{long-noshort}{long}
```

long-noshort-noreg Like long-noshort but doesn't set the regular attribute.

```
  \newabbreviationstyle{long-noshort-noreg}%
  {%
    \GlsXtrUseAbbrStyleSetup{long-noshort}%
```

Unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
    \GlsXtrUseAbbrStyleFmts{long-noshort}%
  }
```

2.2   Predefined Styles (Small Capitals)

These styles use `\textsc` for the short form.

478

\glsxtrscfont   Maintained for backward-compatibility.

```
\newcommand*{\glsxtrscfont}[1]{\textsc{#1}}
```

\glsabbrvscfont   Added for consistent naming.

```
\newcommand*{\glsabbrvscfont}{\glsxtrscfont}
```

\glsxtrfirstscfont   Maintained for backward-compatibility.

```
\newcommand*{\glsxtrfirstscfont}[1]{\glsabbrvscfont{#1}}
```

\glsfirstabbrvscfont   Added for consistent naming.

```
\newcommand*{\glsfirstabbrvscfont}{\glsxtrfirstscfont}
```

and for the default short form suffix:

\glsxtrscsuffix   `\protect` needs to come inside `\glsxtrscsuffix` to avoid interfering with all caps.

```
\newcommand*{\glsxtrscsuffix}{\protect\glstextup{\glsxtrabbrvpluralsuffix}}
```

\glsxtrscrevert   Cancel smallcaps.

```
\newcommand*{\glsxtrscrevert}[1]{\glstextup{#1}}%
```

v1.49: the following now use commands like `\glsfirstinnerfmtabbrvfont` instead of `\glsfirstabbrvscfont` etc. This makes it easier to apply the inner formatting. The scoping added in v1.48 with `\glslinkwrcontent` should prevent formatting leakage in the event of nested commands. The only problem will be if commands like `\glsentryfirst` are used, but those aren't designed for consistent formatting. It will also make it easier to locally redefine `\glsfirstinnerfmtabbrvfont` to strip the formatting if those commands are used (rather than having to define all the possible abbreviation style formatting commands). Since these new commands are robust they don't need protecting.

long-short-sc

```
\newabbreviationstyle{long-short-sc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortname},
  sort={\the\glsshorttok},
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
```

```
        plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
        description={\the\glslongtok}}%
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
  }%
  {%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
      \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
      \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
      \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
      \renewcommand*\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```

Use the default long fonts.

```
      \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
      \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
      \renewcommand*{\glsxtrfullformat}[2]{%
        \glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\glsxtrfullplformat}[2]{%
        \glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\Glsxtrfullformat}[2]{%
        \Glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\Glsxtrfullplformat}[2]{%
        \Glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\GLSxtrfullformat}[2]{%
        \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\GLSxtrfullplformat}[2]{%
        \GLSxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
      }%
  }
```

long-short-sc-desc

```
                \newabbreviationstyle{long-short-sc-desc}%
                 {%
            Set accessibility attributes if enabled.
                     \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
            Setup the default fields.
                     \renewcommand*{\CustomAbbreviationFields}{%
                       name={\glsxtrlongshortdescname},
                       sort={\glsxtrlongshortdescsort},%
                       first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
                        \protect\glsxtrfullsep{\the\glslabeltok}%
                        \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
                       firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
                        \protect\glsxtrfullsep{\the\glslabeltok}%
                        \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
                       text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
                       plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
                     }%
            Unset the regular attribute if it has been set.
                     \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                       \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
                       \glshasattribute{\the\glslabeltok}{regular}%
                       {%
                         \glssetattribute{\the\glslabeltok}{regular}{false}%
                       }%
                       {}%
                     }%
                 }%
                 {%
            As long-short-sc style:
                     \GlsXtrUseAbbrStyleFmts{long-short-sc}%
                 }

short-sc-long   Now the short (long) version
                 \newabbreviationstyle{short-sc-long}%
                 {%
            Set accessibility attributes if enabled.
                     \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
            Setup the default fields.
                     \renewcommand*{\CustomAbbreviationFields}{%
                       name={\glsxtrshortlongname},
                       sort={\the\glsshorttok},
                       description={\the\glslongtok},%
                       first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
                        \protect\glsxtrfullsep{\the\glslabeltok}%
                        \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
                       firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
```

```
    \protect\glsxtrfullsep{\the\glslabeltok}%
    \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrshortlongformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsxtrshortlongplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \Glsxtrshortlongformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \Glsxtrshortlongplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
  }%
  \renewcommand*{\GLSxtrfullformat}[2]{%
    \GLSxtrshortlongformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
  }%
  \renewcommand*{\GLSxtrfullplformat}[2]{%
    \GLSxtrshortlongplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
  }%
}
```

short-sc-long-desc As before but user provides description

```
\newabbreviationstyle{short-sc-long-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortlongdescname},
  sort={\glsxtrshortlongdescsort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

As short-sc-long style:

```
\GlsXtrUseAbbrStyleFmts{short-sc-long}%
}
```

short-sc

```
\newabbreviationstyle{short-sc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortnolongname},
  sort={\the\glsshorttok},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
```

```
        description={\the\glslongtok}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glssetattribute{\the\glslabeltok}{regular}{true}}%
  }%
  {%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
    \renewcommand\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
    \renewcommand\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
    \renewcommand\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```

The inline full form displays the short form followed by the long form in parentheses.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%

    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
```

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.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
```

```
        \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\Glsxtrfullplformat}[2]{%
        \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\GLSxtrfullformat}[2]{%
        \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
      }%
      \renewcommand*{\GLSxtrfullplformat}[2]{%
        \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
      }%
    }
```

short-sc-nolong

```
        \letabbreviationstyle{short-sc-nolong}{short-sc}
```

short-sc-desc

```
        \newabbreviationstyle{short-sc-desc}%
        {%
```
Set accessibility attributes if enabled.
```
        \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```
Setup the default fields.
```
        \renewcommand*{\CustomAbbreviationFields}{%
          name={\glsxtrshortdescname},
          sort={\the\glsshorttok},
          first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
          firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
          text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
          plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
        \renewcommand*{\GlsXtrPostNewAbbreviation}{%
          \glssetattribute{\the\glslabeltok}{regular}{true}}%
        }%
        {%
```
Use smallcaps and adjust the plural suffix to revert to upright.
```
        \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
        \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
        \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
        \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
        \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
        \renewcommand*\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```
The inline full form displays the short format followed by the long form in
parentheses.
```
        \renewcommand*{\glsxtrinlinefullformat}[2]{%
          \glsxtrshortlongformat{##1}{##2}%
            {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\glsxtrinlinefullplformat}[2]{%
```

485

```
          \glsxtrshortlongplformat{##1}{##2}%
            {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\Glsxtrinlinefullformat}[2]{%
          \Glsxtrshortlongformat{##1}{##2}%
            {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
          \Glsxtrshortlongplformat{##1}{##2}%
            {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\GLSxtrinlinefullformat}[2]{%
          \GLSxtrshortlongformat{##1}{##2}%
            {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
          \GLSxtrshortlongplformat{##1}{##2}%
            {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
        }%
      }
```

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.

```
        \renewcommand*{\glsxtrfullformat}[2]{%
          \glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\glsxtrfullplformat}[2]{%
          \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\Glsxtrfullformat}[2]{%
          \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\Glsxtrfullplformat}[2]{%
          \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\GLSxtrfullformat}[2]{%
          \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
        }%
        \renewcommand*{\GLSxtrfullplformat}[2]{%
          \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
        }%
      }
```

short-sc-nolong-desc

```
        \letabbreviationstyle{short-sc-nolong-desc}{short-sc-desc}
```

nolong-short-sc

```
        \newabbreviationstyle{nolong-short-sc}%
        {%
          \GlsXtrUseAbbrStyleSetup{short-sc-nolong}%
        }%
```

```
  {%
    \GlsXtrUseAbbrStyleFmts{short-sc-nolong}%
```
The inline full form displays the long form followed by the short form in parentheses.
```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
  }
```

long-noshort-sc The smallcaps font will only be used if the short form is explicitly invoked through commands like \glsxtrshort. No accessibility attributes needed here.
```
\newabbreviationstyle{long-noshort-sc}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
    firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
    text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
    plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```
Use smallcaps and adjust the plural suffix to revert to upright.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
\renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
\renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%
\renewcommand*\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \Glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \GLSxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \GLSxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
```

The inline full form displays the long format followed by the short form in parentheses.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlongshortplformat{##1}{##2}%
```

```
      {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
    }%
```

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.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
  }
```

long-sc Backward compatibility:

```
    \@glsxtr@deprecated@abbrstyle{long-sc}{long-noshort-sc}
```

long-noshort-sc-desc The smallcaps font will only be used if the short form is explicitly invoked through commands like `\glsxtrshort`.

```
    \newabbreviationstyle{long-noshort-sc-desc}%
    {%
      \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
    }%
    {%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
    \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
    \renewcommand*\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
    \renewcommand*{\glsxtrsubsequentfmt}[2]{%
      \glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
    }%
    \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
      \glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
    }%
```

```
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \Glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \GLSxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
  \GLSxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
```

The inline full form displays the long format followed by the short form in parentheses.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
```

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.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
```

```
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
  }
```

long-desc-sc Backward compatibility:

```
\@glsxtr@deprecated@abbrstyle{long-desc-sc}{long-noshort-sc-desc}
```

short-sc-footnote

```
\newabbreviationstyle{short-sc-footnote}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
     {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
     {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the
regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
```

```
\renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
\renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
\renewcommand*\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```
The full format displays the short form followed by the long form as a footnote.
```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%
```
The first use full form and the inline full form use the short (long) style.
```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
}%
```

492

```
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
  }
```

footnote-sc   Backward compatibility:

```
    \@glsxtr@deprecated@abbrstyle{footnote-sc}{short-sc-footnote}
```

short-sc-footnote-desc   Like short-sc-footnote but with user supplied description.

```
    \newabbreviationstyle{short-sc-footnote-desc}%
    {%
```

Set accessibility attributes if enabled.

```
    \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glsxtrfootnotedescname},
      sort={\glsxtrfootnotedescsort},
      first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
       \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
         {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
      firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
       \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
         {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
      text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
      plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the
regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
    \GlsXtrUseAbbrStyleFmts{short-sc-footnote}%
  }
```

493

short-sc-postfootnote

```
\newabbreviationstyle{short-sc-postfootnote}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

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. The footnote needs to be suppressed in the inline form. Previously this was done by redefining glsxtrsetupfulldefs but that interferes with other styles. Instead, this now uses \glsxtrifwasglslike

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasglslikeandfirstuse
    {%
      \glsxtrdopostpunc{\xpglsxtrpostabbrvfootnote}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
\renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
\renewcommand*\glslongfont[1]{\glslongfootnotefont{##1}}%
\renewcommand*\glsxtrrevert[1]{\glsxtrscrevert{##1}}%
```

The full format displays the short form. The long form is deferred.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
```

```
      \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
    }%
```

The first use full form and the inline full form use the short (long) style.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
    }%
  }
```

postfootnote-sc   Backward compatibility:

```
    \@glsxtr@deprecated@abbrstyle{postfootnote-sc}{short-sc-postfootnote}
```

short-sc-postfootnote-desc   Like short-sc-footnote but with user supplied description.

```
    \newabbreviationstyle{short-sc-postfootnote-desc}%
    {%
```

Set accessibility attributes if enabled.

```
    \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

495

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotedescname},
  sort={\glsxtrfootnotedescsort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

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. The footnote needs to be suppressed in the inline form. Previously this was done by redefining glsxtrsetupfulldefs but that interferes with other styles. Instead, this now uses \glsxtrifwasglslike

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasglslikeandfirstuse
    {%
      \glsxtrdopostpunc{\xpglsxtrpostabbrvfootnote}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-postfootnote}%
}
```

2.3   Predefined Styles (Fake Small Capitals)

These styles require the relsize package, which must be loaded by the user. These styles all use:

\glsxtrsmfont   Maintained for backward compatibility.

```
\newcommand*{\glsxtrsmfont}[1]{\textsmaller{#1}}
```

\glsabbrvsmfont   Added for consistent naming.

```
\newcommand*{\glsabbrvsmfont}{\glsxtrsmfont}
```

\glsxtrfirstsmfont   Maintained for backward compatibility.

```
\newcommand*{\glsxtrfirstsmfont}[1]{\glsabbrvsmfont{#1}}
```

\glsfirstabbrvsmfont   Added for consistent naming.

```
\newcommand*{\glsfirstabbrvsmfont}{\glsxtrfirstsmfont}
```

and for the default short form suffix:

```
\newcommand*{\glsxtrsmsuffix}{\glsxtrabbrvpluralsuffix}
```

```
\newcommand*{\glsxtrsmrevert}[1]{\textlarger{#1}}
```

```
\newabbreviationstyle{long-short-sm}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
    firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%

  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
```

Use the default long fonts.

```
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrlongshortformat{##1}{##2}%
```

```
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
          }%
          \renewcommand*{\glsxtrfullplformat}[2]{%
            \glsxtrlongshortplformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
          }%
          \renewcommand*{\Glsxtrfullformat}[2]{%
            \Glsxtrlongshortformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
          }%
          \renewcommand*{\Glsxtrfullplformat}[2]{%
            \Glsxtrlongshortplformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
          }%
          \renewcommand*{\GLSxtrfullformat}[2]{%
            \GLSxtrlongshortformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
          }%
          \renewcommand*{\GLSxtrfullplformat}[2]{%
            \GLSxtrlongshortplformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
          }%
        }
```

long-short-sm-desc

```
        \newabbreviationstyle{long-short-sm-desc}%
        {%
```

Set accessibility attributes if enabled.

```
          \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
          \renewcommand*{\CustomAbbreviationFields}{%
            name={\glsxtrlongshortdescname},
            sort={\glsxtrlongshortdescsort},%
            first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
             \protect\glsxtrfullsep{\the\glslabeltok}%
             \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
            firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
             \protect\glsxtrfullsep{\the\glslabeltok}%
             \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
            text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
            plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
          }%
```

Unset the regular attribute if it has been set.

```
          \renewcommand*{\GlsXtrPostNewAbbreviation}{%
            \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
            \glshasattribute{\the\glslabeltok}{regular}%
            {%
              \glssetattribute{\the\glslabeltok}{regular}{false}%
```

498

```
          }%
        {}%
      }%
    }%
    {%
```

As long-short-sm style:

```
      \GlsXtrUseAbbrStyleFmts{long-short-sm}%
    }
```

Now the short (long) version

```
    \newabbreviationstyle{short-sm-long}%
    {%
```

Set accessibility attributes if enabled.

```
      \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
      \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtrshortlongname},
        sort={\the\glsshorttok},
        description={\the\glslongtok},%
        first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
        firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
        text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
        plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Unset the regular attribute if it has been set.

```
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
    }%
    {%

      \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
      \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
      \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
      \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
      \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
      \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
      \renewcommand*{\glsxtrfullformat}[2]{%
```

```
            \glsxtrshortlongformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
        }%
        \renewcommand*{\glsxtrfullplformat}[2]{%
            \glsxtrshortlongplformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
        }%
        \renewcommand*{\Glsxtrfullformat}[2]{%
            \Glsxtrshortlongformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
        }%
        \renewcommand*{\Glsxtrfullplformat}[2]{%
            \Glsxtrshortlongplformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
        }%
        \renewcommand*{\GLSxtrfullformat}[2]{%
            \GLSxtrshortlongformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
        }%
        \renewcommand*{\GLSxtrfullplformat}[2]{%
            \GLSxtrshortlongplformat{##1}{##2}%
                {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
        }%
    }
```

short-sm-long-desc As before but user provides description

```
\newabbreviationstyle{short-sm-long-desc}%
{%
```

Set accessibility attributes if enabled.

```
    \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
    \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtrshortlongdescname},
        sort={\glsxtrshortlongdescsort},
        first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}}},%
        firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}}},%
        text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
        plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
    }%
```

Unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
```

```
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
```

As short-sm-long style:

```
    \GlsXtrUseAbbrStyleFmts{short-sm-long}%
  }
```

short-sm

```
  \newabbreviationstyle{short-sm}%
  {%
```

Set accessibility attributes if enabled.

```
    \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glsxtrshortnolongname},
      sort={\the\glsshorttok},
      first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
      firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
      text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
      plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
      description={\the\glslongtok}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glssetattribute{\the\glslabeltok}{regular}{true}}%
  }%
  {%

    \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short form followed by the long form in paren-
theses.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%

    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
```

```
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
        \Glsxtrshortlongplformat{##1}{##2}%
          {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\GLSxtrinlinefullformat}[2]{%
        \GLSxtrshortlongformat{##1}{##2}%
          {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
        \GLSxtrshortlongplformat{##1}{##2}%
          {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
      }%
```

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.

```
      \renewcommand*{\glsxtrfullformat}[2]{%
        \glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\glsxtrfullplformat}[2]{%
        \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\Glsxtrfullformat}[2]{%
        \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\Glsxtrfullplformat}[2]{%
        \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\GLSxtrfullformat}[2]{%
        \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
      }%
      \renewcommand*{\GLSxtrfullplformat}[2]{%
        \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
      }%
    }
```

short-sm-nolong

```
            \letabbreviationstyle{short-sm-nolong}{short-sm}
```

short-sm-desc

```
            \newabbreviationstyle{short-sm-desc}%
            {%
```

Set accessibility attributes if enabled.

```
            \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
            \renewcommand*{\CustomAbbreviationFields}{%
              name={\glsxtrshortdescname},
```

```
    sort={\the\glsshorttok},
    first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
    firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
  \renewcommand*\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%

  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short format followed by the long form in parentheses.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
```

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.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
```

```
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
  }
```

short-sm-nolong-desc

```
    \letabbreviationstyle{short-sm-nolong-desc}{short-sm-desc}
```

nolong-short-sm

```
    \newabbreviationstyle{nolong-short-sm}%
    {%
      \GlsXtrUseAbbrStyleSetup{short-sm-nolong}%
    }%
    {%
      \GlsXtrUseAbbrStyleFmts{short-sm-nolong}%
```

The inline full form displays the long form followed by the short form in parentheses.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrlongshortplformat{##1}{##2}%
```

```
                   {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
              }%
           }
```

long-noshort-sm The smallcaps font will only be used if the short form is explicitly invoked
                 through commands like \glsxtrshort.

```
\newabbreviationstyle{long-noshort-sm}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongnoshortname},
  sort={\the\glsshorttok},
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
  text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
  plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}},%
  description={\the\glslongtok}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%

  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \Glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \GLSxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
```

```
    \GLSxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
```
The inline full form displays the long format followed by the short form in parentheses.
```
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsxtrlongshortplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \Glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \Glsxtrlongshortplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
  \renewcommand*{\GLSxtrinlinefullformat}[2]{%
    \GLSxtrlongshortformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
  \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
    \GLSxtrlongshortplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
```
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.
```
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \Glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \Glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrfullformat}[2]{%
    \GLSxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrfullplformat}[2]{%
    \GLSxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
  }%
 }
```

Backward compatibility:

```
\@glsxtr@deprecated@abbrstyle{long-sm}{long-noshort-sm}
```

The smaller font will only be used if the short form is explicitly invoked through commands like \glsxtrshort.

```
\newabbreviationstyle{long-noshort-sm-desc}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
}%
{%

  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \Glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \Glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentfmt}[2]{%
    \GLSxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
    \GLSxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
```

The inline full form displays the long format followed by the short form in parentheses.

```
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsxtrlongshortplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \Glsxtrlongshortformat{##1}{##2}%
```

```
                       {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
                    }%
                    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
                       \Glsxtrlongshortplformat{##1}{##2}%
                          {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
                    }%
                    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
                       \GLSxtrlongshortformat{##1}{##2}%
                          {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
                    }%
                    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
                       \GLSxtrlongshortplformat{##1}{##2}%
                          {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
                    }%
```

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.

```
                    \renewcommand*{\glsxtrfullformat}[2]{%
                       \glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
                    }%
                    \renewcommand*{\glsxtrfullplformat}[2]{%
                       \glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
                    }%
                    \renewcommand*{\Glsxtrfullformat}[2]{%
                       \Glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
                    }%
                    \renewcommand*{\Glsxtrfullplformat}[2]{%
                       \Glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
                    }%
                    \renewcommand*{\GLSxtrfullformat}[2]{%
                       \GLSxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
                    }%
                    \renewcommand*{\GLSxtrfullplformat}[2]{%
                       \GLSxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
                    }%
                 }
```

long-desc-sm Backward compatibility:

```
                 \@glsxtr@deprecated@abbrstyle{long-desc-sm}{long-noshort-sm-desc}
```

short-sm-footnote

```
                 \newabbreviationstyle{short-sm-footnote}%
                 {%
```

Set accessibility attributes if enabled.

```
                 \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
                 \renewcommand*{\CustomAbbreviationFields}{%
                    name={\glsxtrfootnotename},
                    sort={\the\glsshorttok},
```

```
        description={\the\glslongtok},%
        first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
         \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
            {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
        firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
         \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
            {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
        text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
        plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%

    \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form followed by the long form as a footnote.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
    }%
```

```
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}}%
}%
```

The first use full form and the inline full form use the short (long) style.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
}%
}
```

footnote-sm Backward compatibility:

```
\@glsxtr@deprecated@abbrstyle{footnote-sm}{short-sm-footnote}
```

short-sm-footnote-desc Like short-footnote but with user supplied description.

```
\newabbreviationstyle{short-sm-footnote-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotedescname},
```

510

```
                    sort={\glsxtrfootnotedescsort},
                    first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
                     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
                        {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
                    firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
                     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
                        {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
                    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
                    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the
regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sm-footnote}%
}
```

short-sm-postfootnote

```
  \newabbreviationstyle{short-sm-postfootnote}%
  {%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
    firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

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. The footnote needs to be suppressed
in the inline form. Previously this was done by redefining glsxtrsetupfulldefs but
that interferes with other styles. Instead, this now uses \glsxtrifwasglslike

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasglslikeandfirstuse
      {%
        \glsxtrdopostpunc{\xpglsxtrpostabbrvfootnote}%
```

```
        }%
        {}%
      }%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%

    \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtrsmrevert{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form. The long form is deferred.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvsmfont}%
    }%
```

The first use full form and the inline full form use the short (long) style.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
```

```
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
    }%
  }
```

postfootnote-sm Backward compatibility:

```
\@glsxtr@deprecated@abbrstyle{postfootnote-sm}{short-sm-postfootnote}
```

short-sm-postfootnote-desc Like short-sm-postfootnote but with user supplied description.

```
\newabbreviationstyle{short-sm-postfootnote-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotedescname},
  sort={\glsxtrfootnotedescsort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

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. The footnote needs to be suppressed
in the inline form. Previously this was done by redefining glsxtrsetupfulldefs but
that interferes with other styles. Instead, this now uses \glsxtrifwasglslike

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasglslikeandfirstuse
    {%
```

Ensure \glslabel is expanded as it may be lost by the time the footnote occurs.

```
      \glsxtrdopostpunc{\xpglsxtrpostabbrvfootnote}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
```

513

```
        }%
        {}%
      }%
    }%
    {%
      \GlsXtrUseAbbrStyleFmts{short-sm-postfootnote}%
    }
```

2.4 Predefined Styles (Emphasized)

These styles use \emph for the short form.

\glsabbrvemfont

```
\newcommand*{\glsabbrvemfont}[1]{\emph{#1}}%
```

\glsfirstabbrvemfont

```
\newcommand*{\glsfirstabbrvemfont}[1]{\glsabbrvemfont{#1}}%
```

The default short form suffix:

\glsxtremsuffix

```
\newcommand*{\glsxtremsuffix}{\glsxtrabbrvpluralsuffix}
```

\glsfirstlongemfont Only used by the "long-em" styles.

```
\newcommand*{\glsfirstlongemfont}[1]{\glslongemfont{#1}}%
```

\glslongemfont Only used by the "long-em" styles.

```
\newcommand*{\glslongemfont}[1]{\emph{#1}}%
```

\glsxtremrevert

```
\newcommand*{\glsxtremrevert}[1]{\textup{#1}}%
```

long-short-em The long form is just set in the default long font.

```
\newabbreviationstyle{long-short-em}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
    firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
```

```
      plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
      description={\the\glslongtok}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%

    \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvemfont{##1}}%
    \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
```

Use the default long fonts.

```
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
  }
```

long-short-em-desc

```
\newabbreviationstyle{long-short-em-desc}%
 {%
```
Set accessibility attributes if enabled.
```
    \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```
Setup the default fields.
```
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glsxtrlongshortdescname},
      sort={\glsxtrlongshortdescsort},%
      first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
      firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
      text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
      plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
    }%
```
Unset the regular attribute if it has been set.
```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
 }%
 {%
```
As long-short-em style:
```
    \GlsXtrUseAbbrStyleFmts{long-short-em}%
 }
```

long-em-short-em

```
    \newabbreviationstyle{long-em-short-em}%
 {%
```
Set accessibility attributes if enabled.
```
    \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```
Setup the default fields. `\glslongemfont` is used in the description since `\glsdesc` doesn't set the style.
```
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glsxtrlongshortname},
      sort={\the\glsshorttok},
      first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
      firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
```

516

```
    \protect\glsxtrfullsep{\the\glslabeltok}%
    \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}},%

    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
    description={\protect\glslongemfont{\the\glslongtok}}}%
```

Unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
      \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%

    \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
    \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
```

```
                            }%
                          }
```

long-em-short-em-desc

```
                          \newabbreviationstyle{long-em-short-em-desc}%
                          {%
```

Set accessibility attributes if enabled.

```
                            \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
                            \renewcommand*{\CustomAbbreviationFields}{%
                              name={\glsxtrlongshortdescname},
                              sort={\glsxtrlongshortdescsort},%
                              first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
                               \protect\glsxtrfullsep{\the\glslabeltok}%
                               \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
                              firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
                               \protect\glsxtrfullsep{\the\glslabeltok}%
                               \protect\glsxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
                              text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
                              plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
                            }%
```

Unset the regular attribute if it has been set.

```
                            \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                              \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
                              \glshasattribute{\the\glslabeltok}{regular}%
                              {%
                                \glssetattribute{\the\glslabeltok}{regular}{false}%
                              }%
                              {}%
                            }%
                          }%
                          {%
                            \GlsXtrUseAbbrStyleFmts{long-em-short-em}%
                          }
```

short-em-long Now the short (long) version

```
                          \newabbreviationstyle{short-em-long}%
                          {%
```

Set accessibility attributes if enabled.

```
                            \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
                            \renewcommand*{\CustomAbbreviationFields}{%
                              name={\glsxtrshortlongname},
                              sort={\the\glsshorttok},
                              description={\the\glslongtok},%
                              first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
```

```
      \protect\glsxtrfullsep{\the\glslabeltok}%
      \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
     firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
      \protect\glsxtrfullsep{\the\glslabeltok}%
      \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
     text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
     plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
     \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
     \glshasattribute{\the\glslabeltok}{regular}%
     {%
       \glssetattribute{\the\glslabeltok}{regular}{false}%
     }%
     {}%
    }%
  }%
  {%
```

Mostly as short-long style:

```
    \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
    \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvemfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
     \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
     \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
     \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
     \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
     \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
     \GLSxtrshortlongplformat{##1}{##2}%
```

```
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
      }%
    }
```

short-em-long-desc As before but user provides description

```
    \newabbreviationstyle{short-em-long-desc}%
    {%
```

Set accessibility attributes if enabled.

```
      \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
      \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtrshortlongdescname},
        sort={\glsxtrshortlongdescsort},
        first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
        firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
        text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
        plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
      }%
```

Unset the regular attribute if it has been set.

```
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
    }%
    {%
      \GlsXtrUseAbbrStyleFmts{short-em-long}%
    }
```

short-em-long-em

```
    \newabbreviationstyle{short-em-long-em}%
    {%
```

Set accessibility attributes if enabled.

```
      \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields. \glslongemfont is used in the description since
\glsdesc doesn't set the style.

```
      \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtrshortlongname},
        sort={\the\glsshorttok},
```

```
      description={\protect\glslongemfont{\the\glslongtok}},%
      first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}}},%
      firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}}},%

      text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
      plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}}%
```

Unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
      \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%

    \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
    \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
```

```
    }%
  \renewcommand*{\GLSxtrfullplformat}[2]{%
    \GLSxtrshortlongplformat{##1}{##2}%
      {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
  }%
}
```

short-em-long-em-desc

```
\newabbreviationstyle{short-em-long-em-desc}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},%
    sort={\glsxtrshortlongdescsort},%
    first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}}},%
    firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}}},%
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
  }%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-long-em}%
}
```

short-em

```
\newabbreviationstyle{short-em}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
```

```
      name={\glsxtrshortnolongname},
      sort={\the\glsshorttok},
      first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
      firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
      text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
      plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
      description={\the\glslongtok}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glssetattribute{\the\glslabeltok}{regular}{true}}%
  }%
  {%

  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short form followed by the long form in parentheses.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
```

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.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
    }%
```

```
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
}
```

short-em-nolong

```
\letabbreviationstyle{short-em-nolong}{short-em}
```

short-em-desc

```
\newabbreviationstyle{short-em-desc}%
{%
```

Set accessibility attributes if enabled. The default name includes the long form
but \glsxtrshortdescname could be modified to omit the long form, so include
the nameshortaccess attribute.

```
\glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortdescname},
  sort={\the\glsshorttok},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%

  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short format followed by the long form in
parentheses.
```

524

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
}
```

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.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
}
```

short-em-nolong-desc

```
\letabbreviationstyle{short-em-nolong-desc}{short-em-desc}
```

nolong-short-em

```
\newabbreviationstyle{nolong-short-em}%
{%
  \GlsXtrUseAbbrStyleSetup{short-em-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-nolong}%
```

The inline full form displays the long form followed by the short form in parentheses.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
}
```

long-noshort-em The short form is explicitly invoked through commands like `\glsxtrshort`.

```
\newabbreviationstyle{long-noshort-em}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongnoshortname},
  sort={\the\glsshorttok},
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
  text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
```

```
    plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%

  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \Glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \Glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentfmt}[2]{%
    \GLSxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
    \GLSxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
```

The inline full form displays the long format followed by the short form in parentheses.

```
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsxtrlongshortplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \Glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \Glsxtrlongshortplformat{##1}{##2}%
```

```
      {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
    }%
```

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.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
    }%
  }
```

long-em Backward compatibility:

```
    \@glsxtr@deprecated@abbrstyle{long-em}{long-noshort-em}
```

long-em-noshort-em The short form is explicitly invoked through commands like `\glsxtrshort`.

```
    \newabbreviationstyle{long-em-noshort-em}%
    {%
```

Set accessibility attributes if enabled.

```
    \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glsxtrlongnoshortname},
      sort={\the\glsshorttok},
      first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
      firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
      text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
      plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}},%
```

528

```
      description={\protect\glslongemfont{\the\glslongtok}}%
    }%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
      \glssetattribute{\the\glslabeltok}{regular}{true}}%
  }%
  {%

    \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
    \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
    \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
    \renewcommand*{\glsxtrsubsequentfmt}[2]{%
      \glsxtrlongformat{##1}{##2}{\glslongemfont}%
    }%
    \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
      \glsxtrlongplformat{##1}{##2}{\glslongemfont}%
    }%
    \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
      \Glsxtrlongformat{##1}{##2}{\glslongemfont}%
    }%
    \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
      \Glsxtrlongplformat{##1}{##2}{\glslongemfont}%
    }%
    \renewcommand*{\GLSxtrsubsequentfmt}[2]{%
      \GLSxtrlongformat{##1}{##2}{\glslongemfont}%
    }%
    \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
      \GLSxtrlongplformat{##1}{##2}{\glslongemfont}%
    }%
```

The inline full form displays the long format followed by the short form in parentheses.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrlongshortplformat{##1}{##2}%
```

```
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
      }%
      \renewcommand*{\GLSxtrinlinefullformat}[2]{%
        \GLSxtrlongshortformat{##1}{##2}%
          {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
      }%
      \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
        \GLSxtrlongshortplformat{##1}{##2}%
          {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
      }%
```

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.

```
      \renewcommand*{\glsxtrfullformat}[2]{%
        \glsxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
      }%
      \renewcommand*{\glsxtrfullplformat}[2]{%
        \glsxtrlongplformat{##1}{##2}{\glsfirstlongemfont}%
      }%
      \renewcommand*{\Glsxtrfullformat}[2]{%
        \Glsxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
      }%
      \renewcommand*{\Glsxtrfullplformat}[2]{%
        \Glsxtrlongplformat{##1}{##2}{\glsfirstlongemfont}%
      }%
      \renewcommand*{\GLSxtrfullformat}[2]{%
        \GLSxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
      }%
      \renewcommand*{\GLSxtrfullplformat}[2]{%
        \GLSxtrlongplformat{##1}{##2}{\glsfirstlongemfont}%
      }%
   }
```

long-em-noshort-em-noreg Like long-em-noshort-em but doesn't set the regular attribute.

```
    \newabbreviationstyle{long-em-noshort-em-noreg}%
    {%
```

Set accessibility attributes if enabled.

```
      \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
      \GlsXtrUseAbbrStyleSetup{long-em-noshort-em}%
```

Unset the regular attribute if it has been set.

```
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
```

```
    }%
    {%
      \GlsXtrUseAbbrStyleFmts{long-em-noshort-em}%
    }
```

long-noshort-em-desc The emphasized font will only be used if the short form is explicitly invoked
through commands like `\glsxtrshort`.

```
\newabbreviationstyle{long-noshort-em-desc}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
}%
{%

  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \Glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \Glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentfmt}[2]{%
    \GLSxtrlongformat{##1}{##2}{\glslongdefaultfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
    \GLSxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
  }%
```

The inline full form displays the long format followed by the short form in
parentheses.

```
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsxtrlongshortplformat{##1}{##2}%
      {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
  }%
```

```
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
```

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.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrlongplformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
}
```

long-desc-em   Backward compatibility:

```
\@glsxtr@deprecated@abbrstyle{long-desc-em}{long-noshort-em-desc}
```

long-em-noshort-em-desc   The short form is explicitly invoked through commands like \glsxtrshort. The long form is emphasized. No accessibility attributes need to be set.

```
\newabbreviationstyle{long-em-noshort-em-desc}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortdescname},
    sort={\the\glslongtok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
```

532

```
        firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
        text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
        plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}}}%
    }%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glssetattribute{\the\glslabeltok}{regular}{true}}%
  }%
  {%

  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
```

The format for subsequent use (not used when the regular attribute is set).

```
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsxtrlongformat{##1}{##2}{\glslongemfont}%
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glsxtrlongplformat{##1}{##2}{\glslongemfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \Glsxtrlongformat{##1}{##2}{\glslongemfont}%
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \Glsxtrlongplformat{##1}{##2}{\glslongemfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentfmt}[2]{%
    \GLSxtrlongformat{##1}{##2}{\glslongemfont}%
  }%
  \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
    \GLSxtrlongplformat{##1}{##2}{\glslongemfont}%
  }%
```

The inline full form displays the long format followed by the short form in parentheses.

```
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsxtrlongshortplformat{##1}{##2}%
      {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \Glsxtrlongshortformat{##1}{##2}%
      {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
```

```
    \Glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\GLSxtrinlinefullformat}[2]{%
    \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
  }%
  \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
    \GLSxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
  }%
}
```

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.

```
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsxtrlongplformat{##1}{##2}{\glsfirstlongemfont}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \Glsxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \Glsxtrlongplformat{##1}{##2}{\glsfirstlongemfont}%
  }%
  \renewcommand*{\GLSxtrfullformat}[2]{%
    \GLSxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
  }%
  \renewcommand*{\GLSxtrfullplformat}[2]{%
    \GLSxtrlongplformat{##1}{##2}{\glsfirstlongemfont}%
  }%
}
```

long-em-noshort-em-desc-noreg Like long-em-noshort-em-desc but doesn't set the regular attribute.

```
\newabbreviationstyle{long-em-noshort-em-desc-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{long-em-noshort-em-desc}%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-em-noshort-em-desc}%
}
```

short-em-footnote

```
\newabbreviationstyle{short-em-footnote}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
     {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
   \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
     {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
 }%
}%
{%

  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form followed by the long form as a footnote.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
  \protect\glsxtrabbrvfootnote{##1}%
    {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
```

```
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
    }%
```

The first use inline full form uses the short (long) style.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
  }
```

footnote-em Backward compatibility:

```
                    \@glsxtr@deprecated@abbrstyle{footnote-em}{short-em-footnote}
```

short-em-footnote-desc Like short-em-footnote but with user supplied description.

```
                    \newabbreviationstyle{short-em-footnote-desc}%
                    {%
```
Set accessibility attributes if enabled.
```
                        \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```
Setup the default fields.
```
                    \renewcommand*{\CustomAbbreviationFields}{%
                      name={\glsxtrfootnotedescname},
                      sort={\glsxtrfootnotedescsort},
                      first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
                       \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
                         {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
                      firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
                       \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
                         {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
                      text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
                      plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```
Switch off hyperlinks on first use to prevent nested hyperlinks, and unset the
regular attribute if it has been set.
```
                    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                      \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
                      \glshasattribute{\the\glslabeltok}{regular}%
                      {%
                        \glssetattribute{\the\glslabeltok}{regular}{false}%
                      }%
                      {}%
                    }%
                    }%
                    {%
                      \GlsXtrUseAbbrStyleFmts{short-em-footnote}%
                    }
```

short-em-postfootnote
```
                    \newabbreviationstyle{short-em-postfootnote}%
                    {%
```
Set accessibility attributes if enabled.
```
                        \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```
Setup the default fields.
```
                    \renewcommand*{\CustomAbbreviationFields}{%
                      name={\glsxtrfootnotename},
                      sort={\the\glsshorttok},
                      description={\the\glslongtok},%
                      first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
                      firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
```

```
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

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. Previously this was done by redefining glsxtrsetupfulldefs but that interferes with other styles. Instead, this now uses `\glsxtrifwasglslike`

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasglslikeandfirstuse
    {%
```

Ensure `\glslabel` is expanded as it may be lost by the time the footnote occurs.

```
      \glsxtrdopostpunc{\xpglsxtrpostabbrvfootnote}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
  }%
}%
{%

  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsxtrrevert[1]{\glsxtremrevert{##1}}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
```

The full format displays the short form. The long form is deferred.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvemfont}%
```

```
  }%
```
The inline full form uses the short (long) style.
```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortlongformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortlongplformat{##1}{##2}%
        {\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
    }%
  }
```

postfootnote-em   Backward compatibility:
```
    \@glsxtr@deprecated@abbrstyle{postfootnote-em}{short-em-postfootnote}
```

short-em-postfootnote-desc   Like short-em-postfootnote but with user supplied description.
```
    \newabbreviationstyle{short-em-postfootnote-desc}%
    {%
```
Set accessibility attributes if enabled.
```
      \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```
Setup the default fields.
```
      \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtrfootnotedescname},
        sort={\glsxtrfootnotedescsort},
        first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
        firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
        text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
        plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```
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. Previously this was done by

539

redefining glsxtrsetupfulldefs but that interferes with other styles. Instead, this
now uses `\glsxtrifwasglslike`

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasglslikeandfirstuse
    {%
      \glsxtrdopostpunc{\xpglsxtrpostabbrvfootnote}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-postfootnote}%
}
```

2.5   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:

\glsxtruserfield   Default is the useri field.

```
\newcommand*{\glsxtruserfield}{useri}
```

\glsxtruserparensep   Separator used inside parenthetical content.

```
\newcommand*{\glsxtruserparensep}{, }
```

\glsxtruserfieldfmt   Used to format the value of the field given by `\glsxtruserfield`.

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

\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.

```
\ifdef\glscurrentfieldvalue
{
  \newcommand*{\glsxtruserparen}[2]{%
    \glsxtrfullsep{#2}%
    \glsxtrparen
    {#1\ifglshasfield{\glsxtruserfield}{#2}%
     {\expandafter\glsxtrgenentrytextfmt\expandafter{\glsxtruserparensep}%
      \glsxtruserfieldfmt{\expandafter\glsxtrgenentrytextfmt\expandafter{\glscurrentfieldvalue}}%
     }{}%
```

```
      }%
    }
  }
  {
    \newcommand*{\glsxtruserparen}[2]{%
      \glsxtrfullsep{#2}%
      \glsxtrparen
      {#1\ifglshasfield{\glsxtruserfield}{#2}%
       {\expandafter\glsxtrgenentrytextfmt\expandafter{\glsxtruserparensep}%
        \glsxtruserfieldfmt{\expandafter\glsxtrgenentrytextfmt\expandafter{\@glo@thisvalue}}%
       }{}%
      }%
    }
  }
```

\GLSxtruserparen As above but converts the user supplied information to all-caps. The first ar-
                  gument should be provided in all-caps if required.

```
\ifdef\glscurrentfieldvalue
{
  \newcommand*{\GLSxtruserparen}[2]{%
    \glsxtrfullsep{#2}%
    \glsxtrparen
    {#1\ifglshasfield{\glsxtruserfield}{#2}%
      {\expandafter\glsxtrgenentrytextfmt\expandafter{\glsxtruserparensep}%
       \glsxtruserfieldfmt{\expandafter\mfirstucMakeUppercase\expandafter{\expandafter
         \glsxtrgenentrytextfmt\expandafter{\glscurrentfieldvalue}}}%
      }{}%
    }%
  }
}
{
  \newcommand*{\GLSxtruserparen}[2]{%
    \glsxtrfullsep{#2}%
    \glsxtrparen
    {#1\ifglshasfield{\glsxtruserfield}{#2}%
      {\expandafter\glsxtrgenentrytextfmt\expandafter{\glsxtruserparensep}%
       \glsxtruserfieldfmt{\expandafter\mfirstucMakeUppercase\expandafter{\expandafter
         \glsxtrgenentrytextfmt\expandafter{\@glo@thisvalue}}}%
      }{}%
    }%
  }
}
```

Font used for short form:

\glsabbrvuserfont

```
\newcommand*{\glsabbrvuserfont}[1]{\glsabbrvdefaultfont{#1}}
```

Font used for short form on first use:

`\glsfirstabbrvuserfont`

> `\newcommand*{\glsfirstabbrvuserfont}[1]{\glsabbrvuserfont{#1}}`

Font used for long form:

`\glslonguserfont`

> `\newcommand*{\glslonguserfont}[1]{\glslongdefaultfont{#1}}`

Font used for long form on first use:

`\glsfirstlonguserfont`

> `\newcommand*{\glsfirstlonguserfont}[1]{\glslonguserfont{#1}}`

The default short form suffix:

`\glsxtrusersuffix`

> `\newcommand*{\glsxtrusersuffix}{\glsxtrabbrvpluralsuffix}`

Description encapsulator.

`\glsuserdescription` The first argument is the description. The second argument is the label.

> `\newcommand*{\glsuserdescription}[2]{\glslonguserfont{#1}}`

`long-short-user`

> `\newabbreviationstyle{long-short-user}%`
> `{%`

Set accessibility attributes if enabled.

> `\glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel`

Setup the default fields.

> `\renewcommand*{\CustomAbbreviationFields}{%`
> `  name={\glsxtrlongshortname},`
> `  sort={\the\glsshorttok},`
> `  first={\protect\glsfirstlonguserfont{\the\glslongtok}%`
> `   \protect\glsxtruserparen{\protect\glsfirstabbrvuserfont{\the\glsshorttok}}%`
> `     {\the\glslabeltok}},%`
> `  firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}%`
> `   \protect\glsxtruserparen`
> `     {\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%`
>
> `  text={\protect\glsabbrvuserfont{\the\glsshorttok}},%`
> `  plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%`
> `  description={\protect\glsuserdescription{\the\glslongtok}%`
> `   {\the\glslabeltok}}}%`

Unset the regular attribute if it has been set.

> `\renewcommand*{\GlsXtrPostNewAbbreviation}{%`
> `  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%`
> `  \glsxtrsetcomplexstyle{\the\glslabeltok}{2}%`
> `  \glshasattribute{\the\glslabeltok}{regular}%`
> `  {%`

```
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
    \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtruserlongshortformat{##1}{##2}%
        {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtruserlongshortplformat{##1}{##2}%
        {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtruserlongshortformat{##1}{##2}%
        {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtruserlongshortplformat{##1}{##2}%
        {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtruserlongshortformat{##1}{##2}%
        {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtruserlongshortplformat{##1}{##2}%
        {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
    }%
  }
```

long-postshort-user Like long-short-user but defers the parenthetical matter to after the link.

```
    \newabbreviationstyle{long-postshort-user}%
    {%
```

Set accessibility attributes if enabled.

```
    \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glsxtrlongshortname},
```

```
        sort={\the\glsshorttok},
        first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
        firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%

        text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
        plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
        description={\protect\glsuserdescription{\the\glslongtok}%
         {\the\glslabeltok}}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
      \csdef{glsxtrpostlink\glscategorylabel}{%
        \glsxtrifwasfirstuse
        {%
          \glsxtrpostusershortformat{\glslabel}{\glsfirstabbrvuserfont}%
        }%
        {}%
      }%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
    \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
```

First use full form:

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrlongformat{##1}{##2}{\glsfirstlonguserfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrlongplformat{##1}{##2}{\glsfirstlonguserfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrlongformat{##1}{##2}{\glsfirstlonguserfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrlongplformat{##1}{##2}{\glsfirstlonguserfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrlongformat{##1}{##2}{\glsfirstlonguserfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrlongplformat{##1}{##2}{\glsfirstlonguserfont}%
```

```
                }%
              }
```

Small-caps is awkward, so support for that is added.

\glsabbrvscuserfont
```
              \newcommand*{\glsabbrvscuserfont}{\glsabbrvscfont}%
```

\glsfirstabbrvscuserfont
```
              \newcommand*{\glsfirstabbrvscuserfont}{\glsabbrvscuserfont}%
```

The default short form suffix:

\glsxtrscusersuffix
```
              \newcommand*{\glsxtrscusersuffix}{\glsxtrscsuffix}
```

\glsxtrscuserrevert
```
              \newcommand*{\glsxtrscuserrevert}{\glsxtrscrevert}
```

\glsxtrlongshortscusername The default name format for this style.
```
              \newcommand*{\glsxtrlongshortscusername}{%
                \protect\glsabbrvscuserfont{\the\glsshorttok}%
              }
```

long-postshort-sc-user Like long-postshort-sc-user but uses smallcaps.
```
              \newabbreviationstyle{long-postshort-sc-user}%
              {%
```

Set accessibility attributes if enabled.
```
              \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.
```
              \renewcommand*{\CustomAbbreviationFields}{%
                name={\glsxtrlongshortscusername},
                sort={\the\glsshorttok},
                first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
                firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
                text={\protect\glsabbrvscuserfont{\the\glsshorttok}},%
                plural={\protect\glsabbrvscuserfont{\the\glsshortpltok}},%
                description={\protect\glsuserdescription{\the\glslongtok}%
                 {\the\glslabeltok}}}%
              \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
                \csdef{glsxtrpostlink\glscategorylabel}{%
                  \glsxtrifwasfirstuse
                  {%
                    \glsxtrpostusershortformat{\glslabel}{\glsfirstabbrvscuserfont}%
                  }%
                  {}%
                }%
```

545

```
          \glshasattribute{\the\glslabeltok}{regular}%
          {%
            \glssetattribute{\the\glslabeltok}{regular}{false}%
          }%
          {}%
        }%
      }%
      {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
        \renewcommand*{\abbrvpluralsuffix}{\glsxtrscusersuffix}%
        \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscuserfont{##1}}%
        \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscuserfont{##1}}%
        \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
        \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
        \renewcommand*\glsxtrrevert[1]{\glsxtrscuserrevert{##1}}%
```

First use full form:

```
        \renewcommand*{\glsxtrfullformat}[2]{%
          \glsxtrlongformat{##1}{##2}{\glsfirstlonguserfont}%
        }%
        \renewcommand*{\glsxtrfullplformat}[2]{%
          \glsxtrlongplformat{##1}{##2}{\glsfirstlonguserfont}%
        }%
        \renewcommand*{\Glsxtrfullformat}[2]{%
          \Glsxtrlongformat{##1}{##2}{\glsfirstlonguserfont}%
        }%
        \renewcommand*{\Glsxtrfullplformat}[2]{%
          \Glsxtrlongplformat{##1}{##2}{\glsfirstlonguserfont}%
        }%
        \renewcommand*{\GLSxtrfullformat}[2]{%
          \GLSxtrlongformat{##1}{##2}{\glsfirstlonguserfont}%
        }%
        \renewcommand*{\GLSxtrfullplformat}[2]{%
          \GLSxtrlongplformat{##1}{##2}{\glsfirstlonguserfont}%
        }%
```

In-line format is the same as the first use format.

```
      }
```

\glsxtrlongshortuserdescname

```
      \newcommand*{\glsxtrlongshortuserdescname}{%
        \protect\glslonguserfont{\the\glslongtok}%
        \protect\glsxtruserparen
          {\protect\glsabbrvuserfont{\the\glsshorttok}}{\the\glslabeltok}%
      }
```

long-postshort-user-desc Like long-postshort-user but the user supplies the description.

```
      \newabbreviationstyle{long-postshort-user-desc}%
      {%
```

546

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortuserdescname},
  sort={\the\glslongtok},
  first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
  firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%

  text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasfirstuse
    {%
      \glsxtrpostusershortformat{\glslabel}{\glsfirstabbrvuserfont}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-postshort-user}%
}
```

```
\newcommand*{\glsxtrlongshortscuserdescname}{%
  \protect\glslonguserfont{\the\glslongtok}%
  \protect\glsxtruserparen
    {\protect\glsabbrvscuserfont{\the\glsshorttok}}{\the\glslabeltok}%
}
```

long-postshort-sc-user-desc Like long-postshort-sc-user but the user supplies the description.

```
\newabbreviationstyle{long-postshort-sc-user-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortscuserdescname},
  sort={\the\glslongtok},
  first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
```

```
                    firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
                    text={\protect\glsabbrvscuserfont{\the\glsshorttok}},%
                    plural={\protect\glsabbrvscuserfont{\the\glsshortpltok}}}%
                  }%
                  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                    \csdef{glsxtrpostlink\glscategorylabel}{%
                      \glsxtrifwasfirstuse
                      {%
                        \glsxtrpostusershortformat{\glslabel}{\glsfirstabbrvscuserfont}%
                      }%
                      {}%
                    }%
                    \glshasattribute{\the\glslabeltok}{regular}%
                    {%
                      \glssetattribute{\the\glslabeltok}{regular}{false}%
                    }%
                    {}%
                  }%
                }%
                {%
                  \GlsXtrUseAbbrStyleFmts{long-postshort-sc-user}%
                }
```

short-postlong-user   Like short-long-user but defers the parenthetical matter to after the link.

```
                \newabbreviationstyle{short-postlong-user}%
                {%
```

Set accessibility attributes if enabled.

```
                \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
                \renewcommand*{\CustomAbbreviationFields}{%
                  name={\glsxtrshortlongname},
                  sort={\the\glsshorttok},
                  first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
                  firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%

                  text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
                  plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
                  description={\protect\glsuserdescription{\the\glslongtok}%
                   {\the\glslabeltok}}}%
                \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
                \csdef{glsxtrpostlink\glscategorylabel}{%
                  \glsxtrifwasfirstuse
                  {%
                    \glsxtrpostuserlongformat{\glslabel}{\glsfirstlonguserfont}%
                  }%
                  {}%
                }%
                \glshasattribute{\the\glslabeltok}{regular}%
```

548

```
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
    }%
    {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
      \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
      \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
      \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
      \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
      \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
```

First use full form:

```
      \renewcommand*{\glsxtrfullformat}[2]{%
        \glsxtrshortformat{##1}{##2}{\glsfirstabbrvuserfont}%
      }%
      \renewcommand*{\glsxtrfullplformat}[2]{%
        \glsxtrshortplformat{##1}{##2}{\glsfirstabbrvuserfont}%
      }%
      \renewcommand*{\Glsxtrfullformat}[2]{%
        \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvuserfont}%
      }%
      \renewcommand*{\Glsxtrfullplformat}[2]{%
        \Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvuserfont}%
      }%
      \renewcommand*{\GLSxtrfullformat}[2]{%
        \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvuserfont}%
      }%
      \renewcommand*{\GLSxtrfullplformat}[2]{%
        \GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvuserfont}%
      }%
```

In-line format should be the same.

```
    }
```

\glsxtrshortlonguserdescname

```
    \newcommand*{\glsxtrshortlonguserdescname}{%
     \protect\glsabbrvuserfont{\the\glsshorttok}%
     \protect\glsxtruserparen
       {\protect\glslonguserfont{\the\glslongtok}}%
       {\the\glslabeltok}%
    }
```

short-postlong-user-desc Like short-postlong-user but leaves the user to specify the description.

```
    \newabbreviationstyle{short-postlong-user-desc}%
    {%
```

Set accessibility attributes if enabled.

```
      \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

549

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortlonguserdescname},
  sort={\the\glsshorttok},
  first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
  firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%

  text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasfirstuse
    {%
      \glsxtrpostuserlongformat{\glslabel}{\glsfirstlonguserfont}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-postlong-user}%
}
```

long-short-user-desc

```
\newabbreviationstyle{long-short-user-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortuserdescname},
  sort={\glsxtrlongshortdescsort},%

  first={\protect\glsfirstlonguserfont{\the\glslongtok}%
   \protect\glsxtruserparen{\protect\glsfirstabbrvuserfont{\the\glsshorttok}}%
     {\the\glslabeltok}},%
  firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}%
   \protect\glsxtruserparen
     {\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%
  text={\protect\glsabbrvfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
}%
```

550

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{2}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-short-user}%
}
```

```
\newabbreviationstyle{short-long-user}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

`\glslonguserfont` is used in the description since `\glsdesc` doesn't set the style. (Now in `\glsuserdescription`.)

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortlongname},
  sort={\the\glsshorttok},
  description={\protect\glsuserdescription{\the\glslongtok}%
    {\the\glslabeltok}},%
  first={\protect\glsfirstabbrvuserfont{\the\glsshorttok}%
   \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongtok}}%
     {\the\glslabeltok}},%
  firstplural={\protect\glsfirstabbrvuserfont{\the\glsshortpltok}%
   \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongpltok}}%
     {\the\glslabeltok}},%

  text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{2}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvuserfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrusershortlongformat{##1}{##2}%
    {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrusershortlongplformat{##1}{##2}%
    {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrusershortlongformat{##1}{##2}%
    {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrusershortlongplformat{##1}{##2}%
    {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrusershortlongformat{##1}{##2}%
    {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrusershortlongplformat{##1}{##2}%
    {\glsfirstlonguserfont}{\glsfirstabbrvuserfont}%
}%
}
```

short-long-user-desc

```
\newabbreviationstyle{short-long-user-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortlonguserdescname},
  sort={\glsxtrshortlongdescsort},%

  first={\protect\glsfirstabbrvuserfont{\the\glsshorttok}%
   \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongtok}}%
     {\the\glslabeltok}},%
  firstplural={\protect\glsfirstabbrvuserfont{\the\glsshortpltok}%
   \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongpltok}}%
```

552

```
      {\the\glslabeltok}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
  }%
```

Unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsxtrsetcomplexstyle{\the\glslabeltok}{2}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
    \GlsXtrUseAbbrStyleFmts{short-long-user}%
  }
```

2.6 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 hyphen. If it does, the insert is added to the parenthetical
material. Note that commands like \glsxtrlong set \glsinsert to empty with
the entire link-text stored in \glscustomtext.

\glsxtrifhyphenstart Checks if the argument starts with a hyphen. The argument may be \glsinsert
so check for that and expand.

```
\newrobustcmd*{\glsxtrifhyphenstart}[3]{%
  \ifx\glsinsert#1\relax
   \expandafter\@glsxtrifhyphenstart#1\relax\relax
     \@end@glsxtrifhyphenstart{#2}{#3}%
  \else
   \@glsxtrifhyphenstart#1\relax\relax\@end@glsxtrifhyphenstart{#2}{#3}%
  \fi
}
```

\@glsxtrifhyphenstart

```
\def\@glsxtrifhyphenstart#1#2\@end@glsxtrifhyphenstart#3#4{%
  \ifx-#1\relax#3\else #4\fi
}
```

\glsxtrlonghyphenshort

$\boxed{\texttt{\textbackslash glsxtrlonghyphenshort}\{\langle \mathit{label} \rangle\}\{\langle \mathit{long} \rangle\}\{\langle \mathit{short} \rangle\}\{\langle \mathit{insert} \rangle\}}$

The ⟨long⟩ and ⟨short⟩ arguments may be the plural form. The ⟨long⟩ argu-
ment may also be the first letter uppercase form. This unfortunately doesn't

fit in with the new `\glsxtrshortformat` etc commands, but is retained for backward-compatibility. This means that the inserted part has to have a separate encapsulation for the inner format. The ⟨*long*⟩ and ⟨*short*⟩ arguments will need to include the inner format.

```
\newcommand*{\glsxtrlonghyphenshort}[4]{%
```

Grouping is needed to localise the redefinitions.

```
  {%
```

If ⟨*insert*⟩ starts with a hyphen, redefine `\glsxtrwordsep` 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 `\glsxtrwordsep` if ⟨*insert*⟩ doesn't start with a hyphen.

```
    \glsxtrifhyphenstart{#4}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \glsfirstlonghyphenfont{#2\ifglsxtrinsertinside
      {\glsxtrgenentrytextfmt{#4}}\fi}%
    \ifglsxtrinsertinside\else{\glsxtrgenentrytextfmt{#4}}\fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\glsfirstabbrvhyphenfont{#3\ifglsxtrinsertinside
      {\glsxtrgenentrytextfmt{#4}}\fi}%
     \ifglsxtrinsertinside\else{\glsxtrgenentrytextfmt{#4}}\fi}%
  }%
}
```

`\GLSxtrlonghyphenshort` As above but convert the insert to uppercase. The long and short should already have the case-change applied.

```
\newcommand*{\GLSxtrlonghyphenshort}[4]{%
  {%
    \glsxtrifhyphenstart{#4}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \glsfirstlonghyphenfont{#2\ifglsxtrinsertinside
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}\fi}%
    \ifglsxtrinsertinside\else
     {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\glsfirstabbrvhyphenfont{#3\ifglsxtrinsertinside
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}\fi}%
     \ifglsxtrinsertinside\else
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}%
     \fi}%
  }%
}
```

`\glsxtrshorthyphennolong`

> `\glsxtrshorthyphennolong{`⟨*label*⟩`}{`⟨*short*⟩`}{`⟨*insert*⟩`}`

The ⟨*short*⟩ argument may be the plural form and may also be the first letter uppercase form.

As \glsxtrlonghyphenshort but where only the short form should be shown.

```
\newcommand*{\glsxtrshorthyphennolong}[3]{%
```

Grouping is needed to localise the redefinitions.

```
 {%
```

If ⟨*insert*⟩ starts with a hyphen, redefine \glsxtrwordsep to a hyphen.

```
  \glsxtrifhyphenstart{#3}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
  \glsfirstabbrvhyphenfont{#2\ifglsxtrinsertinside
    {\glsxtrgenentrytextfmt{#3}}\fi}%
  \ifglsxtrinsertinside\else{\glsxtrgenentrytextfmt{#3}}\fi
 }%
}
```

\GLSxtrshorthyphennolong As above but all-caps.

```
\newcommand*{\GLSxtrshorthyphennolong}[3]{%
 {%
  \glsxtrifhyphenstart{#3}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
  \glsfirstabbrvhyphenfont{#2\ifglsxtrinsertinside
    {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#3}}}\fi}%
  \ifglsxtrinsertinside\else
    {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#3}}}%
  \fi
 }%
}
```

\glsabbrvhyphenfont

```
\newcommand*{\glsabbrvhyphenfont}{\glsabbrvdefaultfont}%
```

\glsfirstabbrvhyphenfont

```
\newcommand*{\glsfirstabbrvhyphenfont}{\glsabbrvhyphenfont}%
```

\glslonghyphenfont

```
\newcommand*{\glslonghyphenfont}{\glslongdefaultfont}%
```

\glsfirstlonghyphenfont

```
\newcommand*{\glsfirstlonghyphenfont}{\glslonghyphenfont}%
```

The default short form suffix:

\glsxtrhyphensuffix

```
\newcommand*{\glsxtrhyphensuffix}{\glsxtrabbrvpluralsuffix}
```

\glsxtrlonghyphenshortsort

```
\newcommand*{\glsxtrlonghyphenshortsort}{\expandonce\glsxtrorgshort}
```

long-hyphen-short-hyphen Designed for use with the markwords attribute.

```
\newabbreviationstyle{long-hyphen-short-hyphen}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortname},
  sort={\glsxtrlonghyphenshortsort},
  first={\protect\glsfirstlonghyphenfont{\the\glslongtok}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}}},%
  firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}}},%
  text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
  description={\protect\glslonghyphenfont{\the\glslongtok}}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrhyphensuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
   {%
     \glsifattribute{##1}{markwords}{true}%
     {%
       \glsaccesslong{##1}%
     }%
     {%
       \glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
     }%
   }%
   {%
     \glsifattribute{##1}{markshortwords}{true}%
     {%
       \glsaccessshort{##1}%
```

```
    }%
    {%
      \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslongpl{##1}%
    }%
    {%
      \glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshortpl{##1}%
    }%
    {%
      \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \Glsaccesslong{##1}%
    }%
    {%
      \Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
```

```
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \Glsaccesslongpl{##1}%
      }%
      {%
        \Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \glsaccessshortpl{##1}%
      }%
      {%
        \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \GLSaccesslong{##1}%
      }%
      {%
        \GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \GLSaccessshort{##1}%
      }%
      {%
        \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
```

```
        \GLSaccesslongpl{##1}%
      }%
      {%
        \GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \GLSaccessshortpl{##1}%
      }%
      {%
        \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
```

Subsequent form also needs checking for a hyphen in case the short form has spaces.

```
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennolong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrshorthyphennolong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshortpl{##1}%
    }%
    {%
      \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennolong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
```

```
        {%
          \Glsaccessshort{##1}%
        }%
        {%
          \Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
      \glsxtrshorthyphennolong{##1}%
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
          \Glsaccessshortpl{##1}%
        }%
        {%
          \Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\GLSxtrsubsequentfmt}[2]{%
      \GLSxtrshorthyphennolong{##1}%
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
          \GLSaccessshort{##1}%
        }%
        {%
          \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
      \GLSxtrshorthyphennolong{##1}%
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
          \GLSaccessshortpl{##1}%
        }%
        {%
          \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
  }
```

Like long-hyphen-short-hyphen but the description must be supplied by the user.

```
\newabbreviationstyle{long-hyphen-short-hyphen-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortdescname},
  sort={\glsxtrlongshortdescsort},
  first={\protect\glsfirstlonghyphenfont{\the\glslongtok}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}}},%
  firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}%
   \protect\glsxtrfullsep{\the\glslabeltok}%
   \protect\glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}}},%
  text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
 }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%
}
```

<div style="border:1px solid;border-radius:8px;background:#ffffcc;padding:8px">

$\texttt{\textbackslash glsxtrlonghyphennoshort}\{\langle label\rangle\}\{\langle long\rangle\}\{\langle insert\rangle\}$

</div>

\glsxtrlonghyphennoshort

 As with `\glsxtrlonghyphenshort` this doesn't fit in with the new `\glsxtrshortformat` so the inserted part has to have a separate encapsulation for the inner format. The ⟨*long*⟩ argument will need to include the inner format.

```
\newcommand*{\glsxtrlonghyphennoshort}[3]{%
```

Grouping is needed to localise the redefinitions.

```
  {%
```

If ⟨*insert*⟩ starts with a hyphen, redefine `\glsxtrwordsep` 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 `\glsxtrwordsep` if ⟨*insert*⟩ doesn't start with a hyphen.

```
                           \glsxtrifhyphenstart{#3}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
                           \glsfirstlonghyphenfont{#2\ifglsxtrinsertinside
                             {\glsxtrgenentrytextfmt{#3}}\fi}%
                           \ifglsxtrinsertinside\else{\glsxtrgenentrytextfmt{#3}}\fi
                         }%
                       }
```

\GLSxtrlonghyphennoshort   As above but convert insert to all-caps.

```
                       \newcommand*{\GLSxtrlonghyphennoshort}[3]{%
                        {%
                          \glsxtrifhyphenstart{#3}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
                          \glsfirstlonghyphenfont{#2\ifglsxtrinsertinside
                            {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#3}}}\fi}%
                          \ifglsxtrinsertinside\else
                            {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#3}}}%
                          \fi
                        }%
                       }
```

trlonghyphennoshortdescsort

```
                       \newcommand*{\glsxtrlonghyphennoshortdescsort}{\expandonce\glsxtrorglong}
```

g-hyphen-noshort-desc-noreg   This version doesn't show the short form (except explicitly with \glsxtrshort).
                       Since \glsxtrshort 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. No accessibility attributes need to be
                       set.

```
                       \newabbreviationstyle{long-hyphen-noshort-desc-noreg}%
                        {%
                          \renewcommand*{\CustomAbbreviationFields}{%
                            name={\glsxtrlongnoshortdescname},
                            sort={\glsxtrlonghyphennoshortdescsort},
                            first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
                            firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
                            text={\protect\glslonghyphenfont{\the\glslongtok}},%
                            plural={\protect\glslonghyphenfont{\the\glslongpltok}}%
                          }%
```

                       Unset the regular attribute if it has been set.

```
                          \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                            \glshasattribute{\the\glslabeltok}{regular}%
                            {%
                              \glssetattribute{\the\glslabeltok}{regular}{false}%
                            }%
                            {}%
                          }%
                        }%
                        {%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
```

The inline full form displays the long format followed by the short form in parentheses (as long-hyphen-short-hyphen).

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \glsaccesslong{##1}%
      }%
      {%
        \glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \glsaccessshort{##1}%
      }%
      {%
        \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \glsaccesslongpl{##1}%
      }%
      {%
        \glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \glsaccessshortpl{##1}%
      }%
      {%
        \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
```

```
    }%
    {##2}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \Glsaccesslong{##1}%
      }%
      {%
        \Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \glsaccessshort{##1}%
      }%
      {%
        \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \Glsaccesslongpl{##1}%
      }%
      {%
        \Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \glsaccessshortpl{##1}%
      }%
      {%
        \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlonghyphenshort{##1}%
    {%
```

```
      \glsifattribute{##1}{markwords}{true}%
      {%
        \GLSaccesslong{##1}%
      }%
      {%
        \GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \GLSaccessshort{##1}%
      }%
      {%
        \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlonghyphenshort{##1}%
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \GLSaccesslongpl{##1}%
      }%
      {%
        \GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \GLSaccessshortpl{##1}%
      }%
      {%
        \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
```
The first use full form only displays the long form.
```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslong{##1}%
    }%
```

```
    {%
      \glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslongpl{##1}%
    }%
    {%
      \glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \Glsaccesslong{##1}%
    }%
    {%
      \Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \Glsaccesslongpl{##1}%
    }%
    {%
      \Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
```

```
        \GLSaccesslong{##1}%
      }%
      {%
        \GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \GLSaccesslongpl{##1}%
    }%
    {%
      \GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
```

The format for subsequent use (not used when the regular attribute is set).

```
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslong{##1}%
    }%
    {%
      \glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrlonghyphennoshort{##1}%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslongpl{##1}%
    }%
    {%
      \glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
```

```
      \glsxtrlonghyphennoshort{##1}%
      {%
        \glsifattribute{##1}{markwords}{true}%
        {%
          \Glsaccesslong{##1}%
        }%
        {%
          \Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
      \glsxtrlonghyphennoshort{##1}%
      {%
        \glsifattribute{##1}{markwords}{true}%
        {%
          \Glsaccesslongpl{##1}%
        }%
        {%
          \Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\GLSxtrsubsequentfmt}[2]{%
      \GLSxtrlonghyphennoshort{##1}%
      {%
        \glsifattribute{##1}{markwords}{true}%
        {%
          \GLSaccesslong{##1}%
        }%
        {%
          \GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
      \GLSxtrlonghyphennoshort{##1}%
      {%
        \glsifattribute{##1}{markwords}{true}%
        {%
          \GLSaccesslongpl{##1}%
        }%
        {%
          \GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
```

```
      }%
    }
```

```
          \newcommand*{\glsxtrlonghyphennoshortsort}{\expandonce\glsxtrrorgshort}
```

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.

```
    \newabbreviationstyle{long-hyphen-noshort-noreg}%
    {%
```

Set accessibility attributes if enabled.

```
        \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
        \renewcommand*{\CustomAbbreviationFields}{%
          name={\glsxtrlongnoshortname},
          sort={\glsxtrlonghyphennoshortsort},
          first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
          firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
          text={\protect\glslonghyphenfont{\the\glslongtok}},%
          plural={\protect\glslonghyphenfont{\the\glslongpltok}},%
          description={\the\glslongtok}%
        }%
```

Unset the regular attribute if it has been set.

```
        \renewcommand*{\GlsXtrPostNewAbbreviation}{%
          \glshasattribute{\the\glslabeltok}{regular}%
          {%
            \glssetattribute{\the\glslabeltok}{regular}{false}%
          }%
          {}%
        }%
    }%
    {%

        \GlsXtrUseAbbrStyleFmts{long-hyphen-noshort-desc-noreg}%
    }
```

$\boxed{\text{\texttt{\textbackslash glsxtrlonghyphen\{}\langle\textit{long}\rangle\texttt{\}\{}\langle\textit{label}\rangle\texttt{\}\{}\langle\textit{insert}\rangle\texttt{\}}}}$

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.

The ⟨*long*⟩ argument will need to include the inner format.

```
    \newcommand*{\glsxtrlonghyphen}[3]{%
```

Grouping is needed to localise the redefinitions.

```
      {%
```

```
    \glsxtrifhyphenstart{#3}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \glsfirstlonghyphenfont{#1}%
  }%
}
```

<div style="border:1px solid; background:#fdfac0; padding:8px;">

\glsxtrposthyphenshort{⟨*label*⟩}{⟨*insert*⟩}

</div>

\glsxtrposthyphenshort

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.

```
\newcommand*{\glsxtrposthyphenshort}[2]{%
  {%
    \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \ifglsxtrinsertinside
      {\glsfirstlonghyphenfont{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen
    {%
      \glsxtrshortformat{#1}{#2}{\glsfirstabbrvhyphenfont}%
    }%
  }%
}
```

\GLSxtrposthyphenshort   As above but all caps.

```
\newcommand*{\GLSxtrposthyphenshort}[2]{%
  {%
    \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \ifglsxtrinsertinside
      {\glsfirstlonghyphenfont{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen
    {%
      \GLSxtrshortformat{#1}{#2}{\glsfirstabbrvhyphenfont}%
    }%
  }%
}
```

\glsxtrposthyphenshortpl   As above but plural.

```
\newcommand*{\glsxtrposthyphenshortpl}[2]{%
  {%
    \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \ifglsxtrinsertinside
```

```
      {\glsfirstlonghyphenfont{\glsxtrgenentrytextfmt{#2}}}%
    \else
      {\glsxtrgenentrytextfmt{#2}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen
    {%
      \glsxtrshortplformat{#1}{#2}{\glsfirstabbrvhyphenfont}%
    }%
  }%
}
```

\GLSxtrposthyphenshortpl As above but all caps.

```
\newcommand*{\GLSxtrposthyphenshortpl}[2]{%
 {%
    \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \ifglsxtrinsertinside
      {\glsfirstlonghyphenfont{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen
    {%
      \GLSxtrshortplformat{#1}{#2}{\glsfirstabbrvhyphenfont}%
    }%
  }%
}
```

\xpglsxtrposthyphenshort Expand placeholders and check for all caps.

```
\newcommand*{\xpglsxtrposthyphenshort}{%
  \glsxtrifallcaps
  {%
    \expandafter\GLSxtrposthyphenshort\expandafter\glslabel
    \expandafter{\glsinsert}%
  }%
  {%
    \expandafter\glsxtrposthyphenshort\expandafter\glslabel
    \expandafter{\glsinsert}%
  }%
}
```

\glsxtrposthyphensubsequent{⟨*label*⟩}{⟨*insert*⟩}

\glsxtrposthyphensubsequent

Format in the post-link hook for subsequent use. The label is ignored by default.
This just does the insert part with appropriate formatting.

```
\newcommand*{\glsxtrposthyphensubsequent}[2]{%
  \ifglsxtrinsertinside
```

```
        \glsabbrvfont{{\glsxtrgenentrytextfmt{#2}}}%
      \else
        {\glsxtrgenentrytextfmt{#2}}%
      \fi
    }
```

\GLSxtrposthyphensubsequent As above but all caps.

```
    \newcommand*{\GLSxtrposthyphensubsequent}[2]{%
      \ifglsxtrinsertinside
        \glsabbrvfont{{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
      \else
        {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
      \fi
    }
```

xpglsxtrposthyphensubsequent Expand placeholders and check for all caps.

```
    \newcommand*{\xpglsxtrposthyphensubsequent}{%
      \glsxtrifallcaps
      {%
        \expandafter\GLSxtrposthyphensubsequent\expandafter\glslabel
            \expandafter{\glsinsert}%
      }%
      {%
        \expandafter\glsxtrposthyphensubsequent\expandafter\glslabel
            \expandafter{\glsinsert}%
      }%
    }
```

> ### \glsxtrshorthyphennoinsert{⟨*label*⟩}{⟨*short*⟩}{⟨*insert*⟩}

\glsxtrshorthyphennoinsert

As with **\glsxtrshorthyphennolong** but doesn't actually show the insert.

```
    \newcommand*{\glsxtrshorthyphennoinsert}[3]{%
```

Grouping is needed to localise the redefinitions.

```
      {%
```

If ⟨*insert*⟩ starts with a hyphen, redefine **\glsxtrwordsep** to a hyphen.

```
        \glsxtrifhyphenstart{#3}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
        \glsfirstabbrvhyphenfont{#2}%
      }%
    }
```

long-hyphen-postshort-hyphen Like long-hyphen-short-hyphen but shifts the insert and parenthetical material to the post-link hook.

```
    \newabbreviationstyle{long-hyphen-postshort-hyphen}%
    {%
```

Set accessibility attributes if enabled.

```
      \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongshortname},
  sort={\glsxtrlonghyphenshortsort},
  first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
  firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
  text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
  description={\protect\glslonghyphenfont{\the\glslongtok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasfirstuse
    {%
      \xpglsxtrposthyphenshort
    }%
    {%
```

Put the insertion into the post-link:

```
      \xpglsxtrposthyphensubsequent
    }%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
  }%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
```

Subsequent use needs to omit the insertion but it needs to perform the space-hyphen substitution:

```
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
```

```
    {##2}%
  }%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshortpl{##1}%
    }%
    {%
      \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshort{##1}%
    }%
    {%
      \Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshortpl{##1}%
    }%
    {%
      \Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshort{##1}%
    }%
    {%
      \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
```

```
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
      \glsxtrshorthyphennoinsert{##1}%
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
          \GLSaccessshortpl{##1}%
        }%
        {%
          \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
First use full form:
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrlonghyphen
      {%
        \glsifattribute{##1}{markwords}{true}%
        {%
          \glsaccesslong{##1}%
        }%
        {%
          \glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##1}{##2}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrlonghyphen
      {%
        \glsifattribute{##1}{markwords}{true}%
        {%
          \glsaccesslongpl{##1}%
        }%
        {%
          \glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##1}{##2}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsxtrlonghyphen
      {%
        \glsifattribute{##1}{markwords}{true}%
        {%
```

```
        \Glsaccesslong{##1}%
      }%
      {%
        \Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##1}{##2}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsxtrlonghyphen
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \Glsaccesslongpl{##1}%
      }%
      {%
        \Glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##1}{##2}%
  }%
  \renewcommand*{\GLSxtrfullformat}[2]{%
    \glsxtrlonghyphen
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \GLSaccesslong{##1}%
      }%
      {%
        \GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##1}{##2}%
  }%
  \renewcommand*{\GLSxtrfullplformat}[2]{%
    \glsxtrlonghyphen
    {%
      \glsifattribute{##1}{markwords}{true}%
      {%
        \GLSaccesslongpl{##1}%
      }%
      {%
        \GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##1}{##2}%
  }%
```

In-line format.

```
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
```

576

```
      \glsxtrlongformatgrp{##1}{##2}{\glsfirstlonghyphenfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrlongplformatgrp{##1}{##2}{\glsfirstlonghyphenfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrlongformatgrp{##1}{##2}{\glsfirstlonghyphenfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrlongplformatgrp{##1}{##2}{\glsfirstlonghyphenfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrlongformatgrp{##1}{##2}{\glsfirstlonghyphenfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrlongplformatgrp{##1}{##2}{\glsfirstlonghyphenfont}%
    }%
  }
```

long-hyphen-postshort-hyphen-desc Like long-hyphen-postshort-hyphen but the description must be supplied by the user.

```
 \newabbreviationstyle{long-hyphen-postshort-hyphen-desc}%
 {%
```

Set accessibility attributes if enabled.

```
   \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
   \renewcommand*{\CustomAbbreviationFields}{%
     name={\glsxtrlongshortdescname},
     sort={\glsxtrlongshortdescsort},%
     first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
     firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
     text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
     plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
   }%
   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
     \csdef{glsxtrpostlink\glscategorylabel}{%
       \glsxtrifwasfirstuse
       {%
         \xpglsxtrposthyphenshort
       }%
       {%
```

Put the insertion into the post-link:

```
         \xpglsxtrposthyphensubsequent
       }%
     }%
     \glshasattribute{\the\glslabeltok}{regular}%
     {%
       \glssetattribute{\the\glslabeltok}{regular}{false}%
```

577

```
      }%
      {}%
    }%
  }%
  {%
    \GlsXtrUseAbbrStyleFmts{long-hyphen-postshort-hyphen}%
  }
```

> \glsxtrshorthyphenlong{⟨*label*⟩}{⟨*short*⟩}{⟨*long*⟩}{⟨*insert*⟩}

\glsxtrshorthyphenlong

The ⟨*long*⟩ and ⟨*short*⟩ arguments may be the plural form. The ⟨*long*⟩ argument may also be the first letter uppercase form.

As with `\glsxtrlonghyphenshort` this doesn't fit in with the new `\glsxtrshortformat` so the inserted part has to have a separate encapsulation for the inner format. The ⟨*long*⟩ argument will need to include the inner format.

```
  \newcommand*{\glsxtrshorthyphenlong}[4]{%
```

Grouping is needed to localise the redefinitions.

```
  {%
```

If ⟨*insert*⟩ starts with a hyphen, redefine `\glsxtrwordsep` to a hyphen. The inserted material is also inserted into the parenthetical part. (The inserted material is grouped as a precautionary measure.)

```
    \glsxtrifhyphenstart{#4}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \glsfirstabbrvhyphenfont{#2\ifglsxtrinsertinside
      {\glsxtrgenentrytextfmt{#4}}\fi}%
    \ifglsxtrinsertinside\else{\glsxtrgenentrytextfmt{#4}}\fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\glsfirstlonghyphenfont{#3%
      \ifglsxtrinsertinside{\glsxtrgenentrytextfmt{#4}}\fi}%
     \ifglsxtrinsertinside\else{\glsxtrgenentrytextfmt{#4}}\fi}%
  }%
  }
```

\GLSxtrshorthyphenlong As above but convert insert to all-caps. The long and short form arguments should be provided as all-caps.

```
  \newcommand*{\GLSxtrshorthyphenlong}[4]{%
  {%
    \glsxtrifhyphenstart{#4}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
    \glsfirstabbrvhyphenfont{#2\ifglsxtrinsertinside
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}\fi}%
    \ifglsxtrinsertinside\else
      {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}%
    \fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\glsfirstlonghyphenfont{#3%
      \ifglsxtrinsertinside{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}\fi}%
     \ifglsxtrinsertinside\else
```

```
                          {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#4}}}%
                        \fi}%
                    }%
                    }
```

\glsxtrshorthyphenlongsort

```
                    \newcommand*{\glsxtrshorthyphenlongsort}{\expandonce\glsxtrorgshort}
```

short-hyphen-long-hyphen Designed for use with the markwords attribute.

```
                    \newabbreviationstyle{short-hyphen-long-hyphen}%
                    {%
```

Set accessibility attributes if enabled.

```
                        \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
                    \renewcommand*{\CustomAbbreviationFields}{%
                        name={\glsxtrshortlongname},
                        sort={\glsxtrshorthyphenlongsort},
                        first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
                        \protect\glsxtrfullsep{\the\glslabeltok}%
                        \protect\glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
                        firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
                        \protect\glsxtrfullsep{\the\glslabeltok}%
                        \protect\glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%
                        text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
                        plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
                        description={\protect\glslonghyphenfont{\the\glslongtok}}}%
```

Unset the regular attribute if it has been set.

```
                    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
                        \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
                        \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
                        \glshasattribute{\the\glslabeltok}{regular}%
                        {%
                            \glssetattribute{\the\glslabeltok}{regular}{false}%
                        }%
                        {}%
                    }%
                    }%
                    {%
                        \renewcommand*{\abbrvpluralsuffix}{\glsxtrhyphensuffix}%
                        \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
                        \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
                        \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
                        \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
                    \renewcommand*{\glsxtrfullformat}[2]{%
                        \glsxtrshorthyphenlong{##1}%
                        {%
```

```
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslong{##1}%
    }%
    {%
      \glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrshorthyphenlong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshortpl{##1}%
    }%
    {%
      \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{marklongwords}{true}%
    {%
      \glsaccesslongpl{##1}%
    }%
    {%
      \glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsxtrshorthyphenlong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshort{##1}%
    }%
    {%
      \Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
```

580

```
    }%
  }%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslong{##1}%
    }%
    {%
      \glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsxtrshorthyphenlong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshortpl{##1}%
    }%
    {%
      \Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \glsaccesslongpl{##1}%
    }%
    {%
      \glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshorthyphenlong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshort{##1}%
    }%
    {%
      \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \GLSaccesslong{##1}%
```

```
    }%
    {%
      \GLSaccessfmtlong{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshorthyphenlong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshortpl{##1}%
    }%
    {%
      \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \GLSaccesslongpl{##1}%
    }%
    {%
      \GLSaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
```
Subsequent form also needs checking for a hyphen in case the short form has spaces.
```
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsxtrshorthyphennolong{##1}%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \glsaccessshort{##1}%
      }%
      {%
        \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glsxtrshorthyphennolong{##1}%
    {%
      \glsifattribute{##1}{markshortwords}{true}%
      {%
```

```
        \glsaccessshortpl{##1}%
      }%
      {%
        \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennolong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshort{##1}%
    }%
    {%
      \Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glsxtrshorthyphennolong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshortpl{##1}%
    }%
    {%
      \Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \GLSxtrshorthyphennolong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshort{##1}%
    }%
    {%
      \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
  \GLSxtrshorthyphennolong{##1}%
  {%
```

```
      \glsifattribute{##1}{markshortwords}{true}%
      {%
        \GLSaccessshortpl{##1}%
      }%
      {%
        \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
      }%
    }%
    {##2}%
  }%
}
```

short-hyphen-long-hyphen-desc Like short-hyphen-long-hyphen but the description must be supplied by the user.

```
\newabbreviationstyle{short-hyphen-long-hyphen-desc}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},
    sort={\glsxtrshortlongdescsort},
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \protect\glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
  }%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-hyphen-long-hyphen}%
}
```

\glsxtrshorthyphen{⟨*short*⟩}{⟨*label*⟩}{⟨*insert*⟩}

\glsxtrshorthyphen

Used by short-hyphen-postlong-hyphen. The ⟨*insert*⟩ is checked to determine if it starts with a hyphen but isn't used here as it's moved to the post-link hook.

```
\newcommand*{\glsxtrshorthyphen}[3]{%
```

Grouping is needed to localise the redefinitions.

```
 {%
   \glsxtrifhyphenstart{#3}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
   \glsfirstabbrvhyphenfont{#1}%
 }%
 }
```

<div style="border:1px solid; background:#ffffcc; padding:8px;">

`\glsxtrposthyphenlong{`⟨*label*⟩`}{`⟨*insert*⟩`}`

</div>

`\glsxtrposthyphenlong`

Used in the post-link hook for the short-hyphen-postlong-hyphen style. Much like `\glsxtrshorthyphenlong` but omits the ⟨*short*⟩ part. This always uses the singular long form.

```
\newcommand*{\glsxtrposthyphenlong}[2]{%
 {%
   \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
   \ifglsxtrinsertinside
    {\glsfirstabbrvhyphenfont{\glsxtrgenentrytextfmt{#2}}}%
   \else
    {\glsxtrgenentrytextfmt{#2}}%
   \fi
   \glsxtrfullsep{#1}%
   \glsxtrparen
     {\glsxtrlongformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
 }%
 }
```

`\GLSxtrposthyphenlong`   As above but all-caps.

```
\newcommand*{\GLSxtrposthyphenlong}[2]{%
 {%
   \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
   \ifglsxtrinsertinside
    {\glsfirstabbrvhyphenfont{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
   \else
    {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
   \fi
   \glsxtrfullsep{#1}%
   \glsxtrparen
     {\GLSxtrlongformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
 }%
 }
```

Plural versions in case they are required.

`\glsxtrposthyphenlongpl`

```
\newcommand*{\glsxtrposthyphenlongpl}[2]{%
 {%
   \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
   \ifglsxtrinsertinside
    {\glsfirstabbrvhyphenfont{\glsxtrgenentrytextfmt{#2}}}%
   \else
    {\glsxtrgenentrytextfmt{#2}}%
   \fi
   \glsxtrfullsep{#1}%
   \glsxtrparen
     {\glsxtrlongplformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
 }%
}
```

\GLSxtrposthyphenlongpl As above but all-caps.

```
\newcommand*{\GLSxtrposthyphenlongpl}[2]{%
 {%
   \glsxtrifhyphenstart{#2}{\let\glsxtrwordsep\glsxtrwordsephyphen}{}%
   \ifglsxtrinsertinside
    {\glsfirstabbrvhyphenfont{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}}%
   \else
    {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
   \fi
   \glsxtrfullsep{#1}%
   \glsxtrparen
     {\GLSxtrlongplformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
 }%
}
```

\xpglsxtrposthyphenlong Expand placeholders and check for all caps.

```
\newcommand*{\xpglsxtrposthyphenlong}{%
  \glsxtrifallcaps
  {%
    \expandafter\GLSxtrposthyphenlong\expandafter\glslabel
        \expandafter{\glsinsert}%
  }%
  {%
    \expandafter\glsxtrposthyphenlong\expandafter\glslabel
        \expandafter{\glsinsert}%
  }%
}
```

short-hyphen-postlong-hyphen Like short-hyphen-long-hyphen but shifts the insert and parenthetical material to the post-link hook.

```
\newabbreviationstyle{short-hyphen-postlong-hyphen}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

586

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortlongname},
  sort={\glsxtrshorthyphenlongsort},
  first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}},%
  firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}},%
  text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
  description={\protect\glslonghyphenfont{\the\glslongtok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasfirstuse
    {%
      \xpglsxtrposthyphenlong
    }%
    {%
```

Put the insertion into the post-link:

```
      \xpglsxtrposthyphensubsequent
    }%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
 }%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
```

Subsequent use needs to omit the insertion but it needs to perform the space-hyphen substitution:

```
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
```

```
    {##2}%
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshortpl{##1}%
    }%
    {%
      \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshort{##1}%
    }%
    {%
      \Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshortpl{##1}%
    }%
    {%
      \Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \glsxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshort{##1}%
    }%
    {%
      \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
```

```
        }%
      }%
      {##2}%
    }%
    \renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
      \glsxtrshorthyphennoinsert{##1}%
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
          \GLSaccessshortpl{##1}%
        }%
        {%
          \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##2}%
    }%
First use full form:
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrshorthyphen
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
          \glsaccessshort{##1}%
        }%
        {%
          \glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##1}{##2}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrshorthyphen
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
          \glsaccessshortpl{##1}%
        }%
        {%
          \glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
        }%
      }%
      {##1}{##2}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsxtrshorthyphen
      {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
```

589

```
      \Glsaccessshort{##1}%
    }%
    {%
      \Glsaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##1}{##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsxtrshorthyphen
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshortpl{##1}%
    }%
    {%
      \Glsaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }{##1}{##2}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \glsxtrshorthyphen
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshort{##1}%
    }%
    {%
      \GLSaccessfmtshort{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##1}{##2}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \glsxtrshorthyphen
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshortpl{##1}%
    }%
    {%
      \GLSaccessfmtshortpl{}{\glsxtrgenentrytextfmt}{##1}%
    }%
  }{##1}{##2}%
}%
```

In-line format. Commands like `\glsxtrfull` set `\glsinsert` to empty.
The entire link-text (provided by the following commands) is stored in
`\glscustomtext`. Note that unless the insert is saved, it won't appear in the
post-link hook.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrshortformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrshortplformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrshortformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrshortplformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrshortformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrshortplformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
    }%
  }
```

short-hyphen-postlong-hyphen-desc Like short-hyphen-postlong-hyphen but the description must be supplied by the user.

```
 \newabbreviationstyle{short-hyphen-postlong-hyphen-desc}%
 {%
```

Set accessibility attributes if enabled.

```
   \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
   \renewcommand*{\CustomAbbreviationFields}{%
     name={\glsxtrshortlongdescname},
     sort={\glsxtrshortlongdescsort},%
     first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}},%
     firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}},%
     text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
     plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
   }%
   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
     \csdef{glsxtrpostlink\glscategorylabel}{%
       \glsxtrifwasfirstuse
       {%
         \xpglsxtrposthyphenlong
       }%
       {%
```

Put the insertion into the post-link:

```
         \xpglsxtrposthyphensubsequent
       }%
     }%
     \glshasattribute{\the\glslabeltok}{regular}%
     {%
```

591

```
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
    \GlsXtrUseAbbrStyleFmts{short-hyphen-postlong-hyphen}%
  }
```

2.7 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.

\glsabbrvonlyfont

```
\newcommand*{\glsabbrvonlyfont}{\glsabbrvdefaultfont}%
```

\glsfirstabbrvonlyfont

```
\newcommand*{\glsfirstabbrvonlyfont}{\glsabbrvonlyfont}%
```

\glslongonlyfont

```
\newcommand*{\glslongonlyfont}{\glslongdefaultfont}%
```

\glsfirstlongonlyfont

```
\newcommand*{\glsfirstlongonlyfont}{\glslongonlyfont}%
```

The default short form suffix:

\glsxtronlysuffix

```
\newcommand*{\glsxtronlysuffix}{\glsxtrabbrvpluralsuffix}
```

\glsxtronlyname The default name format for this style.

```
\newcommand*{\glsxtronlyname}{%
  \protect\glsabbrvonlyfont{\the\glsshorttok}%
}
```

long-only-short-only

```
\newabbreviationstyle{long-only-short-only}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtronlyname},
    sort={\the\glsshorttok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},%
    firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},%
    text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
    plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
    description={\protect\glslongonlyfont{\the\glslongtok}}}%
```

Unset the regular attribute if it has been set.

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%

    \renewcommand*{\abbrvpluralsuffix}{\glsxtronlysuffix}%
    \renewcommand*{\glsabbrvfont}[1]{\glsabbrvonlyfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvonlyfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongonlyfont{##1}}%
```

The first use full form doesn't show the short form.

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsxtrlongformat{##1}{##2}{\glsfirstlongonlyfont}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsxtrlongplformat{##1}{##2}{\glsfirstlongonlyfont}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \Glsxtrlongformat{##1}{##2}{\glsfirstlongonlyfont}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \Glsxtrlongplformat{##1}{##2}{\glsfirstlongonlyfont}%
    }%
    \renewcommand*{\GLSxtrfullformat}[2]{%
      \GLSxtrlongformat{##1}{##2}{\glsfirstlongonlyfont}%
    }%
    \renewcommand*{\GLSxtrfullplformat}[2]{%
      \GLSxtrlongplformat{##1}{##2}{\glsfirstlongonlyfont}%
    }%
```

The inline full form does show the short form.

```
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongonlyfont}{\glsfirstabbrvonlyfont}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongonlyfont}{\glsfirstabbrvonlyfont}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \Glsxtrlongshortformat{##1}{##2}%
        {\glsfirstlongonlyfont}{\glsfirstabbrvonlyfont}%
```

```
      }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \Glsxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongonlyfont}{\glsfirstabbrvonlyfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullformat}[2]{%
      \GLSxtrlongshortformat{##1}{##2}%
        {\glsfirstlongonlyfont}{\glsfirstabbrvonlyfont}%
    }%
    \renewcommand*{\GLSxtrinlinefullplformat}[2]{%
      \GLSxtrlongshortplformat{##1}{##2}%
        {\glsfirstlongonlyfont}{\glsfirstabbrvonlyfont}%
    }%
  }
```

\glsxtronlydescsort

```
    \newcommand*{\glsxtronlydescsort}{\the\glslongtok}
```

\glsxtronlydescname

```
    \newcommand*{\glsxtronlydescname}{%
      \protect\glslongfont{\the\glslongtok}%
    }
```

long-only-short-only-desc

```
    \newabbreviationstyle{long-only-short-only-desc}%
    {%
```
Set accessibility attributes if enabled.
```
      \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```
Setup the default fields.
```
      \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtronlydescname},
        sort={\glsxtronlydescsort},%
        first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},%
        firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},%
        text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
        plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
      }%
```
Unset the regular attribute if it has been set.
```
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
    }%
    {%
      \GlsXtrUseAbbrStyleFmts{long-only-short-only}%
```

594

```
}
```

Small-caps is awkward, so support for that is added.

\glsabbrvsconlyfont

```
\newcommand*{\glsabbrvsconlyfont}{\glsabbrvscfont}%
```

\glsfirstabbrvsconlyfont

```
\newcommand*{\glsfirstabbrvsconlyfont}{\glsabbrvsconlyfont}%
```

The default short form suffix:

\glsxtrsconlysuffix

```
\newcommand*{\glsxtrsconlysuffix}{\glsxtrscsuffix}
```

\glsxtrsconlyrevert

```
\newcommand*{\glsxtrsconlyrevert}{\glsxtrscrevert}
```

\glsxtrsconlyname   The default name format for this style.

```
\newcommand*{\glsxtrsconlyname}{%
  \protect\glsabbrvsconlyfont{\the\glsshorttok}%
}
```

long-only-short-sc-only

```
\newabbreviationstyle{long-only-short-sc-only}%
{%
```

Set accessibility attributes if enabled.

```
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrsconlyname},
    sort={\the\glsshorttok},
    first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},%
    firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},%
    text={\protect\glsabbrvsconlyfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsconlyfont{\the\glsshortpltok}},%
    description={\protect\glslongonlyfont{\the\glslongtok}}}%
```

Unset the regular attribute if it has been set.

```
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
```

```
\renewcommand*{\abbrvpluralsuffix}{\glsxtrsconlysuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvsconlyfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvsconlyfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongonlyfont{##1}}%
\renewcommand*\glsxtrrevert[1]{\glsxtrsconlyrevert{##1}}%
```

The first use full form doesn't show the short form.

```
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlongformat{##1}{##2}{\glsfirstlongonlyfont}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlongplformat{##1}{##2}{\glsfirstlongonlyfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrlongformat{##1}{##2}{\glsfirstlongonlyfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrlongplformat{##1}{##2}{\glsfirstlongonlyfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrlongformat{##1}{##2}{\glsfirstlongonlyfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrlongplformat{##1}{##2}{\glsfirstlongonlyfont}%
}%
```

The inline full form does show the short form.

```
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongonlyfont}{\glsfirstabbrvsconlyfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongonlyfont}{\glsfirstabbrvsconlyfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
    {\glsfirstlongonlyfont}{\glsfirstabbrvsconlyfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongonlyfont}{\glsfirstabbrvsconlyfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
    {\glsfirstlongonlyfont}{\glsfirstabbrvsconlyfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlongshortplformat{##1}{##2}%
    {\glsfirstlongonlyfont}{\glsfirstabbrvsconlyfont}%
```

```
      }%
    }
```

\glsxtrsconlydescsort

```
\newcommand*{\glsxtrsconlydescsort}{\glsxtronlydescsort}
```

\glsxtrsconlydescname

```
\newcommand*{\glsxtrsconlydescname}{\glsxtronlydescname}
```

long-only-short-sc-only-desc

```
\newabbreviationstyle{long-only-short-sc-only-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrsconlydescname},
  sort={\glsxtrsconlydescsort},%
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},%
  text={\glsxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-only-short-sc-only}%
}
```

3   Commands Specific to bib2gls (glossaries-extra-bib2gls.sty)

This package provides additional support for bib2gls and is automatically loaded by the record option.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v1.48}{2021-11-22}{glossaries-extra-bib2gls-2021-11-22.sty}
\DeclareCurrentRelease{v1.7}{2025-05-14}
```

Declare package:

```
\ProvidesPackage{glossaries-extra-bib2gls}[2025/05/14 v1.7 (NLCT)]
```

Provide convenient shortcut commands for predefined glossary types.

\printunsrtacronyms

```
\ifglsacronym
  \providecommand*{\printunsrtacronyms}[1][]{%
    \printunsrtglossary[type=\acronymtype,#1]}%
\fi
```

\printunsrtindex

```
\ifglossaryexists{index}
{
  \providecommand*{\printunsrtindex}[1][]{%
  \printunsrtglossary[type=index,#1]}%
}{}
```

\printunsrtsymbols

```
\ifglossaryexists{symbols}
{
  \providecommand*{\printunsrtsymbols}[1][]{%
  \printunsrtglossary[type=symbols,#1]}%
}{}
```

\printunsrtnumbers

```
\ifglossaryexists{numbers}
{
  \providecommand*{\printunsrtnumbers}[1][]{%
  \printunsrtglossary[type=numbers,#1]}%
}{}
```

\printunsrtabbreviations

```
\ifglossaryexists{abbreviations}
{
  \providecommand*{\printunsrtabbreviations}[1][]{%
  \printunsrtglossary[type=abbreviations,#1]}%
}{}
```

\glsdisplaynumberlist Allow \glsdisplaynumberlist and make it robust.

```
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    {\let\bibglsdelimN\glsnumlistsep
     \let\bibglslastDelimN\glsnumlistlastsep
     \glsxtrusefield{#1}{location}%
    }%
  }%
}
\robustify\glsdisplaynumberlist
```

\glsentrynumberlist

```
\renewcommand*{\glsentrynumberlist}[1]{\glsxtrusefield{#1}{location}}
```

\IfTeXParserLib This is defined by the TeX parser library to behave like `\@firstoftwo`. May be used to provide different code in fields that may be interpreted.

```
\providecommand{\IfTeXParserLib}[2]{#2}
```

The next command is similar but is specifically for `bib2gls` and won't in general be recognised by the TeX parser library if used by other applications (such as the converter tools provided with `bib2gls`).

\IfNotBibGls This is defined by the `bib2gls` interpreter to behave like `\@secondoftwo`.

```
\providecommand{\IfNotBibGls}[2]{#1}
```

These are some convenient macros for use with custom rules.

\glshex

```
\newcommand*{\glshex}{\string\u}
```

\glscapturedgroup

```
\newcommand*{\glscapturedgroup}{\string\$}
```

\glshashchar Expands to a literal hash character (similar to `\glsbackslash`)

```
\ifdef\glshashchar
{}
{\edef\glshashchar{\expandafter\@gobble\string\#}}
```

XtrResourceInitEscSequences Protect commands that shouldn't expand in resource options as they have a special meaning in the context of those options. This command may be added to the definition of `\glsxtrresourceinit`.

```
\newcommand*{\GlsXtrResourceInitEscSequences}{%
\def\u{\string\u}%
\def\.{\string\.}%
\def\\{\string\\}%
\def\/{\string\/}%
\def\|{\string\|}%
\def\&{\string\&}%
\def\+{\string\+}%
\def\<{\string\<}%
\def\>{\string\>}%
\def\*{\string\*}%
\def\${\string\$}%
\def\^{\string\^}%
\def\~{\string\~}%
\def\({\string\(}%
\def\){\string\)}%
\def\[{\string\[}%
\def\]{\string\]}%
\def\"{\string\"}%
```

```
                    \def\-{\string\-}%
                    \def\?{\string\?}%
                    \def\#{\string\#}%
                    \def\:{\string\:}%
                    \def\cs##1{\glsbackslash##1}%
                    \def\CS{\string\CS}%
                    \def\NULL{\string\NULL\space}%
                    \def\IN{\string\IN\space}%
                    \def\NIN{\string\NIN\space}%
                    \def\PREFIXOF{\string\PREFIXOF\space}%
                    \def\NOTPREFIXOF{\string\NOTPREFIXOF\space}%
                    \def\SUFFIXOF{\string\SUFFIXOF\space}%
                    \def\NOTSUFFIXOF{\string\NOTSUFFIXOF\space}%
                    \def\LC{\string\LC}%
                    \def\UC{\string\UC}%
                    \def\FIRSTLC{\string\FIRSTLC}%
                    \def\FIRSTUC{\string\FIRSTUC}%
                    \def\TITLE{\string\TITLE}%
                    \def\MGP{\string\MGP}%
                    \def\LEN{\string\LEN}%
                    \def\TRIM{\string\TRIM}%
                    \def\INTERPRET{\string\INTERPRET}%
                    \def\LABELIFY{\string\LABELIFY}%
                    \def\LABELIFYLIST{\string\LABELIFYLIST}%
                    \def\CAT{\string\CAT}%
                  }
```

GlsXtrIfHasNonZeroChildCount For use with `bib2gls`'s `save-child-count` resource option.

```
                  \newcommand*{\GlsXtrIfHasNonZeroChildCount}{%
                    \@ifstar\s@GlsXtrIfHasNonZeroChildCount\@GlsXtrIfHasNonZeroChildCount
                  }
```

GlsXtrIfHasNonZeroChildCount

```
                  \newcommand*{\@GlsXtrIfHasNonZeroChildCount}[3]{%
                    \@GlsXtrIfFieldNonZero{childcount}{#1}{#2}{#3}%
                  }
```

GlsXtrIfHasNonZeroChildCount

```
                  \newcommand*{\s@GlsXtrIfHasNonZeroChildCount}[3]{%
                    \s@GlsXtrIfFieldNonZero{childcount}{#1}{#2}{#3}%
                  }
```

\glsxtrprovidecommand For use in `@preamble`, this behaves like `\providecommand` in the document but
                  like `\renewcommand` in bib2gls.

```
                  \newcommand*{\glsxtrprovidecommand}{\providecommand}
```

\glsrenewcommand Like `\renewcommand` but only generates a warning rather than an error if the
                  command isn't defined.

```
                  \newcommand*{\glsrenewcommand}{\@star@or@long\glsxtr@renewcommand}
```

`\glsxtr@renewcommand`

```
\newcommand*{\glsxtr@renewcommand}[1]{%
 \begingroup \escapechar\m@ne\xdef\@gtempa{{\string#1}}\endgroup
 \expandafter\@ifundefined\@gtempa
   {%
     \GlossariesExtraWarning{can't redefine \noexpand#1(not already defined)}%
   }%
   \relax
  \relax
 \let\@ifdefinable\@rc@ifdefinable
 \new@command#1%
}
```

`\glsxtr@wrglossarylocation`

$\boxed{\texttt{\textbackslash glsxtr@wrglossarylocation\{}\langle \textit{wr-loc} \rangle\texttt{\}\{}\langle \textit{page} \rangle\texttt{\}}}$

For use with indexcounter and bib2gls. This just expands to ⟨*wr-loc*⟩ to allow `\glsnoidxdisplayloc` to obtain the hyperlink target. The page number obtained when bib2gls parses the aux file.

```
\newcommand*{\glsxtr@wrglossarylocation}[2]{#1}
```

`\GlsXtrIndexCounterLink`

$\boxed{\texttt{\textbackslash GlsXtrIndexCounterLink\{}\langle \textit{text} \rangle\texttt{\}\{}\langle \textit{label} \rangle\texttt{\}}}$

For use with indexcounter and bib2gls.

```
\ifdef\hyperref
{%
  \newcommand*{\GlsXtrIndexCounterLink}[2]{%
    \glsxtrifhasfield{indexcounter}{#2}%
    {\hyperref[wrglossary.\glscurrentfieldvalue]{#1}}%
    {#1}%
  }
}
{
  \newcommand*{\GlsXtrIndexCounterLink}[2]{#1}
}
```

`\GlsXtrDualField`

$\boxed{\texttt{\textbackslash GlsXtrDualField}}$

The internal field used to store the dual label. The dual-field defaults to dual if no value is supplied so that's used as the default.

```
\newcommand*{\GlsXtrDualField}{dual}
```

`\GlsXtrDualBackLink`

$\boxed{\texttt{\textbackslash GlsXtrDualBackLink\{}\langle \textit{text} \rangle\texttt{\}\{}\langle \textit{label} \rangle\texttt{\}}}$

Adds a hyperlink to the dual entry.

```
\newcommand*{\GlsXtrDualBackLink}[2]{%
  \glsxtrifhasfield{\GlsXtrDualField}{#2}%
  {\glshyperlink[#1]{\glscurrentfieldvalue}}%
  {#1}%
}
```

`\GlsXtrBibTeXEntryAliases` Convenient shortcut for use with `entry-type-aliases` to alias standard BIBTEX entry types to `@bibtexentry`.

```
\newcommand*{\GlsXtrBibTeXEntryAliases}{%
  article=bibtexentry,
  book=bibtexentry,
  booklet=bibtexentry,
  conference=bibtexentry,
  inbook=bibtexentry,
  incollection=bibtexentry,
  inproceedings=bibtexentry,
  manual=bibtexentry,
  mastersthesis=bibtexentry,
  misc=bibtexentry,
  phdthesis=bibtexentry,
  proceedings=bibtexentry,
  techreport=bibtexentry,
  unpublished=bibtexentry
}
```

`\GlsXtrProvideBibTeXFields` Convenient shortcut to define the standard BIBTEX fields.

```
\newcommand*{\GlsXtrProvideBibTeXFields}{%
  \glsaddstoragekey{address}{}{\glsxtrbibaddress}%
  \glsaddstoragekey{author}{}{\glsxtrbibauthor}%
  \glsaddstoragekey{booktitle}{}{\glsxtrbibbooktitle}%
  \glsaddstoragekey{chapter}{}{\glsxtrbibchapter}%
  \glsaddstoragekey{edition}{}{\glsxtrbibedition}%
  \glsaddstoragekey{howpublished}{}{\glsxtrbibhowpublished}%
  \glsaddstoragekey{institution}{}{\glsxtrbibinstitution}%
  \glsaddstoragekey{journal}{}{\glsxtrbibjournal}%
  \glsaddstoragekey{month}{}{\glsxtrbibmonth}%
  \glsaddstoragekey{note}{}{\glsxtrbibnote}%
  \glsaddstoragekey{number}{}{\glsxtrbibnumber}%
  \glsaddstoragekey{organization}{}{\glsxtrbiborganization}%
  \glsaddstoragekey{pages}{}{\glsxtrbibpages}%
  \glsaddstoragekey{publisher}{}{\glsxtrbibpublisher}%
  \glsaddstoragekey{school}{}{\glsxtrbibschool}%
  \glsaddstoragekey{series}{}{\glsxtrbibseries}%
  \glsaddstoragekey{title}{}{\glsxtrbibtitle}%
  \glsaddstoragekey{bibtextype}{}{\glsxtrbibtype}%
  \glsaddstoragekey{volume}{}{\glsxtrbibvolume}%
}
```

Multiple supplementary references are only supported with `bib2gls`.

`\glsxtrmultisupplocation` This is like `\glsxtrsupphypernumber` but the second argument is the external file name (which isn't obtained from the externallocation attribute). The third argument is the formatting (encap) control sequence *name*. This is ignored by default, but is set by `bib2gls` to the original encap in case it's required.

```
\newcommand*{\glsxtrmultisupplocation}[3]{%
 {%
   \def\glsxtrsupplocationurl{#2}%
   \glshypernumber{#1}%
 }%
}
```

> `\glsxtrdisplaysupploc{⟨prefix⟩}{⟨counter⟩}{⟨format⟩}{⟨src⟩}`
> `{⟨location⟩}`

`\glsxtrdisplaysupploc`

This is like `\glsnoidxdisplayloc` but is used for supplementary locations and so requires an extra argument.

```
\newcommand*\glsxtrdisplaysupploc[5]{%
  \setentrycounter[#1]{#2}%
  \glsxtrmultisupplocation{#5}{#4}{#3}%
}
```

`\glsxtr@setlocationanchor`

```
\ExplSyntaxOn
\cs_new:Npn \glsxtr@setlocationanchor #1 #2
 {
    \group_begin:
      \glswrglossdisableanchorcmds
      \exp_args:NNNe
    \group_end:
    \tl_set:Nn #1 { \text_purify:n { #2 } }
 }
\ExplSyntaxOff
```

`\glsxtrdisplaylocnameref` `\glsxtrdisplaylocnameref{⟨prefix⟩}{⟨counter⟩}{⟨format⟩}{⟨location⟩}{⟨name⟩}` `{⟨href⟩}{⟨hcounter⟩}{⟨external file⟩}` Used with the [nameref]record package option. The ⟨*href*⟩ argument was obtained from `\@currentHref` and the ⟨*hcounter*⟩ argument was obtained from `\theHentrycounter`, which is more reliable. If hyperref hasn't been loaded, this just behaves like `\glsnoidxdisplayloc`.

```
\ifundef\hyperlink
{
  \newcommand*{\glsxtrdisplaylocnameref}[8]{%
    \glsnoidxdisplayloc{#1}{#2}{#3}{#4}%
  }
}
{
```

Default action uses ⟨*hcounter*⟩. Equations and pages typically don't have a title, so check the counter name (otherwise the title may be the section or

chapter title, which can be confusing). As from v1.42, this now checks if the control sequence \glsxtr⟨*counter*⟩locfmt is defined. The prefix argument is redundant.

```
\newcommand*{\glsxtrdisplaylocnameref}[8]{%
  \def\glsxtrrecentanchor{#6}%
  \glsxtr@setlocationanchor\glsxtrlocationanchor{#2.#7}%
```

Initialise \glsxtractualanchor:

```
\let\glsxtractualanchor\glsxtrlocationanchor
\glsxtrsetactualanchor{#2}%
\ifcsdef{glsxtr#2locfmt}%
{\glsxtrnamereflink{#3}{\csuse{glsxtr#2locfmt}{#4}{#5}}{\glsxtractualanchor}{#8}}%
{%
  \ifstrempty{#5}%
  {%
```

No title, so just use the location as the link text.

```
    \glsxtrnamereflink{#3}{#4}{\glsxtractualanchor}{#8}%
  }%
  {%
    \ifstrequal{#2}{page}%
    {\glsxtrnamereflink{#3}{#4}{\glsxtractualanchor}{#8}}%
    {\glsxtrtitlednamereflink{#3}{#4}{#5}{#8}}%
  }%
}%
}
}
```

\glsxtractualanchor Does nothing by default. May be redefined to override the default.

```
\newcommand{\glsxtrsetactualanchor}[1]{}
```

> \glsxtrtitlednamereflink{⟨*format*⟩}{⟨*location*⟩}{⟨*title*⟩}
> {⟨*file*⟩}

\glsxtrtitlednamereflink

```
\newcommand{\glsxtrtitlednamereflink}[4]{%
  \glsxtrnamereflink{#1}{#2}{\glsxtrrecentanchor}{#4}%
}
```

> \glsxtrequationlocfmt{⟨*location*⟩}{⟨*title*⟩}

\glsxtrequationlocfmt

```
\newcommand*{\glsxtrequationlocfmt}[2]{(#1)}
```

> \glsxtrwrglossarylocfmt{⟨*location*⟩}{⟨*title*⟩}

\glsxtrwrglossarylocfmt

```
\newcommand*{\glsxtrwrglossarylocfmt}[2]{%
  {\@@glsxtrwrglosscountermark{#1}%
  \let\glsxtr@wrglossarylocation\@secondoftwo
  #1}%
}
```

\glsxtrnamereflink

> \glsxtrnamereflink{⟨*format*⟩}{⟨*title*⟩}{⟨*href*⟩}{⟨*external file*⟩}

```
\newcommand*{\glsxtrnamereflink}[4]{%
```

Locally change \glshypernumber to \@firstofone to remove the normal location hyperlink.

```
\begingroup
  \let\glshypernumber\@firstofone
```

If the ⟨*external file*⟩ argument is empty, an internal link is used, otherwise an external one is needed.

```
\ifstrempty{#4}%
{\glsxtrfmtinternalnameref{#3}{#1}{#2}}%
{\glsxtrfmtexternalnameref{#3}{#1}{#2}{#4}}%
\endgroup
}
```

\glsxtrnameloclink

> \glsxtrnameloclink{⟨*prefix*⟩}{⟨*counter*⟩}{⟨*format*⟩}
> {⟨*location*⟩}{⟨*text*⟩}{⟨*external file*⟩}

Like \@gls@numberlink, this creates a hyperlink to the target obtained from the prefix, counter and location but uses ⟨*text*⟩ as the hyperlink text. As with regular indexing, this will fail if the target name can't be formed by prefixing the location value.

```
\newcommand{\glsxtrnameloclink}[6]{%
\begingroup
\setentrycounter[#1]{#2}%
\def\glsxtr@locationhypertext{#5}%
\let\glshypernumber\@firstofone
\def\@glsnumberformat{#3}%
\def\glsxtrsupplocationurl{#6}%
\toks@={}%
\@@glsxtr@bibgls@removespaces#4 \@nil
\endgroup
}
```

\@glsxtr@bibgls@removespaces

```
\def\@glsxtr@bibgls@removespaces#1 #2\@nil{%
```

```
        \toks@=\expandafter{\the\toks@#1}%
      \ifx\\#2\\%

        \edef\@glo@tmp{\the\toks@}%
        \ifx\@glo@tmp\empty
        \else
          \protected@edef\@glo@tmp{\glsentrycounter\@glo@counterprefix\the\toks@}%
          \ifdefvoid\glsxtrsupplocationurl
          {%
            \expandafter\glsxtrfmtinternalnameref\expandafter{\@glo@tmp}%
            {\@glsnumberformat}{\glsxtr@locationhypertext}%
          }%
          {%
            \expandafter\glsxtrfmtexternalnameref\expandafter{\@glo@tmp}%
            {\@glsnumberformat}{\glsxtr@locationhypertext}{\glsxtrsupplocationurl}%
          }%
        \fi
      \else
        \@gls@ReturnAfterFi{%
          \@glsxtr@bibgls@removespaces#2\@nil
        }%
      \fi
    }
```

$\boxed{\text{\textbackslash glsxtrfmtinternalnameloc\{⟨target⟩\}\{⟨format⟩\}\{⟨title⟩\}}}$

\glsxtrfmtinternalnameref

```
\newcommand*{\glsxtrfmtinternalnameref}[3]{%
  \csuse{#2}{\glsdohyperlink{#1}{#3}}%
}
```

$\boxed{\begin{array}{l}\text{\textbackslash glsxtrfmtexternalnameloc\{⟨target⟩\}\{⟨format⟩\}\{⟨title⟩\}}\\ \text{\{⟨file⟩\}}\end{array}}$

\glsxtrfmtexternalnameref

```
\newcommand*{\glsxtrfmtexternalnameref}[4]{%
  \csuse{#2}{\hyperref{#4}{}{#1}{#3}}%
}
```

$\boxed{\text{\textbackslash glsxtrSetWidest\{⟨type⟩\}\{⟨level⟩\}\{⟨text⟩\}}}$

\glsxtrSetWidest

As from bib2gls v1.8, this is used by the set-widest resource option for the
alttree and the styles provided by the glossary-longextra package.

`\newcommand*{\glsxtrSetWidest}[3]{%`

Check which style options have been provided. (The style packages may not
have been loaded.)

```
\ifdef\glsupdatewidest
{%
  \ifdef\glslongextraUpdateWidest
  {%
```

Relevant style packages all loaded. If the ⟨*type*⟩ has been given, append to glossary preamble.

```
    \ifstrempty{#1}
    {%
      \glsupdatewidest[#2]{#3}%
      \ifnum#2=0\relax
        \glslongextraUpdateWidest{#3}%
      \else
        \glslongextraUpdateWidestChild{#2}{#3}%
      \fi
    }%
    {%
      \apptoglossarypreamble[#1]{\glsupdatewidest[#2]{#3}}%
      \ifnum#2=0\relax
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
      \else
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
      \fi
    }%
  }%
  {%
```

Only alttree.

```
    \ifstrempty{#1}
    {%
      \glsupdatewidest[#2]{#3}%
    }%
    {%
      \apptoglossarypreamble[#1]{\glsupdatewidest[#2]{#3}}%
    }%
  }%
}%
{%
```

`\glsupdatewidest` hasn't been defined. This could just mean that the **glossaries-extra-stylemods** package hasn't been loaded.

```
  \ifdef\glssetwidest
  {%
    \ifdef\glslongextraUpdateWidest
    {%
```

Relevant **glossary-tree** and **glossary-longextra** have been loaded. If the ⟨*type*⟩ has been given, append to glossary preamble.

```
      \ifstrempty{#1}
      {%
        \glssetwidest[#2]{#3}%
```

607

```
      \ifnum#2=0\relax
        \glslongextraUpdateWidest{#3}%
      \else
        \glslongextraUpdateWidestChild{#2}{#3}%
      \fi
  }%
  {%
    \apptoglossarypreamble[#1]{\glssetwidest[#2]{#3}}%
    \ifnum#2=0\relax
      \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
    \else
      \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
    \fi
  }%
  }%
  {%
```

Only alttree.

```
    \ifstrempty{#1}
    {%
      \glssetwidest[#2]{#3}%
    }%
    {%
      \apptoglossarypreamble[#1]{\glssetwidest[#2]{#3}}%
    }%
  }%
}%
{%
  \ifdef\glslongextraUpdateWidest
  {%
```

glossary-longextra has been loaded.

```
    \ifstrempty{#1}
    {%
      \ifnum#2=0\relax
        \glslongextraUpdateWidest{#3}%
      \else
        \glslongextraUpdateWidestChild{#2}{#3}%
      \fi
    }%
    {%
      \ifnum#2=0\relax
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
      \else
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
      \fi
    }%
  }%
```

Neither glossary-tree nor glossary-longextra have been loaded. Do nothing.

```
  {}%
```

608

```
      }%
    }%
  }
```

\glsxtrSetWidestFallback

$\boxed{\texttt{\textbackslash glsxtrSetWidestFallback\{}\langle\textit{max depth}\rangle\texttt{\}\{}\langle\textit{list}\rangle\texttt{\}}}$

Used when `bib2gls` can't determine the widest name. The ⟨*list*⟩ argument is a comma-separated list of glossary labels. The ⟨*max depth*⟩ refers to the maximum hierarchical depth. This will either be 0 (only top-level entries) or 2 (up to two child-levels).

```
\newcommand*{\glsxtrSetWidestFallback}[2]{%
  \ifnum#1=0\relax
  \ifdef\glsFindWidestTopLevelName
  {%
    \glsFindWidestTopLevelName[#2]%
  }%
  {%
    \GlossariesExtraWarning{You need stylemods={tree} to
      provide a fallback for set-widest}%
  }%
  \else
  \ifdef\glsFindWidestLevelTwo
  {%
    \glsFindWidestLevelTwo[#2]%
    \ifdef\glslongextraUpdateWidestChild
    {%
     \glslongextraUpdateWidestChild{#1}{\csuse{@glswidestnamei}}%
     \glslongextraUpdateWidestChild{#1}{\csuse{@glswidestnameii}}%
    }%
    {}%
  }%
  {%
    \GlossariesExtraWarning{You need stylemods={tree} to
      provide a fallback for set-widest}%
  }%
  \fi
}
```

\@glsxtr@labelprefixes List of label prefixes.

```
\newcommand*{\@glsxtr@labelprefixes}{}
```

\glsxtrclearlabelprefixes List of label prefixes.

```
\newcommand*{\glsxtrclearlabelprefixes}{%
  \renewcommand*{\@glsxtr@labelprefixes}{}%
}
```

609

`\glsxtraddlabelprefix` Add prefix to the list. These should be added in the order of precedence with the last one as a fallback. This doesn't check against duplicates as it may be useful to replicate a prefix at the end as the fallback.

```
\newcommand*{\glsxtraddlabelprefix}[1]{%
  \ifstrempty{#1}%
  {\glsxtraddlabelprefix{\empty}}%
  {%
    \ifdefempty\@glsxtr@labelprefixes
    {\def\@glsxtr@labelprefixes{#1}}%
    {\appto\@glsxtr@labelprefixes{,#1}}%
  }%
}
```

`\glsxtrprependlabelprefix` Inserts at the start of the list.

```
\newcommand*{\glsxtrprependlabelprefix}[1]{%
  \ifstrempty{#1}%
  {\glsxtrprependlabelprefix{\empty}}%
  {%
    \ifdefempty\@glsxtr@labelprefixes
    {\def\@glsxtr@labelprefixes{#1}}%
    {\preto\@glsxtr@labelprefixes{#1,}}%
  }%
}
```

`\glsxtrifinlabelprefixlist`

> `\glsxtrifinlabelprefixlist{⟨prefix⟩}{⟨true⟩}{⟨false⟩}`

Test if the given prefix is in the list.

```
\newcommand*{\glsxtrifinlabelprefixlist}[3]{%
  \ifstrempty{#1}%
  {\glsxtrifinlabelprefixlist{\empty}{#2}{#3}}%
  {%
    \DTLifinlist{#1}{\@glsxtr@labelprefixes}{#2}{#3}%
  }%
}
```

`\@glsxtr@prefixlabellist` This is provided for the benefit of `bib2gls`. It's possible that the user may add more prefixes after the start of the document, but that can lead to inconsistencies. The final element of the list (the fallback) is the only prefix of interest for `bib2gls`.

```
\AtBeginDocument{%
  \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@prefixlabellist}[1]{}}%
  \protected@write\@auxout{}{\string\@glsxtr@prefixlabellist{\@glsxtr@labelprefixes}}%
}
```

Before v1.49, the last label was used as a fallback, but this doesn't make sense when the first matching label is used when entries are defined. The selection should be deferred to `bib2gls`, which means passing the list of label choices to `bib2gls`.

\@glsxtr@dglsnomatch No match found so record all possibilities. Requires `bib2gls` v3.0+. This will add the final insert argument but won't be able to apply any case-changing etc.

```
\def\@glsxtr@dglsnomatch#1#2[#3]{%
  \begingroup
```

This is a cut-down version of `\@@glsxtr@record`. Use the fallback label in the event any hooks have to reference `\glslabel`. This is mainly to prevent an undefined control sequence error. It can't be relied on as the actual label.

```
\let\glslabel\@gls@thislabel
\let\glsnumberformat\@glsxtr@defaultnumberformat
\def\@glsxtr@thevalue{}%
\def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
\let\@glsxtr@org@theHvalue\@glsxtr@theHvalue
\let\@gls@counter\glscounter
\if@glsxtr@equations
  \@glsxtr@use@equation@counter
\fi
\@gls@setdefault@glslink@opts
\@glsxtr@glslink@prekeys
\setkeys{glslink}{#1}%
\glsxtr@do@autoadd{glslink}%
```

Can't increment associated counter.

```
\ifKV@glslink@noindex
  \GlossariesExtraWarning{Can't obtain a match for prefix
   candidates: \@glsxtr@prefixedlist. Check the label spelling or rerun}%
\else
  \ifdefempty{\@glsxtr@thevalue}%
  {%
    \ifx\@glsxtr@org@theHvalue\@glsxtr@theHvalue
    \else
      \let\theHglsentrycounter\@glsxtr@theHvalue
    \fi
  }%
  {%
    \let\theglsentrycounter\@glsxtr@thevalue
    \let\theHglsentrycounter\@glsxtr@theHvalue
  }%
  \glsxtr@saveentrycounter
  \@@glsxtr@dorecord\@glsxtr@prefixedlist
    \glsxtr@select@entry\glsxtr@do@select@nameref@record
```

Issue warning.

```
  \GlossariesExtraWarning{Can't obtain a match for prefix
   candidates: \@glsxtr@prefixedlist. Check the label spelling, use bib2gls v3.0+ to
   select entry and rerun LaTeX}%
\fi
\@glsxtrundeftag#3%
\endgroup
}
```

611

`\glsxtr@select@entry` Instruction to `bib2gls` to select the first found label in the list.

```
\newcommand*{\glsxtr@select@entry}[5]{}
```

`\glsxtr@select@entry@nameref` Instruction to `bib2gls` to select the first found label in the list as though it has a record.

```
\newcommand*{\glsxtr@select@entry@nameref}[8]{}
```

`\glsxtr@do@select@nameref@record` Instruction to `bib2gls` to select the first found label in the list as though it has a record.

```
\newcommand*{\glsxtr@do@select@nameref@record}[5]{%
  \gls@ifnotmeasuring
  {%
    \protected@write\@auxout{}{\string\glsxtr@select@entry@nameref
    {#1}{#2}{#3}{#4}{#5}%
    {\csuse{@currentlabelname}}{\csuse{@currentHref}}%
    {\theHglsentrycounter}}%
  }%
}
```

`\ifGlsXtrPrefixLabelFallbackLast` Determine whether the first or last label should be used as the fallback in the event that there's no match on any prefixes.

```
\newif\ifGlsXtrPrefixLabelFallbackLast
\GlsXtrPrefixLabelFallbackLasttrue
```

`\@glsxtr@get@prefixedlabel` Iterate through all the prefixes and find the first prefix and label combination that exists. If none found, this could mean that it's the first LaTeX run.

```
\newcommand*{\@glsxtr@get@prefixedlabel}[1]{%
```

Grouping is used in case of a nested for loop.

```
\begingroup
```

Initialise to the unprefixed label in the event that the list is empty.

```
\protected@edef\@gls@thislabel{#1}%
```

Save the first label.

```
\let\@glsxtr@prefixedfirstlabel\@gls@thislabel
\def\@glsxtr@set@prefixedfirstlabel{%
  \let\@glsxtr@prefixedfirstlabel\@gls@thislabel
  \let\@glsxtr@set@prefixedfirstlabel\relax
}%
```

List of labels in the event that no combination is found.

```
\let\@glsxtr@prefixedlist\@empty
```

Iterate over all labels.

```
\count@=0\relax
\@for\@glsxtr@prefix:=\@glsxtr@labelprefixes\do
{%
  \advance\count@ by 1\relax
  \protected@edef\@gls@thislabel{\@glsxtr@prefix#1}%
  \@glsxtr@set@prefixedfirstlabel
```

612

Check if this label exists.

```
\ifglsentryexists{\@gls@thislabel}%
{%
  \@endfortrue
```

Found a label that exists. Clear the list.

```
  \let\@glsxtr@prefixedlist\@empty
}%
{%
```

Append or prepend to list.

```
  \ifdefempty\@glsxtr@prefixedlist
  {\let\@glsxtr@prefixedlist\@gls@thislabel}%
  {%
    \ifGlsXtrPrefixLabelFallbackLast
      \epreto\@glsxtr@prefixedlist{\expandonce\@gls@thislabel,}%
    \else
      \eappto\@glsxtr@prefixedlist{,\expandonce\@gls@thislabel}%
    \fi
  }%
}%
}%
\if@endfor
```

Loop ended prematurely, which means label was found.

```
\else
 \ifnum\count@>1\relax
   \ifGlsXtrPrefixLabelFallbackLast
   \else
```

Fallback on first label.

```
    \let\@gls@thislabel\@glsxtr@prefixedfirstlabel
   \fi
 \else
```

Only one prefix so assume that one.

```
   \let\@glsxtr@prefixedlist\@empty
 \fi
\fi

\edef\@glo@tmp{\endgroup
\noexpand\def\noexpand\@glsxtr@prefixedlist{\expandonce\@glsxtr@prefixedlist}%
\noexpand\def\noexpand\@gls@thislabel{\expandonce\@gls@thislabel}}\@glo@tmp
}
```

`\@@dgls@` Used by all the `\dgls`-like commands to find the first match.

```
\newcommand*{\@@dgls@}[3]{%
\@glsxtr@get@prefixedlabel{#2}%
\ifx\@glsxtr@prefixedlist\@empty
 \let\@dgls@@next#3%
\else
 \let\@dgls@@next\@glsxtr@dglsnomatch
```

613

```
      \fi
      \new@ifnextchar[{\@dgls@@next{#1}{\@gls@thislabel}}%
       {\@dgls@@next{#1}{\@gls@thislabel}[]}%
    }
```

\dgls Like \gls but tries the prefixes. (Can't use \pgls as that's provided by
       glossaries-prefix.) Since this command is designed for bib2gls's dual entry sys-
       tem, the "d" stands for "dual".

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

\@dgls

```
\newcommand*{\@dgls}[2][]{\@@dgls@{#1}{#2}{\@gls@}}
```

\dglspl

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

\@dglspl

```
\newcommand*{\@dglspl}[2][]{\@@dgls@{#1}{#2}{\@glspl@}}
```

\dGls

```
\newrobustcmd*{\dGls}{\@gls@hyp@opt\@dGls}
\glsmfuaddmap{\dgls}{\dGls}
```

\@dGls

```
\newcommand*{\@dGls}[2][]{\@@dgls@{#1}{#2}{\@Gls@}}
```

\dGlspl

```
\newrobustcmd*{\dGlspl}{\@gls@hyp@opt\@dGlspl}
\glsmfuaddmap{\dglspl}{\dGlspl}
```

\@dGlspl

```
\newcommand*{\@dGlspl}[2][]{\@@dgls@{#1}{#2}{\@Glspl@}}
```

\dGLS

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

\@dGLS

```
\newcommand*{\@dGLS}[2][]{\@@dgls@{#1}{#2}{\@GLS@}}
```

\dGLSpl

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

\@dGLSpl

```
\newcommand*{\@dGLSpl}[2][]{\@@dgls@{#1}{#2}{\@GLSpl@}}
```

614

\dglslink Like \glslink but tries the prefixes.

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

\@dglslink

```
\newcommand*{\@dglslink}[3][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \glslink[#1]{\@gls@thislabel}{#3}%
}
```

\dGlslink Sentence-case version to provide a mapping.

```
\newrobustcmd*{\dGlslink}{\@gls@hyp@opt\@dGlslink}
\glsmfuaddmap{\dglslink}{\dGlslink}
```

\@dGlslink

```
\newcommand*{\@dGlslink}[3][]{%
  \dglslink[#1]{#2}{\glssentencecase{#3}}%
}
```

\dglsdisp Like \glsdisp but tries the prefixes.

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

\@dglsdisp Like \glsdisp but tries the prefixes.

```
\newcommand*{\@dglsdisp}[3][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \glsdisp[#1]{\@gls@thislabel}{#3}%
}
```

\dGlsdisp Sentence-case version to provide a mapping.

```
\newrobustcmd*{\dGlsdisp}{\@gls@hyp@opt\@dGlsdisp}
\glsmfuaddmap{\dglsdisp}{\dGlsdisp}
```

\@dGlsdisp

```
\newcommand*{\@dGlsdisp}[3][]{%
  \dglsdisp[#1]{#2}{\glssentencecase{#3}}%
}
```

Similar to the above but searches for a match with the given field set.

xtr@get@prefixedlabel@field The second argument is the field's internal label.

```
\newcommand*{\@glsxtr@get@prefixedlabel@field}[2]{%
  \protected@edef\dglsfieldcurrentfieldlabel{#2}%
  \let\dglsfieldactualfieldlabel\dglsfieldcurrentfieldlabel
```

Grouping is used in case of a nested for loop.

```
  \begingroup
```

Initialise to the unprefixed label in the event that the list is empty.

```
  \protected@edef\@gls@thislabel{#1}%
```

Save the first label.

```
\let\@glsxtr@prefixedfirstlabel\@gls@thislabel
\def\@glsxtr@set@prefixedfirstlabel{%
  \let\@glsxtr@prefixedfirstlabel\@gls@thislabel
  \let\@glsxtr@set@prefixedfirstlabel\relax
}%
```

Initialise fallback label.

```
\let\@gls@fallbacklabel\relax
```

List of labels in the event that no combination is found.

```
\let\@glsxtr@prefixedlist\@empty
```

Iterate over all labels.

```
\count@=0\relax
\@for\@glsxtr@prefix:=\@glsxtr@labelprefixes\do
{%
  \advance\count@ by 1\relax
  \protected@edef\@gls@thislabel{\@glsxtr@prefix#1}%
  \@glsxtr@set@prefixedfirstlabel
```

Check if this label exists.

```
\ifglsentryexists{\@gls@thislabel}%
{%
```

Found a label that exists. Has the field been set?

```
\ifcsvoid{glo@\glsdetoklabel{\@gls@thislabel}@#2}%
{%
```

Field hasn't been set. Has a fallback been set yet?

```
\ifx\@gls@fallbacklabel\relax
\ifcsvoid
{glo@\glsdetoklabel{\@gls@thislabel}@\dglsfieldfallbackfieldlabel}%
{%
  \GlossariesExtraInfo{Found entry '\@gls@thislabel' that
   matches prefix '\@glsxtr@prefix' but field '#2' not set
   and fallback field '\dglsfieldfallbackfieldlabel' not set}%
}%
{%
  \let\@gls@fallbacklabel\@gls@thislabel
  \GlossariesExtraInfo{Found entry '\@gls@thislabel' that
   matches prefix '\@glsxtr@prefix' but field '#2' not set.
   Fallback field '\dglsfieldfallbackfieldlabel' is set
   so setting fallback entry to '\@gls@fallbacklabel' with
   field '\dglsfieldfallbackfieldlabel'}%
}%
\else
  \GlossariesExtraInfo{Found entry '\@gls@thislabel' that
   matches prefix '\@glsxtr@prefix' but field '#2' not set.
   Fallback entry: '\@gls@fallbacklabel'}%
\fi
```

616

Add to list. (A new entry with the desired field may have been added, so allow it to be selected.)

```
        \ifdefempty\@glsxtr@prefixedlist
        {\let\@glsxtr@prefixedlist\@gls@thislabel}%
        {%
          \ifGlsXtrPrefixLabelFallbackLast
            \epreto\@glsxtr@prefixedlist{\expandonce\@gls@thislabel,}%
          \else
            \eappto\@glsxtr@prefixedlist{,\expandonce\@gls@thislabel}%
          \fi
        }%
      }%
      {%
        \@endfortrue
```

The field has been set. Clear the list.

```
        \let\@glsxtr@prefixedlist\@empty
      }%
    }%
    {%
```

Append or prepend to list.

```
      \ifdefempty\@glsxtr@prefixedlist
      {\let\@glsxtr@prefixedlist\@gls@thislabel}%
      {%
        \ifGlsXtrPrefixLabelFallbackLast
          \epreto\@glsxtr@prefixedlist{\expandonce\@gls@thislabel,}%
        \else
          \eappto\@glsxtr@prefixedlist{,\expandonce\@gls@thislabel}%
        \fi
      }%
    }%
  }%
  \if@endfor
```

Loop ended prematurely, which means label was found.

```
  \else
```

Label not found. Was the fallback field found?

```
    \ifx\@gls@fallbacklabel\relax
      \GlossariesExtraWarning{No fallback found for '#1'}%
```

No field fallback found.

```
      \ifnum\count@>1\relax
        \ifGlsXtrPrefixLabelFallbackLast
        \else
```

Fallback on first label.

```
          \let\@gls@thislabel\@glsxtr@prefixedfirstlabel
        \fi
      \else
```

Only one prefix so assume that one.

```
        \let\@glsxtr@prefixedlist\@empty
     \fi
   \else
```

Fallback field was found. Use the fallback entry.

```
     \let\@gls@thislabel\@gls@fallbacklabel
     \let\dglsfieldactualfieldlabel\dglsfieldfallbackfieldlabel
```

Clear prefix candidate list.

```
     \let\@glsxtr@prefixedlist\@empty
   \fi
  \fi

  \edef\@glo@tmp{\endgroup
   \noexpand\def\noexpand\@glsxtr@prefixedlist{\expandonce\@glsxtr@prefixedlist}%
   \noexpand\def\noexpand\@gls@thislabel{\expandonce\@gls@thislabel}%
   \noexpand\def\noexpand\dglsfieldactualfieldlabel
    {\expandonce\dglsfieldactualfieldlabel}%
  }%
  \@glo@tmp
 }
```

\@@dgls@@field

> \@@dgls@@field{⟨options⟩}{⟨label⟩}{⟨field⟩}{⟨cs⟩}

```
\newcommand*{\@@dgls@@field}[4]{%
  \@glsxtr@get@prefixedlabel@field{#2}{#3}%
  \ifx\@glsxtr@prefixedlist\@empty
   \let\@dgls@@next#4%
  \else
   \let\@dgls@@next\@glsxtr@dglsnomatch
  \fi
  \new@ifnextchar[{\@dgls@@next{#1}{\@gls@thislabel}}%
   {\@dgls@@next{#1}{\@gls@thislabel}[]}%
 }
```

\dglsfieldcurrentfieldlabel Set by the \dglsfield commands to the current field label. This is the field requested in the argument of \dglsfield.

```
\newcommand*{\dglsfieldcurrentfieldlabel}{}
```

\dglsfieldfallbackfieldlabel The field to use if the required field isn't set.

```
\newcommand*{\dglsfieldfallbackfieldlabel}{text}
```

\dglsfieldactualfieldlabel This is the field that's actually used.

```
\newcommand*{\dglsfieldactualfieldlabel}{\dglsfieldcurrentfieldlabel}
```

\dglsfield

> \dglsfield[⟨options⟩]{⟨label⟩}{⟨field⟩}[⟨insert⟩]

618

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

\@dglsfield

```
\newcommand*{\@dglsfield}[3][]{%
 \@@dgls@@field{#1}{#2}{#3}{\@dgls@field}}
```

\@dgls@field

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

\dGlsfield

$\dGlsfield[\langle options\rangle]\{\langle label\rangle\}\{\langle field\rangle\}[\langle insert\rangle]$

```
\newrobustcmd*{\dGlsfield}{\@gls@hyp@opt\@dGlsfield}
\glsmfuaddmap{\dglsfield}{\dGlsfield}
```

\@dGlsfield

```
\newcommand*{\@dGlsfield}[3][]{%
 \@@dgls@@field{#1}{#2}{#3}{\@dGls@field}%
}
```

\@dGls@field

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

\dGLSfield

$\dGlsfield[\langle options\rangle]\{\langle label\rangle\}\{\langle field\rangle\}[\langle insert\rangle]$

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

\@dGLSfield

```
\newcommand*{\@dGLSfield}[3][]{%
 \@@dgls@@field{#1}{#2}{#3}{\@dGLS@field}%
}
```

\@dGLS@field

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

\d@inner@glsfield

$\d@inner@glsfield\{\langle default\ options\rangle\}\{\langle field\rangle\}\langle modifier\rangle$ $[\langle options\rangle]\{\langle label\rangle\}[\langle insert\rangle]$

619

```
\newrobustcmd*{\d@inner@glsfield}[2]{%
 \ifstrempty{#1}
 {\def\@d@inner@glsfield@opts{}}%
 {\def\@d@inner@glsfield@opts{#1,}}%
 \def\dglsfieldcurrentfieldlabel{#2}%
 \@gls@hyp@opt\@d@inner@glsfield
}
```

\@d@inner@glsfield

```
\newcommand*{\@d@inner@glsfield}[2][]{%
 \expandafter\@@dgls@@field\expandafter
  {\@d@inner@glsfield@opts#1}{#2}{\dglsfieldcurrentfieldlabel}{\@dgls@field}}
```

\d@inner@Glsfield

```
\newrobustcmd*{\d@inner@Glsfield}[2]{%
 \ifstrempty{#1}
 {\def\@d@inner@glsfield@opts{}}%
 {\def\@d@inner@glsfield@opts{#1,}}%
 \def\dglsfieldcurrentfieldlabel{#2}%
 \@gls@hyp@opt\@d@inner@Glsfield
}
```

\@d@inner@Glsfield

```
\newcommand*{\@d@inner@Glsfield}[2][]{%
 \expandafter\@@dgls@@field\expandafter
  {\@d@inner@glsfield@opts#1}{#2}{\dglsfieldcurrentfieldlabel}{\@dGls@field}}
```

\d@inner@GLSfield

```
\newrobustcmd*{\d@inner@GLSfield}[2]{%
 \ifstrempty{#1}
 {\def\@d@inner@glsfield@opts{}}%
 {\def\@d@inner@glsfield@opts{#1,}}%
 \def\dglsfieldcurrentfieldlabel{#2}%
 \@gls@hyp@opt\@d@inner@GLSfield
}
```

\@d@inner@GLSfield

```
\newcommand*{\@d@inner@GLSfield}[2][]{%
 \expandafter\@@dgls@@field\expandafter
  {\@d@inner@glsfield@opts#1}{#2}{\dglsfieldcurrentfieldlabel}{\@dGLS@field}}
```

\newdglsfield

> \newdglsfield[⟨*options*⟩]{⟨*field*⟩}{⟨*cs*⟩}

```
\newrobustcmd*{\newdglsfield}[3][]{%
  \newrobustcmd*{#3}{\d@inner@glsfield{#1}{#2}}%
}
```

$\newdglsfieldlike[\langle options \rangle]\{\langle field \rangle\}\{\langle cs \rangle\}\{\langle Cs \rangle\}\{\langle CS \rangle\}$

```
\newrobustcmd*{\newdglsfieldlike}[5][]{%
  \newrobustcmd*{#3}{\d@inner@glsfield{#1}{#2}}%
  \newrobustcmd*{#4}{\d@inner@Glsfield{#1}{#2}}%
  \newrobustcmd*{#5}{\d@inner@GLSfield{#1}{#2}}%
  \glsmfuaddmap{#3}{#4}%
  \glsmfublocker{#5}%
}
```

Multi (compound/combined) entry commands used by bib2gls.

$\glsxtrmultientryadjustedname\{\langle list1 \rangle\}\{\langle name \rangle\}\{\langle list2 \rangle\}$
$\{\langle label \rangle\}$

This command is used by bib2gls when it adjusts the name field of an entry that's been identified as a main entry in the multi-entry set $\langle label \rangle$.

The final argument $\langle label \rangle$ is the multi-entry label from which the set was obtained. The first argument $\langle list1 \rangle$ is the list of other labels that come before the main label. The third argument $\langle list2 \rangle$ is the remaining list of other labels. The $\langle name \rangle$ argument is the previous name before adjustment.

```
\newrobustcmd*{\glsxtrmultientryadjustedname}[4]{%
  \bgroup
    \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
    \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
    \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
    \let\@glsxtrmultientryadjustednameother\glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefmt\glsxtrmultientryadjustednamefmt
    \let\@glsxtrmultientryadjustednamefirstother\glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefirstfmt\glsxtrmultientryadjustednamefmt
    \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
```

First letter upper case

```
\newrobustcmd*{\Glsxtrmultientryadjustedname}[4]{%
  \bgroup
    \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
    \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
    \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
    \let\@glsxtrmultientryadjustednameother\glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefmt\glsxtrmultientryadjustednamefmt
    \let\@glsxtrmultientryadjustednamefirstother\Glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefirstfmt\Glsxtrmultientryadjustednamefmt
    \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
```

```
                              \glsmfuaddmap{\glsxtrmultientryadjustedname}{\Glsxtrmultientryadjustedname}
```

GlsXtrmultientryadjustedname Title case
```
                              \newrobustcmd*{\GlsXtrmultientryadjustedname}[4]{%
                                \bgroup
                                  \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
                                  \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
                                  \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
                                  \let\@glsxtrmultientryadjustednameother\GlsXtrmultientryadjustednameother
                                  \let\@glsxtrmultientryadjustednamefmt\GlsXtrmultientryadjustednamefmt
                                  \let\@glsxtrmultientryadjustednamefirstother\GlsXtrmultientryadjustednameother
                                  \let\@glsxtrmultientryadjustednamefirstfmt\GlsXtrmultientryadjustednamefmt
                                  \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
                                \egroup
                              }
                              \glsmfublocker{\GlsXtrmultientryadjustedname}
```

GLSxtrmultientryadjustedname All caps.
```
                              \newrobustcmd*{\GLSxtrmultientryadjustedname}[4]{%
                                \bgroup
                                  \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
                                  \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
                                  \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
                                  \let\@glsxtrmultientryadjustednameother\GLSxtrmultientryadjustednameother
                                  \let\@glsxtrmultientryadjustednamefmt\GLSxtrmultientryadjustednamefmt
                                  \let\@glsxtrmultientryadjustednamefirstother\GLSxtrmultientryadjustednameother
                                  \let\@glsxtrmultientryadjustednamefirstfmt\GLSxtrmultientryadjustednamefmt
                                  \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
                                \egroup
                              }
                              \glsmfublocker{\GLSxtrmultientryadjustedname}
```

glsxtrmultientryadjustedname
```
                              \newcommand*{\@glsxtrmultientryadjustedname}[4]{%
                                \letcs\mglscurrentmainlabel{@gls@combined@#4@main}%
                                \letcs\mglscurrentmainlist{@gls@combined@#4@list}%
                                \letcs\mglscurrentmainoptions{@gls@combined@#4@options}%
                                \ifblank{#1}%
                                {%
                                  \@glsxtrmultientryadjustednamefirstfmt{#2}%
                                }%
                                {%
                                  \def\@mgls@previouslabel{}%
                                  \let\@gls@xtradjustedother\@glsxtrmultientryadjustednamefirstother
                                  \@for\mglscurrentlabel:=#1\do{%
                                    \ifx\@mgls@previouslabel\empty
                                    \else
                                      \@glsxtrmultientryadjustednamesep{\@mgls@previouslabel}{\mglscurrentlabel}%
                                    \fi
```

```
                        \@gls@xtradjustedother{\mglscurrentlabel}%
                        \let\@mgls@previouslabel\mglscurrentlabel
                        \let\@gls@xtradjustedother\@glsxtrmultientryadjustednameother
                  }%
                  \@glsxtrmultientryadjustednamepresep{\@mgls@previouslabel}{\mglscurrentmainlabel}%
                  \@glsxtrmultientryadjustednamefmt{#2}%
              }%
              \ifblank{#3}%
              {}%
              {%
                \let\@mgls@previouslabel\mglscurrentmainlabel
                \let\@gls@xtrmultientryadjustednamesep\@glsxtrmultientryadjustednamepostsep
                \@for\mglscurrentlabel:=#3\do{%
                    \@gls@xtrmultientryadjustednamesep{\@mgls@previouslabel}{\mglscurrentlabel}%
                    \@glsxtrmultientryadjustednameother{\mglscurrentlabel}%
                    \let\@mgls@previouslabel\mglscurrentlabel
                    \let\@gls@xtrmultientryadjustednamesep\@glsxtrmultientryadjustednamesep
                }%
              }%
          }
```

```
\newcommand*{\glsxtrmultientryadjustednamesep}{\glscombinedfirstsepfirst}
```

Separator before main name.

```
\newcommand*{\glsxtrmultientryadjustednamepresep}{\glsxtrmultientryadjustednamesep}
```

Separator after main name.

```
\newcommand*{\glsxtrmultientryadjustednamepostsep}{\glsxtrmultientryadjustednamesep}
```

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

```
\newcommand*{\glsxtrmultientryadjustednameother}[1]{\glsentryname{#1}}
```

```
\newcommand*{\Glsxtrmultientryadjustednamefmt}[1]{\glssentencecase{#1}}
```

```
\newcommand*{\Glsxtrmultientryadjustednameother}[1]{\Glsentryname{#1}}
```

```
\newcommand*{\GlsXtrmultientryadjustednameother}[1]{%
 \glsentrytitlecase{#1}{name}}
```

```
\ifdef\glscapitalisewords
{%
```

```
                        \newcommand*{\GlsXtrmultientryadjustednamefmt}[1]{\glscapitalisewords{#1}}
                      }
                      {
                        \newcommand*{\GlsXtrmultientryadjustednamefmt}[1]{\capitalisewords{#1}}
                      }
```

multientryadjustednameother

```
                      \newcommand*{\GLSxtrmultientryadjustednameother}[1]{%
                       \glsuppercase{\glsentryname{#1}}}
```

xtrmultientryadjustednamefmt

```
                      \newcommand*{\GLSxtrmultientryadjustednamefmt}[1]{\glsuppercase{#1}}
```

Provide missing Greek letters for use in maths mode. These commands are recognised by `bib2gls` and will be mapped to the Mathematical Greek Italic letters. This ensures that the Greek letters that have the same shape as Latin letters are kept with the other mathematical Greek letters for sorting purposes. The LaTeX version of these commands (provided here) use an upright font for capitals and italic for lower case to provide a better match with the other Greek symbols provided by the kernel.

\Alpha

```
                      \providecommand*{\Alpha}{\mathrm{A}}
```

\Beta

```
                      \providecommand*{\Beta}{\mathrm{B}}
```

\Epsilon

```
                      \providecommand*{\Epsilon}{\mathrm{E}}
```

\Zeta

```
                      \providecommand*{\Zeta}{\mathrm{Z}}
```

\Eta

```
                      \providecommand*{\Eta}{\mathrm{H}}
```

\Iota

```
                      \providecommand*{\Iota}{\mathrm{I}}
```

\Kappa

```
                      \providecommand*{\Kappa}{\mathrm{K}}
```

\Mu

```
                      \providecommand*{\Mu}{\mathrm{M}}
```

\Nu

```
                      \providecommand*{\Nu}{\mathrm{N}}
```

`\Omicron`

> `\providecommand*{\Omicron}{\mathrm{O}}`

`\Rho`

> `\providecommand*{\Rho}{\mathrm{P}}`

`\Tau`

> `\providecommand*{\Tau}{\mathrm{T}}`

`\Chi`

> `\providecommand*{\Chi}{\mathrm{X}}`

`\Digamma`

> `\providecommand*{\Digamma}{\mathrm{F}}`

`\omicron`

> `\providecommand*{\omicron}{\mathit{o}}`

Provide corresponding upright characters if upgreek has been loaded. (The upper case characters are the same as above.)

> `\@ifpackageloaded{upgreek}%`
> `{`

`\Upalpha`

> `\providecommand*{\Upalpha}{\mathrm{A}}`

`\Upbeta`

> `\providecommand*{\Upbeta}{\mathrm{B}}`

`\Upepsilon`

> `\providecommand*{\Upepsilon}{\mathrm{E}}`

`\Upzeta`

> `\providecommand*{\Upzeta}{\mathrm{Z}}`

`\Upeta`

> `\providecommand*{\Upeta}{\mathrm{H}}`

`\Upiota`

> `\providecommand*{\Upiota}{\mathrm{I}}`

`\Upkappa`

> `\providecommand*{\Upkappa}{\mathrm{K}}`

`\Upmu`

> `\providecommand*{\Upmu}{\mathrm{M}}`

`\Upnu`

```
\providecommand*{\Upnu}{\mathrm{N}}
```

`\Upomicron`

```
\providecommand*{\Upomicron}{\mathrm{O}}
```

`\Uprho`

```
\providecommand*{\Uprho}{\mathrm{P}}
```

`\Uptau`

```
\providecommand*{\Uptau}{\mathrm{T}}
```

`\Upchi`

```
\providecommand*{\Upchi}{\mathrm{X}}
```

`\upomicron`

```
\providecommand*{\upomicron}{\mathrm{o}}
```

```
 }%
 {}% upgreek.sty not loaded
```

This package provides some basic rules, but it's not intended for complete coverage of all locales. The CLDR should provide the appropriate locale-sensitive rules. These macros are primarily to help construct custom rules to include, for example, Greek maths symbols mixed with Latin. For the full rule syntax, see the Java API for RuleBaseCollator

If you want to provide a rule-block for a particular locale to allow for customization within that locale, create a file called `glossariesxtr-⟨tag⟩.ldf` (where ⟨*tag*⟩ identifies the locale) and add similar commands. See the description of `\IfTrackedLanguageFileExists` in the tracklang manual for the allowed forms of ⟨*tag*⟩. The simplest is to just use the root language label or ISO code. The file will then be automatically loaded by glossaries-extra if the document has support for that language.

When combining these blocks of rules, remember to separate them with the appropriate character. For example:

```
%sort-rule={\glsxtrcontrolrules
% ;\glsxtrspacerules
% ;\glsxtrnonprintablerules
% ;\glsxtrcombiningdiacriticrules
% ;\glsxtrhyphenrules
% <\glsxtrgeneralpuncrules
% <\glsxtrdigitrules
% <\glsxtrfractionrules
% <\glsxtrGeneralLatinIVrules
% <\glsxtrMathItalicGreekIrules
%}
%
```

`\glsxtrIgnorableRules` A shortcut command for common ignorable characters.

```
\newcommand{\glsxtrIgnorableRules}{%
\glsxtrcontrolrules
\string;\glsxtrspacerules
\string;\glsxtrnonprintablerules
}
```

`\glsxtrGeneralInitRules` A shortcut command for common initial rules for ignorables, diacritics, punctuation and digits.

```
\newcommand{\glsxtrGeneralInitRules}{%
\glsxtrIgnorableRules
\string;\glsxtrcombiningdiacriticrules
\string;\glsxtrhyphenrules
\string<\glsxtrgeneralpuncrules
\string<\glsxtrdigitrules
\string<\glsxtrfractionrules
}
```

`\glsxtrGeneralPuncRules` A shortcut command for common punctuation and digits.

```
\newcommand{\glsxtrGeneralPuncRules}{%
\glsxtrgeneralpuncmarksrules
\string<\glsxtrgeneralpuncdotrules
\string<\glsxtrgeneralpuncaccentsrules
\string<\glsxtrgeneralpuncquoterules
\string<\glsxtrgeneralpuncbracketrules
\string<\glsxtrgeneralpuncsignrules
\string<\glsxtrcurrencyrules
\string<\glsxtrgeneralpuncIIIrules
\string<\glsxtrdigitrules
\string<\glsxtrfractionrules
}
```

`\glsxtrcontrolrules` These are control characters that are usually placed at the start of a rule in the 'ignored characters' section. These control characters are unlikely to appear in any entry fields but are provided for completeness. (They may appear with the marker commands provided with `--datatool-sort-markers` which emulates the marker commands provided by **datatool-base** for use in the sort hook, in which case those particular control codes shouldn't be ignored.) `\string` is used for punctuation characters in case they've been made active.

```
\newcommand*{\glsxtrcontrolrules}{%
\string'\glshex 200B\string'\string=\glshex 200C\string=\glshex 200D
\string=\glshex 200E\string=\glshex 200F\string=\glshex 0000\string=\glshex 0001
\string=\glshex 0002\string=\glshex 0003\string=\glshex 0004\string=\glshex 0005
\string=\glshex 0006\string=\glshex 0007\string=\glshex 0008
\string=\string'\glshex 0009\string'\string=\string'\glshex 000B\string'
\string=\glshex 000E\string=\glshex 000F\string=\string'\glshex
0010\string'\string=\glshex 0011
\string=\glshex 0012\string=\glshex 0013\string=\glshex 0014\string=\glshex 0015
\string=\glshex 0016\string=\glshex 0017\string=\glshex 0018\string=\glshex 0019
```

```
                  \string=\glshex 001A\string=\glshex 001B\string=\glshex 001C\string=\glshex 001D
                  \string=\glshex 001E\string=\glshex 001F\string=\glshex 007F\string=\glshex 0080
                  \string=\glshex 0081\string=\glshex 0082\string=\glshex 0083\string=\glshex 0084
                  \string=\glshex 0085\string=\glshex 0086\string=\glshex 0087\string=\glshex 0088
                  \string=\glshex 0089\string=\glshex 008A\string=\glshex 008B\string=\glshex 008C
                  \string=\glshex 008D\string=\glshex 008E\string=\glshex 008F\string=\glshex 0090
                  \string=\glshex 0091\string=\glshex 0092\string=\glshex 0093\string=\glshex 0094
                  \string=\glshex 0095\string=\glshex 0096\string=\glshex 0097\string=\glshex 0098
                  \string=\glshex 0099\string=\glshex 009A\string=\glshex 009B\string=\glshex 009C
                  \string=\glshex 009D\string=\glshex 009E\string=\glshex 009F
                  }
```

\glsxtrcontrolIrules Subset of control rules. Doesn't include 0, 1C, 1D, 1E, 1F, and 7F.

```
                  \newcommand*{\glsxtrcontrolIrules}{%
                  \string'\glshex 200B\string'\string=\glshex 200C\string=\glshex 200D
                  \string=\glshex 200E\string=\glshex 200F\string=\glshex 0001
                  \string=\glshex 0002\string=\glshex 0003\string=\glshex 0004\string=\glshex 0005
                  \string=\glshex 0006\string=\glshex 0007\string=\glshex 0008
                  \string=\string'\glshex 0009\string'\string=\string'\glshex 000B\string'
                  \string=\glshex 000E\string=\glshex 000F\string=\string'\glshex
                  0010\string'\string=\glshex 0011
                  \string=\glshex 0012\string=\glshex 0013\string=\glshex 0014\string=\glshex 0015
                  \string=\glshex 0016\string=\glshex 0017\string=\glshex 0018\string=\glshex 0019
                  \string=\glshex 001A\string=\glshex 001B\string=\glshex 0080
                  \string=\glshex 0081\string=\glshex 0082\string=\glshex 0083\string=\glshex 0084
                  \string=\glshex 0085\string=\glshex 0086\string=\glshex 0087\string=\glshex 0088
                  \string=\glshex 0089\string=\glshex 008A\string=\glshex 008B\string=\glshex 008C
                  \string=\glshex 008D\string=\glshex 008E\string=\glshex 008F\string=\glshex 0090
                  \string=\glshex 0091\string=\glshex 0092\string=\glshex 0093\string=\glshex 0094
                  \string=\glshex 0095\string=\glshex 0096\string=\glshex 0097\string=\glshex 0098
                  \string=\glshex 0099\string=\glshex 009A\string=\glshex 009B\string=\glshex 009C
                  \string=\glshex 009D\string=\glshex 009E\string=\glshex 009F
                  }
```

\glsxtrcontrolIIrules Subset of ordered control rules (information separators). Doesn't include 7F.

```
                  \newcommand*{\glsxtrcontrolIIrules}{%
                  \glshex 001C\string<\glshex 001D
                  \string<\glshex 001E\string<\glshex 001F
                  }
```

\glsxtrspacerules These are space characters.

```
                  \newcommand*{\glsxtrspacerules}{%
                  \string' \string'\string;
                  \string'\glshex 00A0\string'\string;
                  \string'\glshex 2000\string'\string;
                  \string'\glshex 2001\string'\string;
                  \string'\glshex 2002\string'\string;
                  \string'\glshex 2003\string'\string;
                  \string'\glshex 2004\string'\string;
```

```
                              \string'\glshex 2005\string'\string;
                              \string'\glshex 2006\string'\string;
                              \string'\glshex 2007\string'\string;
                              \string'\glshex 2008\string'\string;
                              \string'\glshex 2009\string'\string;
                              \string'\glshex 200A\string'\string;
                              \string'\glshex 3000\string'
                             }
```

\glsxtrnonprintablerules These are non-printable characters (BOM, tabs, line feed and carriage return).

```
                             \newcommand*{\glsxtrnonprintablerules}{%
                              \string'\glshex FEFF\string'\string;
                              \string'\glshex 000A\string'\string;
                              \string'\glshex 0009\string'\string;
                              \string'\glshex 000C\string'\string;
                              \string'\glshex 000B\string'
                             }
```

\glsxtrcombiningdiacriticrules Combining diacritic marks. This is split into multiple macros.

```
                             \newcommand*{\glsxtrcombiningdiacriticrules}{%
                              \glsxtrcombiningdiacriticIrules\string;
                              \glsxtrcombiningdiacriticIIrules\string;
                              \glsxtrcombiningdiacriticIIIrules\string;
                              \glsxtrcombiningdiacriticIVrules
                             }
```

\glsxtrcombiningdiacriticIrules First set of combining diacritic marks.

```
                             \newcommand*{\glsxtrcombiningdiacriticIrules}{%
                              \glshex 0301\string;% combining acute
                              \glshex 0300\string;% combining grave
                              \glshex 0306\string;% combining breve
                              \glshex 0302\string;% combining circumflex
                              \glshex 030C\string;% combining caron
                              \glshex 030A\string;% combining ring
                              \glshex 030D\string;% combining vertical line above
                              \glshex 0308\string;% combining diaeresis
                              \glshex 030B\string;% combining double acute
                              \glshex 0303\string;% combining tilde
                              \glshex 0307\string;% combining dot above
                              \glshex 0304% combining macron
                             }
```

\glsxtrcombiningdiacriticIIrules Second set of combining diacritic marks.

```
                             \newcommand*{\glsxtrcombiningdiacriticIIrules}{%
                              \glshex 0337\string;% combining short solidus overlay
                              \glshex 0327\string;% combining cedilla
                              \glshex 0328\string;% combining ogonek
                              \glshex 0323\string;% combining dot below
                              \glshex 0332\string;% combining low line
```

```
                    \glshex 0305\string;% combining overline
                    \glshex 0309\string;% combining hook above
                    \glshex 030E\string;% combining double vertical line above
                    \glshex 030F\string;% combining double grave accent
                    \glshex 0310\string;% combining candrabindu
                    \glshex 0311\string;% combining inverted breve
                    \glshex 0312\string;% combining turned comma above
                    \glshex 0313\string;% combining comma above
                    \glshex 0314\string;% combining reversed comma above
                    \glshex 0315\string;% combining comma above right
                    \glshex 0316\string;% combining grave accent below
                    \glshex 0317% combining acute accent below
                  }
```

\glsxtrcombiningdiacriticIIIrules Third set of combining diacritic marks.

```
                  \newcommand*{\glsxtrcombiningdiacriticIIIrules}{%
                  \glshex 0318\string;% combining left tack below
                  \glshex 0319\string;% combining right tack below
                  \glshex 031A\string;% combining left angle above
                  \glshex 031B\string;% combining horn
                  \glshex 031C\string;% combining left half ring below
                  \glshex 031D\string;% combining up tack below
                  \glshex 031E\string;% combining down tack below
                  \glshex 031F\string;% combining plus sign below
                  \glshex 0320\string;% combining minus sign below
                  \glshex 0321\string;% combining palatalized hook below
                  \glshex 0322\string;% combining retroflex hook below
                  \glshex 0324\string;% combining diaresis below
                  \glshex 0325\string;% combining ring below
                  \glshex 0326\string;% combining comma below
                  \glshex 0329\string;% combining vertical line below
                  \glshex 032A\string;% combining bridge below
                  \glshex 032B\string;% combining inverted double arch below
                  \glshex 032C\string;% combining caron below
                  \glshex 032D\string;% combining circumflex accent below
                  \glshex 032E\string;% combining breve below
                  \glshex 032F\string;% combining inverted breve below
                  \glshex 0330\string;% combining tilde below
                  \glshex 0331\string;% combining macron below
                  \glshex 0333\string;% combining double low line
                  \glshex 0334\string;% combining tilde overlay
                  \glshex 0335\string;% combining short stroke overlay
                  \glshex 0336\string;% combining long stroke overlay
                  \glshex 0338\string;% combining long solidus overlay
                  \glshex 0339\string;% combining combining right half ring below
                  \glshex 033A\string;% combining inverted bridge below
                  \glshex 033B\string;% combining square below
                  \glshex 033C\string;% combining seagull below
                  \glshex 033D\string;% combining x above
                  \glshex 033E\string;% combining vertical tilde
```

```
                    \glshex 033F\string;% combining double overline
                    \glshex 0342\string;% combining Greek perispomeni
                    \glshex 0344\string;% combining Greek dialytika tonos
                    \glshex 0345\string;% combining Greek ypogegrammeni
                    \glshex 0360\string;% combining double tilde
                    \glshex 0361\string;% combining double inverted breve
                    \glshex 0483\string;% combining Cyrillic titlo
                    \glshex 0484\string;% combining Cyrillic palatalization
                    \glshex 0485\string;% combining Cyrillic dasia pneumata
                    \glshex 0486% combining Cyrillic psili pneumata
                  }
```

trcombiningdiacriticIVrules Fourth set of combining diacritic marks.

```
                  \newcommand*{\glsxtrcombiningdiacriticIVrules}{%
                  \glshex 20D0\string;% combining left harpoon above
                  \glshex 20D1\string;% combining right harpoon above
                  \glshex 20D2\string;% combining long vertical line overlay
                  \glshex 20D3\string;% combining short vertical line overlay
                  \glshex 20D4\string;% combining anticlockwise arrow above
                  \glshex 20D5\string;% combining clockwise arrow above
                  \glshex 20D6\string;% combining left arrow above
                  \glshex 20D7\string;% combining right arrow above
                  \glshex 20D8\string;% combining ring overlay
                  \glshex 20D9\string;% combining clockwise ring overlay
                  \glshex 20DA\string;% combining anticlockwise ring overlay
                  \glshex 20DB\string;% combining three dots above
                  \glshex 20DC\string;% combining four dots above
                  \glshex 20DD\string;% combining enclosing circle
                  \glshex 20DE\string;% combining enclosing square
                  \glshex 20DF\string;% combining enclosing diamond
                  \glshex 20E0\string;% combining enclosing circle backslash
                  \glshex 20E1%  combining left right arrow above
                  }
```

\glsxtrhyphenrules Hyphens.

```
                  \newcommand*{\glsxtrhyphenrules}{%
                  \glsxtrhyphenIrules\string;% hyphens
                  \glsxtrminusrules% minus signs
                  }
```

\glsxtrhyphenIrules Textual hyphens.

```
                  \newcommand*{\glsxtrhyphenIrules}{%
                  \string'\string-\string'\string;% ASCII hyphen
                  \glshex 00AD\string;% soft hyphen
                  \glshex 2010\string;% hyphen
                  \glshex 2011\string;% non-breaking hyphen
                  \glshex 2012\string;% figure dash
                  \glshex 2013\string;% en dash
                  \glshex 2014\string;% em dash
```

```
\glshex 2015% horizontal bar
}
```

\glsxtrhyphenIIrules   Alternative rule for textual hyphens.

```
\newcommand*{\glsxtrhyphenIIrules}{%
\string'\string-\string'% ASCII hyphen
\string<\glshex 00AD% soft hyphen
\string<\glshex 2010% hyphen
\string<\glshex 2011% non-breaking hyphen
\string<\glshex 2012% figure dash
\string<\glshex 2013% en dash
\string<\glshex 2014% em dash
\string<\glshex 2015% horizontal bar
}
```

\glsxtrminusrules   Minus signs.

```
\newcommand*{\glsxtrminusrules}{%
\glshex 2212\string=\glshex 207B\string=\glshex 208B% minus sign
}
```

\glsxtrgeneralpuncrules   General punctuation.

```
\newcommand*{\glsxtrgeneralpuncrules}{%
  \glsxtrgeneralpuncIrules
  \string<\glsxtrcurrencyrules
  \string<\glsxtrgeneralpuncIIrules
}
```

\glsxtrgeneralpuncIrules   First set of general punctuation.

```
\newcommand*{\glsxtrgeneralpuncIrules}{%
\glsxtrgeneralpuncmarksrules
\string<\glsxtrgeneralpuncaccentsrules
\string<\glsxtrgeneralpuncquoterules
\string<\glsxtrgeneralpuncbracketrules
\string<\glsxtrgeneralpuncsignrules
}
```

\glsxtrgeneralpuncmarksrules   Punctuation marks subset.

```
\newcommand*{\glsxtrgeneralpuncmarksrules}{%
\string'\glshex 005F\string'% underscore
\string<\glshex 00AF% macron
\string<\string'\glshex 002C\string'% comma
\string<\string'\glshex 003B\string'% semi-colon
\string<\string'\glshex 003A\string'% colon
\string<\string'\glshex 0021\string'% exclamation mark
\string<\glshex 00A1% inverted exclamation mark
\string<\string'\glshex 003F\string'% question mark
\string<\glshex 00BF% inverted question mark
\string<\string'\glshex 002F\string'% solidus
\string<\string'\glshex 002E\string'% full stop
}
```

`\glsxtrgeneralpuncdotrules` Punctuation marks subset: dots.

```
\newcommand*\glsxtrgeneralpuncdotrules{%
  \glshex 2024% one dot leader
  \string<\glshex 2025% two dot leader
  \string<\glshex 2026% horizontal ellipsis
  \string<\glshex 204F% reversed semicolon
  \string<\glshex 205A% vertical two dots
  \string<\glshex 205D% vertical three dots
  \string<\glshex 205E% vertical four dots
  \string<\glshex 2056% three dot punctuation
  \string<\glshex 2058% four dot punctuation
  \string<\glshex 2059% five dot punctuation
  \string<\glshex 205B% four dot mark
  \string<\glshex 203B% reference mark
  \string<\glshex 203C% dotted cross
}
```

`\glsxtrgeneralpuncaccentsrules` Punctuation marks subset: accent characters.

```
\newcommand*{\glsxtrgeneralpuncaccentsrules}{%
\glshex 00B4% acute accent
\string<\string'\glshex 0060\string'% grave accent
\string<\string'\glshex 005E\string'% circumflex accent
\string<\glshex 00A8% diaersis
\string<\string'\glshex 007E\string'% tilde
\string<\glshex 00B7% middle dot
\string<\glshex 00B8% cedilla
}
```

`\glsxtrgeneralpuncquoterules` Punctuation marks subset: quotes.

```
\newcommand*{\glsxtrgeneralpuncquoterules}{%
\string'\glshex 0027\string'% straight apostrophe
\string<\string'\glshex 0022\string'% straight double quote
\string<\glshex 00AB% left guillemet
\string<\glshex 00BB% right guillemet
}
```

`\glsxtrgeneralpuncbracketrules` Punctuation marks subset: brackets. May be redefined to include extra bracket subsets.

```
\newcommand*{\glsxtrgeneralpuncbracketrules}{%
\glsxtrgeneralpuncbracketIrules
}
```

`\glsxtrgeneralpuncbracketIrules` First set of bracket rules (general brackets).

```
\newcommand*{\glsxtrgeneralpuncbracketIrules}{%
\string'\glshex 0028\string'% left parenthesis
\string=\glshex 207D\string=\glshex 208D% super/subscript left parenthesis
\string<\string'\glshex 0029\string'% right parenthesis
\string=\glshex 207E\string=\glshex 208E% super/subscript right parenthesis
\string<\string'\glshex 005B\string'% left square bracket
```

633

```
                          \string<\string'\glshex 005D\string'% right square bracket
                          \string<\string'\glshex 007B\string'% left curly bracket
                          \string<\string'\glshex 007D\string'% right curly bracket
                        v1.56 added:
                          \string<\glshex 2045% left square bracket with quill
                          \string<\glshex 2046% right square bracket with quill
                          \string<\glshex 2329% left angle bracket
                          \string<\glshex 232A% right angle bracket
                         }
```

trgeneralpuncbracketIIrules Second set of bracket rules (miscellaneous mathematical symbols-A).

```
                        \newcommand*\glsxtrgeneralpuncbracketIIrules{%
                        \glshex 27E6% mathematical left white square bracket
                        \string<\glshex 27E7% mathematical right white square bracket
                        \string<\glshex 27E8% mathematical left angle bracket
                        \string<\glshex 27E9% mathematical right angle bracket
                        \string<\glshex 27EA% mathematical left chevron bracket
                        \string<\glshex 27EB% mathematical right chevron bracket
                        \string<\glshex 27EC% mathematical left white tortoise shell bracket
                        \string<\glshex 27ED% mathematical right white tortoise shell bracket
                        \string<\glshex 27EE% mathematical left flattened parenthesis
                        \string<\glshex 27EF% mathematical right flattened parenthesis
                         }
```

rgeneralpuncbracketIIIrules Third set of bracket rules (miscellaneous mathematical symbols-B).

```
                        \newcommand*\glsxtrgeneralpuncbracketIIIrules{%
                        \glshex 2983% left white curly bracket
                        \string<\glshex 2984% right white curly bracket
                        \string<\glshex 2985% left white parenthesis
                        \string<\glshex 2986% right white parenthesis
                        \string<\glshex 2987% left image bracket
                        \string<\glshex 2988% right image bracket
                        \string<\glshex 2989% left binding bracket
                        \string<\glshex 298A% right binding bracket
                        \string<\glshex 298B% left square bracket with underbar
                        \string<\glshex 298C% right square bracket with underbar
                        \string<\glshex 298D% left square bracket with top tick
                        \string<\glshex 298E% right square bracket with bottom tick
                        \string<\glshex 298F% left square bracket with bottom tick
                        \string<\glshex 2990% right square bracket with top tick
                        \string<\glshex 2991% left angle bracket with dot
                        \string<\glshex 2992% right angle bracket with dot
                        \string<\glshex 2993% left arc less-than bracket
                        \string<\glshex 2994% right arc greater-than bracket
                        \string<\glshex 2995% double left arc less-than bracket
                        \string<\glshex 2996% double right arc greater-than bracket
                        \string<\glshex 2997% left black tortoise shell bracket
                        \string<\glshex 2998% right black tortoise shell bracket
                        \string<\glshex 29FC% left curved angle bracket
```

```
                                   \string<\glshex 29FD% right curved angle bracket
                                   }
```

\glsxtrgeneralpuncbracketIVrules Fourth set of bracket rules (dingbat brackets). Quotation marks not included.
They should go with other quote marks,

```
\newcommand*\glsxtrgeneralpuncbracketIVrules{%
\glshex 2768% medium left parenthesis ornament
\string<\glshex 2769% medium right parenthesis ornament
\string<\glshex 276A% medium flattened left parenthesis ornament
\string<\glshex 276B% medium flattened right parenthesis ornament
\string<\glshex 276C% medium left angle bracket ornament
\string<\glshex 276D% medium right angle bracket ornament
\string<\glshex 2770% heavy left angle bracket ornament
\string<\glshex 2771% heavy right angle bracket ornament
\string<\glshex 2772% light left tortoise shell bracket ornament
\string<\glshex 2773% light right tortoise shell bracket ornament
\string<\glshex 2774% medium left curly bracket ornament
\string<\glshex 2775% medium right curly bracket ornament
}
```

\glsxtrgeneralpuncsignrules Punctuation marks subset: signs.

```
\newcommand*{\glsxtrgeneralpuncsignrules}{%
\glshex 00A7% section sign
\string<\glshex 00B6% pilcrow sign
\string<\glshex 00A9% copyright sign
\string<\glshex 00AE% registered sign
\string<\string'\glshex 0040\string'% at sign
}
```

\glsxtrcurrencyrules General punctuation.

```
\newcommand*{\glsxtrcurrencyrules}{%
\glshex 00A4% currency sign
\string<\glshex 0E3F% Thai currency symbol baht
\string<\glshex 00A2% cent sign
\string<\glshex 20A1% colon sign
\string<\glshex 20A2% cruzeiro sign
\string<\string'\glshex 0024\string'% dollar sign
\string<\glshex 20AB% dong sign
\string<\glshex 20AC% euro sign
\string<\glshex 20A3% French franc sign
\string<\glshex 20A4% lira sign
\string<\glshex 20A5% mill sign
\string<\glshex 20A6% naira sign
\string<\glshex 20A7% peseta sign
\string<\glshex 00A3% pound sign
\string<\glshex 20A8% rupee sign
\string<\glshex 20AA% new sheqel sign
\string<\glshex 20A9% won sign
\string<\glshex 00A5% yen sign
}
```

\glsxtrgeneralpuncIIrules Second set of general punctuation.

```
\newcommand*{\glsxtrgeneralpuncIIrules}{%
 \string'\glshex 002A\string'% asterisk
 \string<\string'\glshex 005C\string'% backslash
 \string<\string'\glshex 0026\string'% ampersand
 \string<\string'\glshex 0023\string'% hash sign
 \string<\string'\glshex 0025\string'% percent sign
 \string<\string'\glshex 002B\string'% plus sign
  \string=\glshex 207A\string=\glshex 208A% super/subscript plus sign
 \string<\glshex 00B1% plus-minus sign
 \string<\glshex 00F7% division sign
 \string<\glshex 00D7% multiplication sign
 \string<\string'\glshex 003C\string'% less-than sign
 \string<\string'\glshex 003D\string'% equals sign
 \string<\string'\glshex 003E\string'% greater-than sign
 \string<\glshex 00AC% not sign
 \string<\string'\glshex 007C\string'% vertical bar (pipe)
 \string<\glshex 00A6% broken bar
 \string<\glshex 00B0% degree sign
 \string<\glshex 00B5% micron sign
}
```

\glsxtrgeneralpuncIIIrules Alternative set of general punctuation.

```
\newcommand*{\glsxtrgeneralpuncIIIrules}{%
 \string'\glshex 002A\string'% asterisk
 \string<\string'\glshex 005C\string'% backslash
 \string<\string'\glshex 0026\string'% ampersand
 \string<\string'\glshex 0023\string'% hash sign
 \string<\string'\glshex 0025\string'% percent sign
 \string<\glsxtrhyphenIIrules % hyphens
 \string<\glshex 2052% commercial minus sign
 \string<\glsxtrminusrules % minus signs
 \string<\string'\glshex 002B\string'% plus sign
  \string=\glshex 207A\string=\glshex 208A% super/subscript plus sign
 \string<\glshex 00B1% plus-minus sign
 \string<\glshex 00F7% division sign
 \string<\glshex 00D7% multiplication sign
 \string<\string'\glshex 003C\string'% less-than sign
 \string<\string'\glshex 003D\string'% equals sign
 \string<\string'\glshex 003E\string'% greater-than sign
 \string<\glshex 00AC% not sign
 \string<\string'\glshex 007C\string'% vertical bar (pipe)
 \string<\glshex 00A6% broken bar
 \string<\glshex 00B0% degree sign
 \string<\glshex 00B5% micron sign
}
```

\glsxtrGeneralLatinIrules Basic Latin alphabet.

```
\newcommand*{\glsxtrGeneralLatinIrules}{%
```

```
\glsxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glsxtrLatinL
\string<\glsxtrLatinM
\string<\glsxtrLatinN
\string<\glsxtrLatinO
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
\string<\glsxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z
}
```

\glsxtrGeneralLatinIIrules General Latin alphabet (eth between D and E, ß treated as SS).

```
\newcommand*{\glsxtrGeneralLatinIIrules}{%
\glsxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glsxtrLatinEth
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glsxtrLatinL
\string<\glsxtrLatinM
\string<\glsxtrLatinN
\string<\glsxtrLatinO
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
```

```
\string& SS \string, \glsxtrLatinEszettSs
\string<\glsxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z%
}
```

\glsxtrGeneralLatinIIIrules General Latin alphabet (eth between D and E, ß treated as SZ).

```
\newcommand*{\glsxtrGeneralLatinIIIrules}{%
\glsxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glsxtrLatinEth
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glsxtrLatinL
\string<\glsxtrLatinM
\string<\glsxtrLatinN
\string<\glsxtrLatinO
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
\string& SZ , \glsxtrLatinEszettSz
\string<\glsxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z%
}
```

\glsxtrGeneralLatinIVrules General Latin alphabet (Æ treated as AE and Œtreated as OE, Þtreated as TH, ß treated as SS, eth between D and E).

```
\newcommand*{\glsxtrGeneralLatinIVrules}{%
\glsxtrLatinA
\string& AE , \glsxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<d,D%
```

638

```
                    \string<\glsxtrLatinEth
                    \string<\glsxtrLatinE
                    \string<f,F%
                    \string<g,G%
                    \string<\glsxtrLatinH
                    \string<\glsxtrLatinI
                    \string<j,J%
                    \string<\glsxtrLatinK
                    \string<\glsxtrLatinL
                    \string<\glsxtrLatinM
                    \string<\glsxtrLatinN
                    \string<\glsxtrLatinO
                    \string& OE , \glsxtrLatinOELigature
                    \string<\glsxtrLatinP
                    \string<q,Q%
                    \string<r,R%
                    \string<\glsxtrLatinS
                    \string& SS , \glsxtrLatinEszettSs
                    \string<\glsxtrLatinT
                    \string& th =\glshex 00DE
                    \string& TH =\glshex 00FE
                    \string<u,U%
                    \string<v,V%
                    \string<w,W%
                    \string<\glsxtrLatinX
                    \string<y,Y%
                    \string<z,Z%
                   }
```

\glsxtrGeneralLatinVrules General Latin alphabet (eth between D and E, ß treated as SS, Þ treated as
                           TH).

```
 \newcommand*{\glsxtrGeneralLatinVrules}{%
 \glsxtrLatinA
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinEth
 \string<\glsxtrLatinE
 \string<f,F%
 \string<g,G%
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string<\glsxtrLatinP
 \string<q,Q%
```

```
\string<r,R%
\string<\glsxtrLatinS
\string& SS , \glsxtrLatinEszettSs
\string<\glsxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z%
}
```

\glsxtrGeneralLatinVIrules General Latin alphabet (eth between D and E, ß treated as SZ, Þ treated as TH).

```
\newcommand*{\glsxtrGeneralLatinVIrules}{%
\glsxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glsxtrLatinEth
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glsxtrLatinL
\string<\glsxtrLatinM
\string<\glsxtrLatinN
\string<\glsxtrLatinO
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
\string& SZ , \glsxtrLatinEszettSz
\string<\glsxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z%
}
```

`\glsxtrGeneralLatinVIIrules` General Latin alphabet (Æ between A and B, eth between D and E, insular G as G, Œ between O and P, long S equivalent to S, Þ between T and U and wynn as W).

```
\newcommand*{\glsxtrGeneralLatinVIIrules}{%
\glsxtrLatinA
\string<\glsxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glsxtrLatinEth
\string<\glsxtrLatinE
\string<f,F%
\string<\glsxtrLatinInsularG
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glsxtrLatinL
\string<\glsxtrLatinM
\string<\glsxtrLatinN
\string<\glsxtrLatinO
\string<\glsxtrLatinOELigature
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glshex 017F=\glsxtrLatinS % s and long s
\string<\glsxtrLatinT
\string<\glsxtrLatinThorn
\string<u,U%
\string<v,V%
\string< w\string=\glshex 01BF, W\string=\glshex 01F7
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z%
}
```

`\glsxtrGeneralLatinVIIIrules` General Latin alphabet (Æ treated as AE and Œtreated as OE, Þtreated as TH, ß treated as SS, eth treated as D, Ø treated as O, Ł treated as L).

```
\newcommand*{\glsxtrGeneralLatinVIIIrules}{%
\glsxtrLatinA
\string& AE , \glsxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<\glshex 00F0\string;d,\glshex 00D0\string;D% D and eth
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
```

```
\string<j,J%
\string<\glsxtrLatinK
\string<\glshex 0142\string=\glsxtrLatinL\string=\glshex 0141% L and \L
\string<\glsxtrLatinM
\string<\glsxtrLatinN
\string<\glshex 00F8\string=\glsxtrLatinO\string=\glshex 00D8% O and \O
\string& OE , \glsxtrLatinOELigature
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
\string& SS , \glsxtrLatinEszettSs
\string<\glsxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z%
}
```

Fragments.

Basic Latin alphabet A–M.

```
\newcommand*{\glsxtrGeneralLatinAtoMrules}{%
\glsxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glsxtrLatinL
\string<\glsxtrLatinM
}
```

Basic Latin alphabet N–Z.

```
\newcommand*{\glsxtrGeneralLatinNtoZrules}{%
\string<\glsxtrLatinN
\string<\glsxtrLatinO
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
```

```
\string<\glsxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z
}
```

\glsxtrGeneralLatinAtoGrules Basic Latin alphabet A–G.

```
\newcommand*{\glsxtrGeneralLatinAtoGrules}{%
\glsxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
}
```

\glsxtrGeneralLatinHtoMrules Basic Latin alphabet H–M.

```
\newcommand*{\glsxtrGeneralLatinHtoMrules}{%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glsxtrLatinL
\string<\glsxtrLatinM
}
```

\glsxtrGeneralLatinNtoSrules Basic Latin alphabet N–S.

```
\newcommand*{\glsxtrGeneralLatinNtoSrules}{%
\string<\glsxtrLatinN
\string<\glsxtrLatinO
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
}
```

\glsxtrGeneralLatinTtoZrules Basic Latin alphabet T–Z.

```
\newcommand*{\glsxtrGeneralLatinTtoZrules}{%
\string<\glsxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z
}
```

\glsxtrLatinA

```
\newcommand*{\glsxtrLatinA}{%
  a\string=\glshex 00AA\string=\glshex 2090,A
}
```

\glsxtrLatinE

```
\newcommand*{\glsxtrLatinE}{%
  e\string=\glshex 2091,E
}
```

\glsxtrLatinH

```
\newcommand*{\glsxtrLatinH}{%
  h\string=\glshex 2095,H
}
```

\glsxtrLatinI

```
\newcommand*{\glsxtrLatinI}{%
  i\string=\glshex 2071,I
}
```

\glsxtrLatinK

```
\newcommand*{\glsxtrLatinK}{%
  k\string=\glshex 2096,K
}
```

\glsxtrLatinL

```
\newcommand*{\glsxtrLatinL}{%
  l\string=\glshex 2097,L
}
```

\glsxtrLatinM

```
\newcommand*{\glsxtrLatinM}{%
  m\string=\glshex 2098,M
}
```

\glsxtrLatinN

```
\newcommand*{\glsxtrLatinN}{%
  n\string=\glshex 207F\string=\glshex 2099,N
}
```

\glsxtrLatinO

```
\newcommand*{\glsxtrLatinO}{%
  o\string=\glshex 00BA\string=\glshex 2092,O
}
```

\glsxtrLatinP

```
\newcommand*{\glsxtrLatinP}{%
  p\string=\glshex 209A,P
}
```

\glsxtrLatinS

```
\newcommand*{\glsxtrLatinS}{%
  s\string=\glshex 209B,S
}
```

\glsxtrLatinT

```
\newcommand*{\glsxtrLatinT}{%
  t\string=\glshex 209C,T
}
```

\glsxtrLatinX

```
\newcommand*{\glsxtrLatinX}{%
  x\string=\glshex 2093,X
}
```

\glsxtrLatinSchwa   Latin schwa (lower case, subscript and upper case).

```
\newcommand*{\glsxtrLatinSchwa}{%
  \glshex 0259\string=\glshex 2094,\glshex 018F
}
```

\glsxtrLatinEszettSs   SS=ss

```
\newcommand*{\glsxtrLatinEszettSs}{%
 \glshex 00DF% eszett
 \string=\glshex 017Fs % "long S"s
}
```

\glsxtrLatinEszettSz   SS=sz

```
\newcommand*{\glsxtrLatinEszettSz}{%
 \glshex 00DF% eszett
 \string= \glshex 017Fz % "long S"z
}
```

\glsxtrLatinEth

```
\newcommand*{\glsxtrLatinEth}{%
 \glshex 00F0,\glshex 00D0% eth
}
```

\glsxtrLatinThorn

```
\newcommand*{\glsxtrLatinThorn}{%
 \glshex 00FE,\glshex 00DE% thorn
}
```

\glsxtrLatinAELigature

```
\newcommand*{\glsxtrLatinAELigature}{%
 \glshex 00E6,\glshex 00C6% AE-ligature
}
```

`\glsxtrLatinOELigature`

```
\newcommand*{\glsxtrLatinOELigature}{%
 \glshex 0153,\glshex 0152% OE-ligature
}
```

`\glsxtrLatinAA`

```
\newcommand*{\glsxtrLatinAA}{%
 \glshex 00E5=a\glshex 030A,% \aa
 \glshex 00C5=A\glshex 030A% \AA
}
```

`\glsxtrLatinWynn`

```
\newcommand*{\glsxtrLatinWynn}{%
 \glshex 01BF,\glshex 01F7% wynn
}
```

`\glsxtrLatinInsularG`

```
\newcommand*{\glsxtrLatinInsularG}{%
\glshex 1D79,\glshex A77D% insular G
 \string; g, G
}
```

`\glsxtrLatinOslash`

```
\newcommand*{\glsxtrLatinOslash}{%
 \glshex 00F8,\glshex 00D8% \o, \O
}
```

`\glsxtrLatinLslash`

```
\newcommand*{\glsxtrLatinLslash}{%
 \glshex 0142,\glshex 0141% \l, \L
}
```

`\glsxtrMathUpGreekIrules` Includes digamma between epsilon and zeta.

```
\newcommand*{\glsxtrMathUpGreekIrules}{%
 \glsxtrUpAlpha
 \string<\glsxtrUpBeta
 \string<\glsxtrUpGamma
 \string<\glsxtrUpDelta
 \string<\glsxtrUpEpsilon
 \string<\glsxtrUpDigamma
 \string<\glsxtrUpZeta
 \string<\glsxtrUpEta
 \string<\glsxtrUpTheta
 \string<\glsxtrUpIota
 \string<\glsxtrUpKappa
 \string<\glsxtrUpLambda
 \string<\glsxtrUpMu
 \string<\glsxtrUpNu
 \string<\glsxtrUpXi
```

```
                      \string<\glsxtrUpOmicron
                      \string<\glsxtrUpPi
                      \string<\glsxtrUpRho
                      \string<\glsxtrUpSigma
                      \string<\glsxtrUpTau
                      \string<\glsxtrUpUpsilon
                      \string<\glsxtrUpPhi
                      \string<\glsxtrUpChi
                      \string<\glsxtrUpPsi
                      \string<\glsxtrUpOmega
                    }
```

\glsxtrMathUpGreekIIrules Doesn't include digamma.

```
                    \newcommand*{\glsxtrMathUpGreekIIrules}{%
                    \glsxtrUpAlpha
                    \string<\glsxtrUpBeta
                    \string<\glsxtrUpGamma
                    \string<\glsxtrUpDelta
                    \string<\glsxtrUpEpsilon
                    \string<\glsxtrUpZeta
                    \string<\glsxtrUpEta
                    \string<\glsxtrUpTheta
                    \string<\glsxtrUpIota
                    \string<\glsxtrUpKappa
                    \string<\glsxtrUpLambda
                    \string<\glsxtrUpMu
                    \string<\glsxtrUpNu
                    \string<\glsxtrUpXi
                    \string<\glsxtrUpOmicron
                    \string<\glsxtrUpPi
                    \string<\glsxtrUpRho
                    \string<\glsxtrUpSigma
                    \string<\glsxtrUpTau
                    \string<\glsxtrUpUpsilon
                    \string<\glsxtrUpPhi
                    \string<\glsxtrUpChi
                    \string<\glsxtrUpPsi
                    \string<\glsxtrUpOmega
                    }
```

\glsxtrMathItalicGreekIrules Includes (upright) digamma between epsilon and zeta (there isn't an italic
digamma), so don't mix with \glsxtrMathUpGreekIrules or there may be
unexpected results.

```
                    \newcommand*{\glsxtrMathItalicGreekIrules}{%
                    \glsxtrMathItalicAlpha
                    \string<\glsxtrMathItalicBeta
                    \string<\glsxtrMathItalicGamma
                    \string<\glsxtrMathItalicDelta
                    \string<\glsxtrMathItalicEpsilon
```

```
                              \string<\glsxtrUpDigamma
                              \string<\glsxtrMathItalicZeta
                              \string<\glsxtrMathItalicEta
                              \string<\glsxtrMathItalicTheta
                              \string<\glsxtrMathItalicIota
                              \string<\glsxtrMathItalicKappa
                              \string<\glsxtrMathItalicLambda
                              \string<\glsxtrMathItalicMu
                              \string<\glsxtrMathItalicNu
                              \string<\glsxtrMathItalicXi
                              \string<\glsxtrMathItalicOmicron
                              \string<\glsxtrMathItalicPi
                              \string<\glsxtrMathItalicRho
                              \string<\glsxtrMathItalicSigma
                              \string<\glsxtrMathItalicTau
                              \string<\glsxtrMathItalicUpsilon
                              \string<\glsxtrMathItalicPhi
                              \string<\glsxtrMathItalicChi
                              \string<\glsxtrMathItalicPsi
                              \string<\glsxtrMathItalicOmega
                            }
```

glsxtrMathItalicGreekIIrules Doesn't include digamma.

```
                            \newcommand*{\glsxtrMathItalicGreekIIrules}{%
                            \glsxtrMathItalicAlpha
                            \string<\glsxtrMathItalicBeta
                            \string<\glsxtrMathItalicGamma
                            \string<\glsxtrMathItalicDelta
                            \string<\glsxtrMathItalicEpsilon
                            \string<\glsxtrMathItalicZeta
                            \string<\glsxtrMathItalicEta
                            \string<\glsxtrMathItalicTheta
                            \string<\glsxtrMathItalicIota
                            \string<\glsxtrMathItalicKappa
                            \string<\glsxtrMathItalicLambda
                            \string<\glsxtrMathItalicMu
                            \string<\glsxtrMathItalicNu
                            \string<\glsxtrMathItalicXi
                            \string<\glsxtrMathItalicOmicron
                            \string<\glsxtrMathItalicPi
                            \string<\glsxtrMathItalicRho
                            \string<\glsxtrMathItalicSigma
                            \string<\glsxtrMathItalicTau
                            \string<\glsxtrMathItalicUpsilon
                            \string<\glsxtrMathItalicPhi
                            \string<\glsxtrMathItalicChi
                            \string<\glsxtrMathItalicPsi
                            \string<\glsxtrMathItalicOmega
                            }
```

\glsxtrMathItalicUpperGreekIrules Upper case only (includes upright digamma).

```
\newcommand*{\glsxtrMathItalicUpperGreekIrules}{%
\glshex 1D6E2% upper case alpha (maths italic)
\string<\glshex 1D6E3% upper case beta (maths italic)
\string<\glshex 1D6E4% upper case gamma (maths italic)
\string<\glshex 1D6E5% upper case delta (maths italic)
\string<\glshex 1D6E6% upper case epsilon (maths italic)
\string<\glshex 03DC% upper case digamma
\string<\glshex 1D6E7% upper case zeta (maths italic)
\string<\glshex 1D6E8% upper case eta (maths italic)
\string<\glshex 1D6E9% upper case theta (maths italic)
\string=\glshex 1D6F3% upper case theta variant (maths italic)
\string<\glshex 1D6EA% upper case iota (maths italic)
\string<\glshex 1D6EB% upper case kappa (maths italic)
\string<\glshex 1D6EC% upper case lambda (maths italic)
\string<\glshex 1D6ED% upper case mu (maths italic)
\string<\glshex 1D6EE% upper case nu (maths italic)
\string<\glshex 1D6EF% upper case xi (maths italic)
\string<\glshex 1D6F0% upper case omicron (maths italic)
\string<\glshex 1D6F1% upper case pi (maths italic)
\string<\glshex 1D6F2% upper case rho (maths italic)
\string<\glshex 1D6F4% upper case sigma (maths italic)
\string<\glshex 1D6F5% upper case tau (maths italic)
\string<\glshex 1D6F6% upper case upsilon (maths italic)
\string<\glshex 1D6F7% upper case phi (maths italic)
\string<\glshex 1D6F8% upper case chi (maths italic)
\string<\glshex 1D6F9% upper case psi (maths italic)
\string<\glshex 1D6FA% upper case omega (maths italic)
}
```

\glsxtrMathItalicUpperGreekIIrules Upper case only (doesn't include upright digamma).

```
\newcommand*{\glsxtrMathItalicUpperGreekIIrules}{%
\glshex 1D6E2% upper case alpha (maths italic)
\string<\glshex 1D6E3% upper case beta (maths italic)
\string<\glshex 1D6E4% upper case gamma (maths italic)
\string<\glshex 1D6E5% upper case delta (maths italic)
\string<\glshex 1D6E6% upper case epsilon (maths italic)
\string<\glshex 1D6E7% upper case zeta (maths italic)
\string<\glshex 1D6E8% upper case eta (maths italic)
\string<\glshex 1D6E9% upper case theta (maths italic)
\string=\glshex 1D6F3% upper case theta variant (maths italic)
\string<\glshex 1D6EA% upper case iota (maths italic)
\string<\glshex 1D6EB% upper case kappa (maths italic)
\string<\glshex 1D6EC% upper case lambda (maths italic)
\string<\glshex 1D6ED% upper case mu (maths italic)
\string<\glshex 1D6EE% upper case nu (maths italic)
\string<\glshex 1D6EF% upper case xi (maths italic)
\string<\glshex 1D6F0% upper case omicron (maths italic)
\string<\glshex 1D6F1% upper case pi (maths italic)
\string<\glshex 1D6F2% upper case rho (maths italic)
```

```
                                    \string<\glshex 1D6F4% upper case sigma (maths italic)
                                    \string<\glshex 1D6F5% upper case tau (maths italic)
                                    \string<\glshex 1D6F6% upper case upsilon (maths italic)
                                    \string<\glshex 1D6F7% upper case phi (maths italic)
                                    \string<\glshex 1D6F8% upper case chi (maths italic)
                                    \string<\glshex 1D6F9% upper case psi (maths italic)
                                    \string<\glshex 1D6FA% upper case omega (maths italic)
                                  }
```

rMathItalicLowerGreekIrules Lower case only (includes upright digamma).

```
                                  \newcommand*{\glsxtrMathItalicLowerGreekIrules}{%
                                  \glshex 1D6FC% lower case alpha (maths italic)
                                  \string<\glshex 1D6FD% lower case beta (maths italic)
                                  \string<\glshex 1D6FE% lower case gamma (maths italic)
                                  \string<\glshex 1D6FF% lower case delta (maths italic)
                                  \string<\glshex 1D700% lower case epsilon (maths italic)
                                  \string=\glshex 1D716% lower case epsilon variant (maths italic)
                                  \string<\glshex 03DD% lower case digamma
                                  \string<\glshex 1D701% lower case zeta (maths italic)
                                  \string<\glshex 1D702% lower case eta (maths italic)
                                  \string<\glshex 1D703% lower case theta (maths italic)
                                  \string=\glshex 1D717% lower case theta variant (maths italic)
                                  \string<\glshex 1D704% lower case iota (maths italic)
                                  \string<\glshex 1D705% lower case kappa (maths italic)
                                  \string=\glshex 1D718% lower case kappa variant (maths italic)
                                  \string<\glshex 1D706% lower case lambda (maths italic)
                                  \string<\glshex 1D707% lower case mu (maths italic)
                                  \string<\glshex 1D708% lower case nu (maths italic)
                                  \string<\glshex 1D709% lower case xi (maths italic)
                                  \string<\glshex 1D70A% lower case omicron (maths italic)
                                  \string<\glshex 1D70B% lower case pi (maths italic)
                                  \string=\glshex 1D71B% lower case pi variant (maths italic)
                                  \string<\glshex 1D70C% lower case rho (maths italic)
                                  \string=\glshex 1D71A% lower case rho variant (maths italic)
                                  \string<\glshex 1D70D% lower case final sigma (maths italic)
                                  \string=\glshex 1D70E% lower case sigma (maths italic)
                                  \string<\glshex 1D70F% lower case tau (maths italic)
                                  \string<\glshex 1D710% lower case upsilon (maths italic)
                                  \string<\glshex 1D711% lower case phi (maths italic)
                                  \string=\glshex 1D719% lower case phi variant (maths italic)
                                  \string<\glshex 1D712% lower case chi (maths italic)
                                  \string<\glshex 1D713% lower case psi (maths italic)
                                  \string<\glshex 1D714% lower case omega (maths italic)
                                  }
```

MathItalicLowerGreekIIrules Lower case only (doesn't includes upright digamma).

```
                                  \newcommand*{\glsxtrMathItalicLowerGreekIIrules}{%
                                  \glshex 1D6FC% lower case alpha (maths italic)
                                  \string<\glshex 1D6FD% lower case beta (maths italic)
                                  \string<\glshex 1D6FE% lower case gamma (maths italic)
```

```
                    \string<\glshex 1D6FF% lower case delta (maths italic)
                    \string<\glshex 1D700% lower case epsilon (maths italic)
                    \string=\glshex 1D716% lower case epsilon variant (maths italic)
                    \string<\glshex 1D701% lower case zeta (maths italic)
                    \string<\glshex 1D702% lower case eta (maths italic)
                    \string<\glshex 1D703% lower case theta (maths italic)
                    \string=\glshex 1D717% lower case theta variant (maths italic)
                    \string<\glshex 1D704% lower case iota (maths italic)
                    \string<\glshex 1D705% lower case kappa (maths italic)
                    \string=\glshex 1D718% lower case kappa variant (maths italic)
                    \string<\glshex 1D706% lower case lambda (maths italic)
                    \string<\glshex 1D707% lower case mu (maths italic)
                    \string<\glshex 1D708% lower case nu (maths italic)
                    \string<\glshex 1D709% lower case xi (maths italic)
                    \string<\glshex 1D70A% lower case omicron (maths italic)
                    \string<\glshex 1D70B% lower case pi (maths italic)
                    \string=\glshex 1D71B% lower case pi variant (maths italic)
                    \string<\glshex 1D70C% lower case rho (maths italic)
                    \string=\glshex 1D71A% lower case rho variant (maths italic)
                    \string<\glshex 1D70D% lower case final sigma (maths italic)
                    \string=\glshex 1D70E% lower case sigma (maths italic)
                    \string<\glshex 1D70F% lower case tau (maths italic)
                    \string<\glshex 1D710% lower case upsilon (maths italic)
                    \string<\glshex 1D711% lower case phi (maths italic)
                    \string=\glshex 1D719% lower case phi variant (maths italic)
                    \string<\glshex 1D712% lower case chi (maths italic)
                    \string<\glshex 1D713% lower case psi (maths italic)
                    \string<\glshex 1D714% lower case omega (maths italic)
                   }
```

`\glsxtrMathGreekIrules` Includes both upright and italic with digamma between epsilon and zeta.

```
                   \newcommand*{\glsxtrMathGreekIrules}{%
                   \glsxtrMathItalicAlpha
                   \string;\glsxtrUpAlpha
                   \string<\glsxtrMathItalicBeta
                   \string;\glsxtrUpBeta
                   \string<\glsxtrMathItalicGamma
                   \string;\glsxtrUpGamma
                   \string<\glsxtrMathItalicDelta
                   \string;\glsxtrUpDelta
                   \string<\glsxtrMathItalicEpsilon
                   \string;\glsxtrUpEpsilon
                   \string<\glsxtrUpDigamma
                   \string<\glsxtrMathItalicZeta
                   \string;\glsxtrUpZeta
                   \string<\glsxtrMathItalicEta
                   \string;\glsxtrUpEta
                   \string<\glsxtrMathItalicTheta
                   \string;\glsxtrUpTheta
                   \string<\glsxtrMathItalicIota
```

```
                    \string;\glsxtrUpIota
                    \string<\glsxtrMathItalicKappa
                    \string;\glsxtrUpKappa
                    \string<\glsxtrMathItalicLambda
                    \string;\glsxtrUpLambda
                    \string<\glsxtrMathItalicMu
                    \string;\glsxtrUpMu
                    \string<\glsxtrMathItalicNu
                    \string;\glsxtrUpNu
                    \string<\glsxtrMathItalicXi
                    \string;\glsxtrUpXi
                    \string<\glsxtrMathItalicOmicron
                    \string;\glsxtrUpOmicron
                    \string<\glsxtrMathItalicPi
                    \string;\glsxtrUpPi
                    \string<\glsxtrMathItalicRho
                    \string;\glsxtrUpRho
                    \string<\glsxtrMathItalicSigma
                    \string;\glsxtrUpSigma
                    \string<\glsxtrMathItalicTau
                    \string;\glsxtrUpTau
                    \string<\glsxtrMathItalicUpsilon
                    \string;\glsxtrUpUpsilon
                    \string<\glsxtrMathItalicPhi
                    \string;\glsxtrUpPhi
                    \string<\glsxtrMathItalicChi
                    \string;\glsxtrUpChi
                    \string<\glsxtrMathItalicPsi
                    \string;\glsxtrUpPsi
                    \string<\glsxtrMathItalicOmega
                    \string;\glsxtrUpOmega
                  }
```

\glsxtrMathGreekIIrules Includes both upright and italic (digamma not included).

```
                    \newcommand*{\glsxtrMathGreekIIrules}{%
                    \glsxtrMathItalicAlpha
                    \string;\glsxtrUpAlpha
                    \string<\glsxtrMathItalicBeta
                    \string;\glsxtrUpBeta
                    \string<\glsxtrMathItalicGamma
                    \string;\glsxtrUpGamma
                    \string<\glsxtrMathItalicDelta
                    \string;\glsxtrUpDelta
                    \string<\glsxtrMathItalicEpsilon
                    \string;\glsxtrUpEpsilon
                    \string<\glsxtrMathItalicZeta
                    \string;\glsxtrUpZeta
                    \string<\glsxtrMathItalicEta
                    \string;\glsxtrUpEta
                    \string<\glsxtrMathItalicTheta
```

```
              \string;\glsxtrUpTheta
              \string<\glsxtrMathItalicIota
              \string;\glsxtrUpIota
              \string<\glsxtrMathItalicKappa
              \string;\glsxtrUpKappa
              \string<\glsxtrMathItalicLambda
              \string;\glsxtrUpLambda
              \string<\glsxtrMathItalicMu
              \string;\glsxtrUpMu
              \string<\glsxtrMathItalicNu
              \string;\glsxtrUpNu
              \string<\glsxtrMathItalicXi
              \string;\glsxtrUpXi
              \string<\glsxtrMathItalicOmicron
              \string;\glsxtrUpOmicron
              \string<\glsxtrMathItalicPi
              \string;\glsxtrUpPi
              \string<\glsxtrMathItalicRho
              \string;\glsxtrUpRho
              \string<\glsxtrMathItalicSigma
              \string;\glsxtrUpSigma
              \string<\glsxtrMathItalicTau
              \string;\glsxtrUpTau
              \string<\glsxtrMathItalicUpsilon
              \string;\glsxtrUpUpsilon
              \string<\glsxtrMathItalicPhi
              \string;\glsxtrUpPhi
              \string<\glsxtrMathItalicChi
              \string;\glsxtrUpChi
              \string<\glsxtrMathItalicPsi
              \string;\glsxtrUpPsi
              \string<\glsxtrMathItalicOmega
              \string;\glsxtrUpOmega
             }
```

\glsxtrUpAlpha

```
              \newcommand*{\glsxtrUpAlpha}{%
               \glshex 03B1,% lower case alpha
               \glshex 0391% upper case alpha
              }
```

 \glsxtrUpBeta

```
              \newcommand*{\glsxtrUpBeta}{%
               \glshex 03B2,% lower case beta
               \glshex 0392% upper case beta
              }
```

\glsxtrUpGamma

```
              \newcommand*{\glsxtrUpGamma}{%
```

```
                 \glshex 03B3,% lower case gamma
                 \glshex 0393% upper case gamma
               }
```

\glsxtrUpDelta

```
               \newcommand*{\glsxtrUpDelta}{%
               \glshex 03B4,% lower case delta
               \glshex 0394% upper case delta
               }
```

\glsxtrUpEpsilon

```
               \newcommand*{\glsxtrUpEpsilon}{%
               \glshex 03B5% lower case epsilon
               \string=\glshex 03F5,% lower case epsilon variant
               \glshex 0395% upper case epsilon
               }
```

\glsxtrUpDigamma

```
               \newcommand*{\glsxtrUpDigamma}{%
               \glshex 03DD,% lower case digamma
               \glshex 03DC% upper case digamma
               }
```

\glsxtrUpZeta

```
               \newcommand*{\glsxtrUpZeta}{%
               \glshex 03B6,% lower case zeta
               \glshex 0396% upper case zeta
               }
```

\glsxtrUpEta

```
               \newcommand*{\glsxtrUpEta}{%
               \glshex 03B7,% lower case eta
               \glshex 0397% upper case eta
               }
```

\glsxtrUpTheta

```
               \newcommand*{\glsxtrUpTheta}{%
               \glshex 03B8% lower case theta
               \string=\glshex 03D1,% lower case theta variant
               \glshex 0398% upper case theta
               }
```

\glsxtrUpIota

```
               \newcommand*{\glsxtrUpIota}{%
               \glshex 03B9,% lower case iota
               \glshex 0399% upper case iota
               }
```

\glsxtrUpKappa

```
\newcommand*{\glsxtrUpKappa}{%
 \glshex 03BA% lower case kappa
 \string=\glshex 03F0,% lower case kappa variant
 \glshex 039A% upper case kappa
}
```

\glsxtrUpLambda

```
\newcommand*{\glsxtrUpLambda}{%
 \glshex 03BB,% lower lambda
 \glshex 039B% upper case lambda
}
```

\glsxtrUpMu

```
\newcommand*{\glsxtrUpMu}{%
 \glshex 03BC,% lower case mu
 \glshex 039C% upper case mu
}
```

\glsxtrUpNu

```
\newcommand*{\glsxtrUpNu}{%
 \glshex 03BD,% lower case nu
 \glshex 039D% upper case nu
}
```

\glsxtrUpXi

```
\newcommand*{\glsxtrUpXi}{%
 \glshex 03BE,% lower case xi
 \glshex 039E% upper case xi
}
```

\glsxtrUpOmicron

```
\newcommand*{\glsxtrUpOmicron}{%
 \glshex 03BF,% lower case omicron
 \glshex 039F% upper case omicron
}
```

\glsxtrUpPi

```
\newcommand*{\glsxtrUpPi}{%
 \glshex 03C0% lower case pi
 \string=\glshex 03D6,% lower case pi variant
 \glshex 03A0% upper case pi
}
```

\glsxtrUpRho

```
\newcommand*{\glsxtrUpRho}{%
 \glshex 03C1% lower case rho
 \string=\glshex 03F1,% lower case rho variant
 \glshex 03A1% upper case rho
}
```

`\glsxtrUpSigma`

```
\newcommand*{\glsxtrUpSigma}{%
 \glshex 03C2% lower case sigma
 \string=\glshex 03C3,% lower case sigma
 \glshex 03A3% upper case sigma
}
```

`\glsxtrUpTau`

```
\newcommand*{\glsxtrUpTau}{%
 \glshex 03C4,% lower case tau
 \glshex 03A4% upper case tau
}
```

`\glsxtrUpUpsilon`

```
\newcommand*{\glsxtrUpUpsilon}{%
 \glshex 03C5,% lower case upsilon
 \glshex 03A5% upper case upsilon
}
```

`\glsxtrUpPhi`

```
\newcommand*{\glsxtrUpPhi}{%
 \glshex 03C6% lower case phi
 \string=\glshex 03D5,% lower case phi variant
 \glshex 03A6% upper case phi
}
```

`\glsxtrUpChi`

```
\newcommand*{\glsxtrUpChi}{%
 \glshex 03C7,% lower case chi
 \glshex 03A7% upper case chi
}
```

`\glsxtrUpPsi`

```
\newcommand*{\glsxtrUpPsi}{%
 \glshex 03C8,% lower case psi
 \glshex 03A8% upper case psi
}
```

`\glsxtrUpOmega`

```
\newcommand*{\glsxtrUpOmega}{%
 \glshex 03C9,% lower case omega
 \glshex 03A9% upper case omega
}
```

`\glsxtrMathItalicAlpha`

```
\newcommand*{\glsxtrMathItalicAlpha}{%
 \glshex 1D6FC,% lower case alpha (maths italic)
 \glshex 1D6E2% upper case alpha (maths italic)
}
```

656

\glsxtrMathItalicBeta

```
\newcommand*{\glsxtrMathItalicBeta}{%
 \glshex 1D6FD,% lower case beta (maths italic)
 \glshex 1D6E3% upper case beta (maths italic)
}
```

\glsxtrMathItalicGamma

```
\newcommand*{\glsxtrMathItalicGamma}{%
 \glshex 1D6FE,% lower case gamma (maths italic)
 \glshex 1D6E4% upper case gamma (maths italic)
}
```

\glsxtrMathItalicDelta

```
\newcommand*{\glsxtrMathItalicDelta}{%
 \glshex 1D6FF,% lower case delta (maths italic)
 \glshex 1D6E5% upper case delta (maths italic)
}
```

\glsxtrMathItalicEpsilon

```
\newcommand*{\glsxtrMathItalicEpsilon}{%
 \glshex 1D700% lower case epsilon (maths italic)
 \string=\glshex 1D716,% lower case epsilon variant (maths italic)
 \glshex 1D6E6% upper case epsilon (maths italic)
}
```

\glsxtrMathItalicZeta

```
\newcommand*{\glsxtrMathItalicZeta}{%
 \glshex 1D701,% lower case zeta (maths italic)
 \glshex 1D6E7% upper case zeta (maths italic)
}
```

\glsxtrMathItalicEta

```
\newcommand*{\glsxtrMathItalicEta}{%
 \glshex 1D702,% lower case eta (maths italic)
 \glshex 1D6E8% upper case eta (maths italic)
}
```

\glsxtrMathItalicTheta

```
\newcommand*{\glsxtrMathItalicTheta}{%
 \glshex 1D703% lower case theta (maths italic)
 \string=\glshex 1D717,% lower case theta variant (maths italic)
 \glshex 1D6E9% upper case theta (maths italic)
 \string=\glshex 1D6F3% upper case theta variant (maths italic)
}
```

\glsxtrMathItalicIota

```
\newcommand*{\glsxtrMathItalicIota}{%
 \glshex 1D704,% lower case iota (maths italic)
 \glshex 1D6EA% upper case iota (maths italic)
}
```

\glsxtrMathItalicKappa

> \newcommand*{\glsxtrMathItalicKappa}{%
> \glshex 1D705% lower case kappa (maths italic)
> \string=\glshex 1D718,% lower case kappa variant (maths italic)
> \glshex 1D6EB% upper case kappa (maths italic)
> }

\glsxtrMathItalicLambda

> \newcommand*{\glsxtrMathItalicLambda}{%
> \glshex 1D706,% lower case lambda (maths italic)
> \glshex 1D6EC% upper case lambda (maths italic)
> }

\glsxtrMathItalicMu

> \newcommand*{\glsxtrMathItalicMu}{%
> \glshex 1D707,% lower case mu (maths italic)
> \glshex 1D6ED% upper case mu (maths italic)
> }

\glsxtrMathItalicNu

> \newcommand*{\glsxtrMathItalicNu}{%
> \glshex 1D708,% lower case nu (maths italic)
> \glshex 1D6EE% upper case nu (maths italic)
> }

\glsxtrMathItalicXi

> \newcommand*{\glsxtrMathItalicXi}{%
> \glshex 1D709,% lower case xi (maths italic)
> \glshex 1D6EF% upper case xi (maths italic)
> }

\glsxtrMathItalicOmicron

> \newcommand*{\glsxtrMathItalicOmicron}{%
> \glshex 1D70A,% lower case omicron (maths italic)
> \glshex 1D6F0% upper case omicron (maths italic)
> }

\glsxtrMathItalicPi

> \newcommand*{\glsxtrMathItalicPi}{%
> \glshex 1D70B% lower case pi (maths italic)
> \string=\glshex 1D71B,% lower case pi variant (maths italic)
> \glshex 1D6F1% upper case pi (maths italic)
> }

\glsxtrMathItalicRho

> \newcommand*{\glsxtrMathItalicRho}{%
> \glshex 1D70C% lower case rho (maths italic)
> \string=\glshex 1D71A,% lower case rho variant (maths italic)
> \glshex 1D6F2% upper case rho (maths italic)
> }

\glsxtrMathItalicSigma

```
\newcommand*{\glsxtrMathItalicSigma}{%
 \glshex 1D70D% lower case final sigma (maths italic)
 \string=\glshex 1D70E,% lower case sigma (maths italic)
 \glshex 1D6F4% upper case sigma (maths italic)
}
```

\glsxtrMathItalicTau

```
\newcommand*{\glsxtrMathItalicTau}{%
 \glshex 1D70F,% lower case tau (maths italic)
 \glshex 1D6F5% upper case tau (maths italic)
}
```

\glsxtrMathItalicUpsilon

```
\newcommand*{\glsxtrMathItalicUpsilon}{%
 \glshex 1D710,% lower case upsilon (maths italic)
 \glshex 1D6F6% upper case upsilon (maths italic)
}
```

\glsxtrMathItalicPhi

```
\newcommand*{\glsxtrMathItalicPhi}{%
 \glshex 1D711% lower case phi (maths italic)
 \string=\glshex 1D719,% lower case phi variant (maths italic)
 \glshex 1D6F7% upper case phi (maths italic)
}
```

\glsxtrMathItalicChi

```
\newcommand*{\glsxtrMathItalicChi}{%
 \glshex 1D712,% lower case chi (maths italic)
 \glshex 1D6F8% upper case chi (maths italic)
}
```

\glsxtrMathItalicPsi

```
\newcommand*{\glsxtrMathItalicPsi}{%
 \glshex 1D713,% lower case psi (maths italic)
 \glshex 1D6F9% upper case psi (maths italic)
}
```

\glsxtrMathItalicOmega

```
\newcommand*{\glsxtrMathItalicOmega}{%
 \glshex 1D714,% lower case omega (maths italic)
 \glshex 1D6FA% upper case omega (maths italic)
}
```

\glsxtrMathItalicPartial

```
\newcommand*{\glsxtrMathItalicPartial}{%
 \glshex 1D715% partial differential (maths italic)
}
```

\glsxtrMathItalicNabla

```
\newcommand*{\glsxtrMathItalicNabla}{%
\glshex 1D6FB% nabla (maths italic)
}
```

\glsxtrdigitrules Digits from the Basic Latin set and subscript and superscript digit rules.

```
\newcommand*{\glsxtrdigitrules}{%
0\string=\glshex 2080\string=\glshex 2070
\string<1\string=\glshex 2081\string=\glshex 00B9
\string<2\string=\glshex 2082\string=\glshex 00B2
\string<3\string=\glshex 2083\string=\glshex 00B3
\string<4\string=\glshex 2084\string=\glshex 2074
\string<5\string=\glshex 2085\string=\glshex 2075
\string<6\string=\glshex 2086\string=\glshex 2076
\string<7\string=\glshex 2087\string=\glshex 2077
\string<8\string=\glshex 2088\string=\glshex 2078
\string<9\string=\glshex 2089\string=\glshex 2079
}
```

\glsxtrBasicDigitrules Digits from the Basic Latin set.

```
\newcommand*{\glsxtrBasicDigitrules}{%
0\string<1\string<2\string<3\string<4%
\string<5\string<6\string<7\string<8\string<9%
}
```

\glsxtrSubScriptDigitrules Subscript digits.

```
\newcommand*{\glsxtrSubScriptDigitrules}{%
\glshex 2080% subscript 0
\string<\glshex 2081% subscript 1
\string<\glshex 2082% subscript 2
\string<\glshex 2083% subscript 3
\string<\glshex 2084% subscript 4
\string<\glshex 2085% subscript 5
\string<\glshex 2086% subscript 6
\string<\glshex 2087% subscript 7
\string<\glshex 2088% subscript 8
\string<\glshex 2089% subscript 9
}
```

\glsxtrSuperScriptDigitrules Superscript digits.

```
\newcommand*{\glsxtrSuperScriptDigitrules}{%
\glshex 2070% superscript 0
\string<\glshex 00B9% superscript 1
\string<\glshex 00B2% superscript 2
\string<\glshex 00B3% superscript 3
\string<\glshex 2074% superscript 4
\string<\glshex 2075% superscript 5
\string<\glshex 2076% superscript 6
\string<\glshex 2077% superscript 7
```

```
                \string<\glshex 2078% superscript 8
                \string<\glshex 2079% superscript 9
              }
```

\glsxtrfractionrules Vulgar fractions.

```
              \newcommand*{\glsxtrfractionrules}{%
              \glshex 215F% fraction numerator one (1/)
              \string<\glshex 2189% zero thirds (0/3 = 0)
              \string<\glshex 2152% one tenth (1/10 = 0.1)
              \string<\glshex 2151% one ninth (1/9 ~ 0.111)
              \string<\glshex 215B% one eighth (1/8 = 0.125)
              \string<\glshex 2150% one seventh (1/7 ~ 0.143)
              \string<\glshex 2159% one sixth (1/6 ~ 0.167)
              \string<\glshex 2155% one fifth (1/5 = 0.2)
              \string<\glshex 00BC% one quarter (1/4 = 0.25)
              \string<\glshex 2153% one third (1/3 ~ 0.333)
              \string<\glshex 215C% three eighths (3/8 = 0.375)
              \string<\glshex 2156% two fifths (2/5 = 0.4)
              \string<\glshex 00BD% one half (1/2 = 0.5)
              \string<\glshex 2157% three fifths (3/5 = 0.6)
              \string<\glshex 215D% five eighths (5/8 = 0.625)
              \string<\glshex 2154% two thirds (2/3 ~ 0.667)
              \string<\glshex 00BE% three quarters (3/4 = 0.75)
              \string<\glshex 2158% four fifths (4/5 = 0.8)
              \string<\glshex 215A% five sixths (5/6 ~ 0.833)
              \string<\glshex 215E% seven eighths (7/8 = 0.875)
              }
```

\@glsxtrdialecthook Check for scripts associated with the document dialects.

```
              \renewcommand{\@glsxtrdialecthook}{%
               \ifundef\CurrentTrackedScript
               {%
                 \TrackLangIfHasDefaultScript{\CurrentTrackedLanguage}%
                 {%
                  \edef\CurrentTrackedScript{%
                    \TrackLangGetDefaultScript\CurrentTrackedLanguage}%
                 }%
                 {}%
               }%
               {}%
               \ifdef\CurrentTrackedScript
               {%
                 \let\gls@orgTrackLangRequireDialectPrefix\TrackLangRequireDialectPrefix
                 \def\TrackLangRequireDialectPrefix{glossariesxtr-}%
                 \let\CurrentTrackedTag\CurrentTrackedScript
                 \IfFileExists{\TrackLangRequireDialectPrefix\CurrentTrackedTag.ldf}
                 {\RequireGlossariesExtraLang{\CurrentTrackedTag}}%
                 {}%
                 \let\TrackLangRequireDialectPrefix\gls@orgTrackLangRequireDialectPrefix
               }%
```

```
    {}%
  }
```

If `\glsxtr@loaddialect` has been defined, then glossaries-extra-bib2gls has been
loaded after glossaries-extra. (For example, through `\glossariesextrasetup`.)
Not recommended, but if this has been done try to find the associated language
resources.

```
\ifdef\glsxtr@loaddialect
{%
  \@ifpackageloaded{tracklang}
  {%
    \AnyTrackedLanguages
    {%
      \ForEachTrackedDialect{\this@dialect}{\glsxtr@loaddialect}%
    }%
    {}%
  }
  {}
}
{}
```

4   Style Adjustments (**glossaries-extra-stylemods.sty**)

This package adjusts the predefined styles so that they include the post descrip-
tion hook. Also, some other minor adjustments may be made to make existing
styles more flexible.

4.1   Package Initialisation

First identify package:

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v1.48}{2021-11-22}{glossaries-extra-stylemods-2021-11-22.sty}
\DeclareCurrentRelease{v1.7}{2025-05-14}
```

Declare package:

```
\ProvidesPackage{glossaries-extra-stylemods}[2025/05/14 v1.7 (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`.

`\@glsxtr@loadstyles`

```
\newcommand*{\@glsxtr@loadstyles}{}
```

all Provide all known styles.

```
\DeclareOption{all}{%
```

```
        \appto\@glsxtr@loadstyles{%
          \RequirePackage{glossary-inline}%
          \RequirePackage{glossary-list}%
          \RequirePackage{glossary-tree}%
          \RequirePackage{glossary-mcols}%
          \RequirePackage{glossary-long}%
          \RequirePackage{glossary-longragged}%
          \RequirePackage{glossary-longbooktabs}%
          \RequirePackage{glossary-super}%
          \RequirePackage{glossary-superragged}%
          \RequirePackage{glossary-bookindex}%
          \RequirePackage{glossary-longextra}%
          \RequirePackage{glossary-topic}%
          \RequirePackage{glossary-table}%
        }
      }

      \DeclareOption*{%
        \IfFileExists{glossary-\CurrentOption.sty}
        {\eappto\@glsxtr@loadstyles{%
            \noexpand\RequirePackage{glossary-\CurrentOption}}%
        }%
        {%
          \PackageError{glossaries-extra-styles}%
          {Unknown option '\CurrentOption'}{}%
        }%
      }
```

Process the package options:

```
\ProcessOptions
```

Load the required packages:

```
\@glsxtr@loadstyles
```

Adjust the styles so that they all have the post description hook. Also, instead of having a hard-coded \space before the location, use:

\glsxtrprelocation This uses \providecommand as the same command is also provided by glossary-bookindex.

```
\providecommand*{\glsxtrprelocation}{\space}
```

In case we have an old version of glossaries:

\renewglossarystyle

```
\providecommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries-extra}{Glossary style '#1' isn't already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
```

663

4.2 List-Like Styles

The list-like styles mostly already use the post description hook. Only the listdotted style need modifying to add this.

```
\ifdef{\@glsstyle@listdotted}
{%
 \renewglossarystyle{listdotted}{%
   \setglossarystyle{list}%
   \renewcommand*{\glossentry}[2]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}%
      \glossentrydesc{##1}\glspostdescription}%
   \renewcommand*{\subglossentry}[3]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\glossentryname{##2}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}%
    \glossentrydesc{##2}\glspostdescription}%
  }
 }
 {%
```

Assume the style isn't required if it hasn't already been defined.

```
 }
```

The sublistdotted style doesn't display the description for top-level entries. Sub-level entries use the listdottedstyle.

The other list styles would be easier to adapt if the space before the number list wasn't hard coded.

```
\ifdef{\@glsstyle@list}
{%
```

\glslistprelocation   Space before number list for top-level entries.

```
\newcommand{\glslistprelocation}{\glsxtrprelocation}
```

\glslistchildprelocation   Space before number list for child entries.

```
\newcommand{\glslistchildprelocation}{\glslistprelocation}
```

\glslistchildpostlocation   Full stop after number list.

```
\newcommand{\glslistchildpostlocation}{.}
```

\glslistdesc

```
\newcommand{\glslistdesc}[1]{\glossentrydesc{#1}\glspostdescription}
```

\glslistgroupskip

```
\newcommand{\glslistgroupskip}{\nobreak\indexspace\nobreak}
```

`\glslistitem`

```
\newcommand{\glslistitem}[1]{%
  \item[\glsentryitem{#1}%
        \glstarget{#1}{\glossentryname{#1}}]%
}
```

`\glslistinit` This command was only added to glossary-list v4.48 so provide it if it hasn't been defined:

```
\providecommand{\glslistinit}{%
  \ifdef\GetTitleStringDisableCommands
  {%
    \GetTitleStringSetup{expand}%
    \GetTitleStringDisableCommands{%
      \let\glsentryitem\@gobble
      \let\glstarget\@secondoftwo
      \let\glossentryname\glslistexpandedname
      \let\glslistgroupheaderfmt\@firstofone
      \let\glsgetgrouptitle\@firstofone
```

Technically this has an optional argument but it's not used in the list styles.

```
      \let\glsnavhypertarget\@secondoftwo
      \let\glsnavigation\relax
    }%
  }%
  {}%
}
```

`\glslistexpandedname` This command was only added to glossary-list v4.48 so provide it if it hasn't been defined. The original definition uses `\glsunexpandedfieldvalue` which was added to glossaries v4.48 (so if `\glslistexpandedname` hasn't been defined then neither will `\glsunexpandedfieldvalue`).

```
\providecommand{\glslistexpandedname}[1]{%
  \ifcsname glo@\glsdetoklabel{#1}@name\endcsname
   \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\expandafter\endcsname
  \fi
}
```

Redefine list to use these commands.

```
\renewglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\glslistinit\begin{description}}{\end{description}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the list styles.

```
  \renewcommand*{\glssubgroupheading}[4]{}%
  \renewcommand*{\glossentry}[2]{%
    \glslistitem{##1}\glslistdesc{##1}\glslistprelocation ##2}%
```

```
                      \renewcommand*{\subglossentry}[3]{%
                        \glssubentryitem{##2}%
                        \glstarget{##2}{\strut}\space
                        \glslistdesc{##2}%
                        \glslistchildprelocation ##3\glslistchildpostlocation}%
                      \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glslistgroupskip\fi}%
                    }
                  }
                  {}
```

Similarly for altlist. Since it requires list, the new commands should have been
defined above.

```
                  \ifdef{\@glsstyle@altlist}
                  {%
```

\glsaltlistitem

```
                      \newcommand{\glsaltlistitem}[1]{%
                        \glslistitem{#1}%
                        \mbox{}\par\nobreak\@afterheading
                      }

                      \renewglossarystyle{altlist}{%
                        \setglossarystyle{list}%
                        \renewcommand*{\glossentry}[2]{%
                          \glsaltlistitem{##1}%
                          \glslistdesc{##1}\glslistprelocation ##2}%
                        \renewcommand{\subglossentry}[3]{%
                          \par
                          \glssubentryitem{##2}%
                          \glstarget{##2}{\strut}\glslistdesc{##2}%
                          \glslistchildprelocation ##3}%
                      }
                    }
                    {}
```

Redefine listgroup so that it discourages a break after group headings.

```
                  \ifdef{\@glsstyle@listgroup}
                  {%
```

\glslistgroupheaderitem

```
                      \newcommand{\glslistgroupheaderitem}[2]{\item[{#2}]}
```

\glslistgroupafterheader

```
                      \newcommand{\glslistgroupafterheader}{%
                          \mbox{}\par\nobreak\@afterheading
                      }

                      \renewglossarystyle{listgroup}{%
                        \setglossarystyle{list}%
                        \renewcommand*{\glsgroupheading}[1]{%
```

```
      \glslistgroupheaderitem{##1}{\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}}%
      \glslistgroupafterheader
    }%
  }
}
{}
```
Similarly for listhypergroup.
```
  \ifdef{\@glsstyle@listhypergroup}
  {%
    \renewglossarystyle{listhypergroup}{%
      \setglossarystyle{list}%
      \renewcommand*{\glossaryheader}{%
        \glslistnavigationitem{\glsnavigation}}%
      \renewcommand*{\glsgroupheading}[1]{%
        \glslistgroupheaderitem{##1}{\glslistgroupheaderfmt
          {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}}%
        \glslistgroupafterheader
      }%
    }
  }
  {}
```
Similarly for altlistgroup.
```
  \ifdef{\@glsstyle@altlistgroup}
  {%
    \renewglossarystyle{altlistgroup}{%
      \setglossarystyle{altlist}%
      \renewcommand*{\glsgroupheading}[1]{%
        \glslistgroupheaderitem{##1}%
          {\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}}%
        \glslistgroupafterheader
      }%
    }
  }
  {}
```
Similarly for altlisthypergroup.
```
  \ifdef{\@glsstyle@altlisthypergroup}
  {%
    \renewglossarystyle{altlisthypergroup}{%
      \setglossarystyle{altlist}%
      \renewcommand*{\glossaryheader}{%
        \glslistnavigationitem{\glsnavigation}}%
      \renewcommand*{\glsgroupheading}[1]{%
        \glslistgroupheaderitem{##1}{\glslistgroupheaderfmt
          {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}}%
        \glslistgroupafterheader
      }%
    }
  }
```

```
{}
```

4.3 Longtable Styles

The three and four column styles require adjustment to add the post-description hook. The two column styles need the hard-coded `\space` changed to `\glsxtrprelocation`.

```
\ifcsdef{@glsstyle@long}
{%
  \renewglossarystyle{long}{%
    \renewenvironment{theglossary}%
      {\begin{longtable}{lp{\glsdescwidth}}}%
      {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the long styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription
      \glsxtrprelocation ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
      \glsxtrprelocation ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
    \fi
  }
}
{}
```

Three column style:

```
\ifcsdef{@glsstyle@long3col}
{%
  \renewglossarystyle{long3col}{%
    \renewenvironment{theglossary}%
      {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
      {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the long styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      ##3\tabularnewline
    }%
```

Conditional needs to be outside of `\glsgroupskip` otherwise it can cause "Incomplete `\iftrue`" errors.

```
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& &\tabularnewline}%
    \fi
  }
}
{}
```

Four column style:

```
 \ifcsdef{@glsstyle@long4col}
 {%
   \renewglossarystyle{long4col}{%
     \renewenvironment{theglossary}%
       {\begin{longtable}{llll}}%
       {\end{longtable}}%
     \renewcommand*{\glossaryheader}{}%
     \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the long styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription &
      \glossentrysymbol{##1} &
      ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      \glossentrysymbol{##2} & ##3\tabularnewline
    }%
```

```
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
    \fi
  }
}
{}
```

The styles in glossary-longbooktabs are all based on the styles in glossary-long, so no adjustments are needed for that package.

4.4 Long Ragged Styles

The three and four column styles require adjustment for the post-description hook, but not the two column styles. However, the two-column styles need to have \space replaced with \glsxtrprelocation.

```
\ifcsdef{@glsstyle@longragged}
{%
  \renewglossarystyle{longragged}{%
    \renewenvironment{theglossary}%
       {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
       {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with \printunsrtglossary. This does nothing as the sub-entries don't have the name displayed for the longragged styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription\glsxtrprelocation ##2%
      \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}%
      \glspostdescription\glsxtrprelocation ##3%
       \tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
    \fi
  }
}
{}
```

Three and four column styles don't use `\glsxtrprelocation` since the number list is in its own column.

```
\ifcsdef{@glsstyle@longragged3col}
{%
  \renewglossarystyle{longragged3col}{%
    \renewenvironment{theglossary}%
      {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
          >{\raggedright}p{\glspagelistwidth}}}%
      {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the longragged styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      ##3\tabularnewline
    }%

    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& &\tabularnewline}%
    \fi
  }
}
{}
```

Four column style:

```
\ifcsdef{@glsstyle@altlongragged4col}
{%
  \renewglossarystyle{altlongragged4col}{%
    \renewenvironment{theglossary}%
      {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
          >{\raggedright}p{\glspagelistwidth}}}%
      {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the longragged styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
```

```
      \glossentrydesc{##1}\glspostdescription & \glossentrysymbol{##1} &
        ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
       \glssubentryitem{##2}%
       \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
       \glossentrysymbol{##2} & ##3\tabularnewline
    }%

    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
    \fi
  }
}
{}
```

4.5 Supertabular Styles

The three and four column styles require adjustment to add the post-description
hook. The two column styles need the hard-coded `\space` changed to
`\glsxtrprelocation`.

```
\ifcsdef{@glsstyle@super}
{%
  \renewglossarystyle{super}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}%
       \begin{supertabular}{lp{\glsdescwidth}}}%
      {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing
as the sub-entries don't have the name displayed for the super styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription
      \glsxtrprelocation ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
       \glssubentryitem{##2}%
       \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
       \glsxtrprelocation ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
```

```
      \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
      \fi
    }
  }
  {}
```
Three column style:
```
  \ifcsdef{@glsstyle@super3col}
  {%
    \renewglossarystyle{super3col}{%
      \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}%
         \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{supertabular}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
```
Sub-groups are only supported with \printunsrtglossary. This does nothing as the sub-entries don't have the name displayed for the super styles.
```
      \renewcommand*{\glssubgroupheading}[4]{}%
      \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
        ##3\tabularnewline
      }%

      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{ & &\tabularnewline}%
      \fi
    }
  }
  {}
```
Four column styles:
```
  \ifcsdef{@glsstyle@super4col}
  {%
    \renewglossarystyle{super4col}{%
      \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}%
         \begin{supertabular}{llll}}{%
         \end{supertabular}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the super styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription &
      \glossentrysymbol{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      \glossentrysymbol{##2} & ##3\tabularnewline
    }%

    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
    \fi
  }
 }
 {}
```

4.6   Super Ragged Styles

The three and four column styles require adjustment for the post-description hook, but not the two column styles. However, the two-column styles need to have `\space` replaced with `\glsxtrprelocation`.

```
 \ifcsdef{@glsstyle@superragged}
 {%
   \renewglossarystyle{superragged}{%
     \renewenvironment{theglossary}%
       {\tablehead{}\tabletail{}%
        \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
       {\end{supertabular}}%
     \renewcommand*{\glossaryheader}{}%
     \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`. This does nothing as the sub-entries don't have the name displayed for the super styles.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription\glsxtrprelocation ##2%
      \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
```

```
            \glssubentryitem{##2}%
            \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
            \glsxtrprelocation ##3%
            \tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
      \fi
    }
  }
  {}
```

Three column style:

```
  \ifcsdef{@glsstyle@superragged3col}
  {%
    \renewglossarystyle{superragged3col}{%
      \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}%
         \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
             >{\raggedright}p{\glspagelistwidth}}}%
        {\end{supertabular}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with \printunsrtglossary. This does nothing as the sub-entries don't have the name displayed for the super styles.

```
      \renewcommand*{\glssubgroupheading}[4]{}%
      \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription &
        ##2\tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
         &
         \glssubentryitem{##2}%
         \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
         ##3\tabularnewline
      }%

      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{ & &\tabularnewline}%
      \fi
    }
  }
  {}
```

Four columns:

```
\ifcsdef{@glsstyle@altsuperragged4col}
{%
  \renewglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}%
       \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
         >{\raggedright}p{\glspagelistwidth}}}%
      {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription &
      \glossentrysymbol{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      \glossentrysymbol{##2} & ##3\tabularnewline
    }%

    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
    \fi
  }
}
{}
```

4.7   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`.

```
\ifdef{\@glsstyle@inline}
{%
    \renewcommand*{\glspostinline}{.\spacefactor\sfcode`\.}
```

Just use `\glsxtrpostdescription` instead of `\glspostdescription`.

```
    \renewcommand*{\glsinlinedescformat}[3]{%
      \space#1\glsxtrpostdescription}
    \renewcommand*{\glsinlinesubdescformat}[3]{%
      #1\glsxtrpostdescription}
```

The default settings don't show the location lists, so there's no adjustment for `\glsxtrprelocation`.

```
}
```

```
{}
```

4.8 Tree Styles

Redefine both `\glstreenamefmt` and `\glstreegroupheaderfmt` in terms of `\glstreedefaultnamefmt` to make it easier to change both at the same time or only change one without affecting the other.

```
\ifdef\glstreenamefmt
{
```

`\glstreedefaultnamefmt`

```
\newcommand{\glstreedefaultnamefmt}[1]{\textbf{#1}}
```

`\glstreenamefmt`

```
\renewcommand{\glstreenamefmt}[1]{\glstreedefaultnamefmt{#1}}
```

`\glstreegroupheaderfmt` This command was only introduced to glossary-tree v4.22, so it may not be defined.

```
\def\glstreegroupheaderfmt#1{\glstreedefaultnamefmt{#1}}
```

`\glstreenavigationfmt` This command was only introduced to glossary-tree v4.22, so it may not be defined.

```
\def\glstreenavigationfmt#1{\glstreedefaultnamefmt{#1}}
```

`\glstreePreHeader` Takes the label as the first argument and title as the second argument so this can be modified to add a bookmark.

```
\newcommand{\glstreePreHeader}[2]{}
```

`\glstreeSubPreHeader`

> `\glstreeSubPreHeader{⟨previous group level⟩}{⟨level⟩}`
> `{⟨parent label⟩}{⟨group label⟩}{⟨title⟩}`

```
\newcommand{\glstreeSubPreHeader}[5]{}
```

```
}
{}
```

The index style is redefined so that the space before the number list isn't hard coded.

```
\ifdef{\@glsstyle@index}
{
```

`\glstreeprelocation` The space before the number list for top-level entries. This is shared by the other tree styles.

```
\newcommand*{\glstreeprelocation}{\glsxtrprelocation}
```

`\glstreechildprelocation` The space before the number list for child entries. This is shared by the other tree styles.

```
\newcommand*{\glstreechildprelocation}{\glstreeprelocation}
```

Don't prohibit a page break at the start of a new group if there's no header.

`\glstreegroupskip`

```
\newcommand{\glstreegroupskip}{\indexspace}
```

`\glstreegroupheaderskip` This doesn't include `\@afterheading` as it can cause interference with some styles.

```
\newcommand{\glstreegroupheaderskip}{\nopagebreak\glstreegroupskip\nobreak}
```

Modify the index style.

```
\renewglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
     \let\item\glstreeitem
     \let\subitem\glstreesubitem
     \let\subsubitem\glstreesubsubitem
    }%
  {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with `\printunsrtglossary`.

```
  \renewcommand*{\glssubgroupheading}[4]{}%
  \renewcommand*{\glossentry}[2]{%
      \item\glsentryitem{##1}%
      \glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
      \glstreesymbol{##1}%
      \glstreeDescLoc{##1}{##2}%
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifcase##1\relax
      \item
    \or
      \subitem
      \glssubentryitem{##2}%
    \else
      \subsubitem
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \glstreechildsymbol{##2}%
    \glstreeChildDescLoc{##2}{##3}%
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
 }
}
```

```
{}
```

The indexgroup style is redefined to discourage a page break after the heading.

```
\ifdef{\@glsstyle@indexgroup}
{%
```

Provide formatting command for sub-headings to make it easier to adjust.

> \glsindexsubgroupitem{⟨*previous group level*⟩}{⟨*level*⟩}
> {⟨*parent label*⟩}{⟨*group label*⟩}{⟨*title*⟩}

```
\newcommand*{\glsindexsubgroupitem}[5]{%
  \ifcase#2\relax
```

This case shouldn't occur as `\glsgroupheading` will be used instead, but include for completeness.

```
    \item \glstreegroupheaderfmt{#5}%
    \glstreegroupheaderskip
  \or
    \smallskip
    \subitem \glstreegroupheaderfmt{#5}%
    \smallskip
  \else
    \smallskip
    \subsubitem \glstreegroupheaderfmt{#5}%
    \smallskip
  \fi
}

\renewglossarystyle{indexgroup}{%
  \setglossarystyle{index}%
```

Group heading.

```
  \renewcommand*{\glsgroupheading}[1]{%
    \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
    \glstreePreHeader{##1}{\glsxtr@grptitle}%
    \item\glstreegroupheaderfmt{\glsxtr@grptitle}%
    \glstreegroupheaderskip\@afterheading
  }%
```

Sub-groups are only supported with `\printunsrtglossary`.

```
  \renewcommand*{\glssubgroupheading}[4]{%
    \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
    \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
    \glsindexsubgroupitem{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
    \@afterheading
  }%
}
```

```
    }
    {}
```

Similarly for indexhypergroup.

```
\ifdef{\@glsstyle@indexhypergroup}
{%
  \renewglossarystyle{indexhypergroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glossaryheader}{%
      \item\glstreenavigationfmt{\glsnavigation}%
       \glstreegroupheaderskip\@afterheading}%
```

Group heading.

```
    \renewcommand*{\glsgroupheading}[1]{%
    \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
    \glstreePreHeader{##1}{\glsxtr@grptitle}%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
    \glstreegroupheaderskip\@afterheading}%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}[4]{%
    \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
    \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
    \glsindexsubgroupitem{##1}{##2}{##3}{##4}%
      {\glsnavhypertarget{##4}{\glsxtr@grptitle}}%
    \@afterheading
    }%
  }%
}
{}
```

Adjust tree style to remove hard coded space before number list.

```
\ifdef{\@glsstyle@tree}
{%
```

The original alttree style doesn't use \glstreepredesc but since v1.42 the modified style (below) has switched to using \glstreeDescLoc so provide an alternative that can be used with alttree.

\glsxtrtreepredesc

```
\newcommand{\glsxtrtreepredesc}{\glstreepredesc}
```

\glsxtrtreechildpredesc

```
\newcommand{\glsxtrtreechildpredesc}{\glstreechildpredesc}
```

Provide a command for use with the tree styles that displays the pre-description separator, the description and post-description hook.

\glstreedesc

```
\newcommand{\glstreedesc}[1]{%
  \glsxtrtreepredesc\glossentrydesc{#1}\glspostdescription
}
```

> **\glstreeDescLoc{⟨*label*⟩}{⟨*location*⟩}**

This checks for the description and symbol. If both are missing, a different separator may be required. For example, a comma and space if there's no description or symbol but just a space if either of those fields are present.

```
\newcommand{\glstreeDescLoc}[2]{%
  \ifglshasdesc{#1}%
  {\glstreedesc{#1}\glstreeprelocation}%
  {\ifglshassymbol{#1}{\glstreeprelocation}{\glstreeNoDescSymbolPreLocation}}%
  #2%
}
```

> **\glstreeNoDescSymbolPreLocation**

```
\newcommand{\glstreeNoDescSymbolPreLocation}{\space}
```

Similarly for the symbol.

\glstreesymbol

```
\newcommand{\glstreesymbol}[1]{%
  \ifglshassymbol{#1}{\space(\glossentrysymbol{#1})}{}%
}%
```

And for the child entries:

\glstreechilddesc

```
\newcommand{\glstreechilddesc}[1]{%
  \glsxtrtreechildpredesc\glossentrydesc{#1}\glspostdescription
}%
```

\glstreeChildDescLoc

```
\newcommand{\glstreeChildDescLoc}[2]{%
  \ifglshasdesc{#1}%
  {\glstreechilddesc{#1}\glstreechildprelocation}%
  {\ifglshassymbol{#1}{\glstreechildprelocation}%
    {\glstreeNoDescSymbolPreLocation}%
  }%
  #2%
}%
```

\glstreechildsymbol This just behaves in the same way as the top-level.

```
\newcommand{\glstreechildsymbol}[1]{%
  \glstreesymbol{#1}%
}%
```

Redefine tree style.

```
\renewglossarystyle{tree}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
```

Group heading.

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

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}[4]{}%
```

Top level entry.

```
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \glstreesymbol{##1}%
    \glstreeDescLoc{##1}{##2}\par
  }%
```

Sub entries.

```
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \glstreechildsymbol{##2}%
    \glstreeChildDescLoc{##2}{##3}\par
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
 }%
}
{}
```

The treegroup style is redefined to discourage a page break after the heading.

```
\ifdef{\@glsstyle@treegroup}
{%
```

Provide formatting command for sub-headings to make it easier to adjust.

\glstreesubgroupitem{⟨*previous group level*⟩}{⟨*level*⟩}
{⟨*parent label*⟩}{⟨*group label*⟩}{⟨*title*⟩}

\glstreesubgroupitem

```
    \newcommand*{\glstreesubgroupitem}[5]{%
     \par\smallskip\noindent\hspace{#2\glstreeindent}%
     \glstreegroupheaderfmt{#5}\smallskip\par
    }
```

Redefine treegroup style.

```
    \renewglossarystyle{treegroup}{%
     \setglossarystyle{tree}%
```

Group heading.

```
    \renewcommand{\glsgroupheading}[1]{%
     \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
     \glstreePreHeader{##1}{\glsxtr@grptitle}%
     \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
     \glstreegroupheaderskip\@afterheading}%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}[4]{%
     \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
     \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
     \glstreesubgroupitem{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
    }%
    }
  }
  {}
```

   Similarly for treehypergroup

```
 \ifdef{\@glsstyle@treehypergroup}
 {%
  \renewglossarystyle{treehypergroup}{%
   \setglossarystyle{tree}%
   \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}%
    \glstreegroupheaderskip\@afterheading}%
```

Group heading.

```
    \renewcommand*{\glsgroupheading}[1]{%
     \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
     \glstreePreHeader{##1}{\glsxtr@grptitle}%
     \par\noindent
     \glstreegroupheaderfmt
       {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
     \glstreegroupheaderskip\@afterheading}%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}[4]{%
     \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
     \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
     \glstreesubgroupitem{##1}{##2}{##3}{##4}%
       {\glsnavhypertarget{##4}{\glsxtr@grptitle}}%
    }%
    }
```

```
}
{}
```

Adjust treenoname style to remove hard coded space before number list.

```
\ifdef{\@glsstyle@treenoname}
{%
```

Provide a command for use with the treenoname styles that displays the pre-description separator, the description and post-description hook.

\glstreenonamedesc

```
\newcommand{\glstreenonamedesc}[1]{%
  \glstreepredesc\glossentrydesc{#1}\glspostdescription
}%
```

Similarly for the symbol.

\glstreenonamesymbol

```
\newcommand{\glstreenonamesymbol}[1]{%
  \ifglshassymbol{#1}{\space(\glossentrysymbol{#1})}{}%
}%
```

\glstreenonameDescLoc

```
\newcommand{\glstreenonameDescLoc}[2]{%
  \glstreenonamedesc{#1}\glstreeprelocation#2%
}
```

\glstreenonamechilddesc The child entry doesn't have the pre-description separator as the name isn't displayed.

```
\newcommand{\glstreenonamechilddesc}[1]{%
  \glossentrydesc{#1}\glspostdescription
}%
```

\glstreenonameChildDescLoc

```
\newcommand{\glstreenonameChildDescLoc}[2]{%
  \glstreenonamechilddesc{#1}\glstreechildprelocation#2%
}
```

Redefine treenoname style

```
\renewglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
```

Group heading.

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

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}[4]{}%
    \renewcommand{\glossentry}[2]{%
      \hangindent0pt\relax
      \parindent0pt\relax
      \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
      \glstreenonamesymbol{##1}%

      \glstreenonameDescLoc{##1}{##2}\par
    }%
    \renewcommand{\subglossentry}[3]{%
      \hangindent##1\glstreeindent\relax
      \parindent##1\glstreeindent\relax
      \ifnum##1=1\relax
        \glssubentryitem{##2}%
      \fi
      \glstarget{##2}{\strut}%
      \glstreenonameChildDescLoc{##2}{##3}\par
    }%
    \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
  }
}
{}
```

The treenonamegroup style is redefined to discourage a page break after the heading. There are no sub-groups as sub-entries don't have the name shown.

```
\ifdef{\@glsstyle@treenonamegroup}
{%
  \renewglossarystyle{treenonamegroup}{%
    \setglossarystyle{treenoname}%
    \renewcommand{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}
{}
```

Similarly for treenonamehypergroup

```
\ifdef{\@glsstyle@treenonamehypergroup}
{%
  \renewglossarystyle{treenonamehypergroup}{%
    \setglossarystyle{treenoname}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip\@afterheading}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
```

```
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading}%
    }
  }
  {}
```

Version 1.6: Check for the existence of new function added to glossary-tree v4.59.

```
\ExplSyntaxOn
\cs_if_exist:NT \glossaries_tree_set_widest_name:nn
 {
```

\gglssetwidest

```
    \NewDocumentCommand \gglssetwidest { O{0} m }
     {
       \glossaries_tree_gset_widest_name:nn { #1 } { #2 }
     }
```

\eglssetwidest Locally set expanded value. NB protected expansion is required in case of fragile commands.

```
    \NewDocumentCommand \eglssetwidest { O{0} m }
     {
       \int_compare:nNnT
       { #1 } > { \l__glossaries_tree_max_known_level_int }
       {
         \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
       }
       \protected@csedef { @glswidestname \romannumeral #1 } { #2 }
     }
```

\xglssetwidest Globally set expanded value. NB protected expansion is required in case of fragile commands.

```
    \NewDocumentCommand \xglssetwidest { O{0} m }
     {
       \int_compare:nNnT
       { #1 } > { \l__glossaries_tree_max_known_level_int }
       {
         \int_gset:Nn \l__glossaries_tree_max_known_level_int { #1 }
       }
       \protected@csxdef { @glswidestname \romannumeral #1 } { #2 }
     }
```

Only set if wider than the current value.

```
\cs_new:Nn \__glossaries_extra_update_widest:Nnn
 {
  \ifcsundef { @glswidestname \romannumeral #2 }
   { #1 [ #2 ] { #3 } }
```

686

```
                    {
                      \exp_args:NNv \glsmeasurewidth \l__glossaries_tmpa_dim
                       { @glswidestname \romannumeral #2 }
                      \glsmeasurewidth \l__glossaries_tmpb_dim { #3 }
                      \dim_compare:nNnT
                         { \l__glossaries_tmpb_dim } > { \l__glossaries_tmpa_dim }
                      {
                        #1 [ #2 ] { #3 }
                      }
                    }
                  }
```

\glsupdatewidest Only sets if new value is wider than old value.

```
                  \NewDocumentCommand \glsupdatewidest { O{0} m }
                  {
                    \__glossaries_extra_update_widest:Nnn
                     \glssetwidest { #1 } { #2 }
                  }
```

\gglsupdatewidest As above but global definition.

```
                  \NewDocumentCommand \gglsupdatewidest { O{0} m }
                  {
                    \__glossaries_extra_update_widest:Nnn
                     \gglssetwidest { #1 } { #2 }
                  }
```

\eglsupdatewidest As \glsupdatewidest but expands value.

```
                  \NewDocumentCommand \eglsupdatewidest { O{0} m }
                  {
                    \__glossaries_extra_update_widest:Nnn
                     \eglssetwidest { #1 } { #2 }
                  }
```

\xglsupdatewidest As above but global.

```
                  \NewDocumentCommand \xglsupdatewidest { O{0} m }
                  {
                    \__glossaries_extra_update_widest:Nnn
                     \xglssetwidest { #1 } { #2 }
                  }

                 }
                \ExplSyntaxOff
```

The alttree style is redefined to make it easier to made minor adjustments.

```
                \ifdef{\@glsstyle@alttree}
                {%
```

Only redefine this style if it's already been defined.

\glsalttreepredesc

```
                \newcommand{\glsalttreepredesc}{}
```

`\glsalttreechildpredesc`

```
\newcommand{\glsalttreechildpredesc}{\glsalttreepredesc}
```

`\glsxtralttreeSymbolDescLocation{⟨label⟩}{⟨location list⟩}`

`xtralttreeSymbolDescLocation`

Layout the symbol, description and location for top-level entries.

```
\newcommand{\glsxtralttreeSymbolDescLocation}[2]{%
  {%
    \let\par\glsxtrAltTreePar

    \let\glsxtrtreepredesc\glsalttreepredesc
    \let\glsxtrtreechildpredesc\glsalttreechildpredesc
    \ifglshassymbol{#1}{(\glossentrysymbol{#1})\space}{}%

    \glstreeDescLoc{#1}{#2}\par
  }%
}
```

`\glsxtrAltTreeIndent` Paragraph indent for subsequent paragraphs in multi-paragraph descriptions.

```
\newlength\glsxtrAltTreeIndent
```

`\glsxtrAltTreePar` Multi-paragraph descriptions need to keep the hanging indent.

```
\newcommand{\glsxtrAltTreePar}{%
  \@@par
  \glsxtrAltTreeSetHangIndent
  \setlength{\parindent}{\dimexpr\hangindent+\glsxtrAltTreeIndent}%
}
```

`\glsxtralttreeSubSymbolDescLocation{⟨level⟩}{⟨label⟩} {⟨location list⟩}`

`alttreeSubSymbolDescLocation`

Layout the symbol, description and location for sub-entries. Defaults to the same as the top-level.

```
\newcommand{\glsxtralttreeSubSymbolDescLocation}[3]{%
  \glsxtralttreeSymbolDescLocation{#2}{#3}%
}
```

`\glsxtrtreetopindent` 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.

```
\newlength\glsxtrtreetopindent
```

`\glsxtralttreeInit` User-level initialisation for the alttree style.

```
\newcommand*{\glsxtralttreeInit}{%
  \glsmeasurewidth{\glsxtrtreetopindent}{\glstreenamefmt{\glsgetwidestname\space}}%
  \glsxtrAltTreeIndent=\parindent
}
```

Older definitions. Now use `\ProvideDocumentCommand` to provide fallback definitions if an older version of glossary-tree is installed.

Globally set the widest value.

```
\ProvideDocumentCommand\gglssetwidest{O{0}m}{%
  \csgdef{@glswidestname\romannumeral#1}{#2}%
}
```

Local protected expansion:

```
\ProvideDocumentCommand\eglssetwidest{O{0}m}{%
  \protected@csedef{@glswidestname\romannumeral#1}{#2}%
}
```

Like the above but uses `\protected@csxdef`.

```
\ProvideDocumentCommand\xglssetwidest{O{0}m}{%
  \protected@csxdef{@glswidestname\romannumeral#1}{#2}%
}
```

Only sets if new value is wider than old value.

```
\ProvideDocumentCommand\glsupdatewidest{O{0}m}{%
 \ifcsundef{@glswidestname\romannumeral#1}%
 {\csdef{@glswidestname\romannumeral#1}{#2}}%
 {%
   \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
   \settowidth{\dimen@ii}{#2}%
   \ifdim\dimen@ii>\dimen@
    \csdef{@glswidestname\romannumeral#1}{#2}%
   \fi
 }%
}
```

As above but global definition.

```
\ProvideDocumentCommand\gglsupdatewidest{O{0}m}{%
 \ifcsundef{@glswidestname\romannumeral#1}%
 {\csgdef{@glswidestname\romannumeral#1}{#2}}%
 {%
   \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
   \settowidth{\dimen@ii}{#2}%
   \ifdim\dimen@ii>\dimen@
    \csgdef{@glswidestname\romannumeral#1}{#2}%
   \fi
 }%
}
```

As `\glsupdatewidest` but expands value.

```
\ProvideDocumentCommand\eglsupdatewidest{O{0}m}{%
```

```
                           \ifcsundef{@glswidestname\romannumeral#1}%
                           {\protected@csedef{@glswidestname\romannumeral#1}{#2}}%
                           {%
                             \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
                             \settowidth{\dimen@ii}{#2}%
                             \ifdim\dimen@ii>\dimen@
                              \protected@csedef{@glswidestname\romannumeral#1}{#2}%
                             \fi
                           }%
                           }

                           \ProvideDocumentCommand\xglsupdatewidest{O{0}m}{%
                           \ifcsundef{@glswidestname\romannumeral#1}%
                           {\protected@csxdef{@glswidestname\romannumeral#1}{#2}}%
                           {%
                             \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
                             \settowidth{\dimen@ii}{#2}%
                             \ifdim\dimen@ii>\dimen@
                              \protected@csxdef{@glswidestname\romannumeral#1}{#2}%
                             \fi
                           }%
                           }
```

\glsgetwidestname   Provide a user-level macro to obtain the widest top-level name.

```
                     \newcommand*{\glsgetwidestname}{\@glswidestname}
```

\glsgetwidestsubname   Provide a user-level macro to obtain the widest sub-entry name.

```
                        \newcommand*{\glsgetwidestsubname}[1]{%
                        \ifcsundef{@glswidestname\romannumeral#1}%
                        {\@glswidestname}%
                        {\csuse{@glswidestname\romannumeral#1}}%
                        }
```

\glsFindWidestTopLevelName   CamelCase is easier for long command names. Provide a CamelCase synonym
                             of \glsfindwidesttoplevelname.

```
                              \let\glsFindWidestTopLevelName\glsfindwidesttoplevelname
```

\glsFindWidestUsedTopLevelName   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.

```
                                  \newrobustcmd*{\glsFindWidestUsedTopLevelName}[1][\@glo@types]{%
                                  \dimen@=0pt\relax
                                  \gls@tmplen=0pt\relax
                                  \forallglossaries[#1]{\@gls@type}%
                                  {%
                                    \forglsentries[\@gls@type]{\@glo@label}%
                                    {%
                                      \ifglsused{\@glo@label}%
```

690

```
          {%
            \ifglshasparent{\@glo@label}%
            {}%
            {%
              \glsmeasurewidth{\dimen@}%
               {\glstreenamefmt{\glsentryname{\@glo@label}}}%
              \ifdim\dimen@>\gls@tmplen
                \gls@tmplen=\dimen@
                \eglssetwidest{\glsentryname{\@glo@label}}}%
              \fi
            }%
          }%
          {}%
        }%
      }%
    }
```

\glsFindWidestUsedAnyName Like the above but doesn't check the parent key. Useful if all levels should have
                           the same width for the name.

```
    \newrobustcmd*{\glsFindWidestUsedAnyName}[1][\@glo@types]{%
      \dimen@=0pt\relax
      \gls@tmplen=0pt\relax
      \forallglossaries[#1]{\@gls@type}%
      {%
        \forglsentries[\@gls@type]{\@glo@label}%
        {%
          \ifglsused{\@glo@label}%
          {%
            \glsmeasurewidth{\dimen@}%
             {\glstreenamefmt{\glsentryname{\@glo@label}}}%
            \ifdim\dimen@>\gls@tmplen
              \gls@tmplen=\dimen@
              \eglssetwidest{\glsentryname{\@glo@label}}}%
            \fi
          }%
          {}%
        }%
      }%
    }
```

\glsFindWidestAnyName Like the above but doesn't check is the entry has been used.

```
    \newrobustcmd*{\glsFindWidestAnyName}[1][\@glo@types]{%
      \dimen@=0pt\relax
      \gls@tmplen=0pt\relax
      \forallglossaries[#1]{\@gls@type}%
      {%
        \forglsentries[\@gls@type]{\@glo@label}%
        {%
          \glsmeasurewidth{\dimen@}%
           {\glstreenamefmt{\glsentryname{\@glo@label}}}%
```

```
            \ifdim\dimen@>\gls@tmplen
              \gls@tmplen=\dimen@
              \eglssetwidest{\glsentryname{\@glo@label}}%
            \fi
          }%
        }%
      }
```

\glsFindWidestUsedLevelTwo This is like \glsFindWidestUsedTopLevelName but also sets the first two sub-
levels as well. Any entry that has a great-grandparent is ignored.

```
\newrobustcmd*{\glsFindWidestUsedLevelTwo}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \dimen@i=0pt\relax
  \dimen@ii=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \ifglshasparent{\@glo@label}%
        {%

          \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@label}@parent}}%
          \ifglshasparent{\@glo@parent}%
          {%
            \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@parent}@parent}}%
            \ifglshasparent{\@glo@parent}%
            {}%
            {%
              \glsmeasurewidth{\gls@tmplen}%
                 {\glstreenamefmt{\glsentryname{\@glo@label}}}%
              \ifdim\gls@tmplen>\dimen@ii
                \dimen@ii=\gls@tmplen
                \eglssetwidest[2]{\glsentryname{\@glo@label}}%
              \fi
            }%
          }%
          {%
            \glsmeasurewidth{\gls@tmplen}%
               {\glstreenamefmt{\glsentryname{\@glo@label}}}%
            \ifdim\gls@tmplen>\dimen@i
              \dimen@i=\gls@tmplen
              \eglssetwidest[1]{\glsentryname{\@glo@label}}%
            \fi
          }%
        }%
        {%
          \glsmeasurewidth{\gls@tmplen}%
             {\glstreenamefmt{\glsentryname{\@glo@label}}}%
```

```
                        \ifdim\gls@tmplen>\dimen@
                          \dimen@=\gls@tmplen
                          \eglssetwidest{\glsentryname{\@glo@label}}}%
                        \fi
                      }%
                    }%
                    {}%
                  }%
                }%
              }
```

`\glsFindWidestLevelTwo` This is like `\glsFindWidestUsedLevelTwo` but doesn't check if the entry has been used.

```
\newrobustcmd*{\glsFindWidestLevelTwo}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \dimen@i=0pt\relax
  \dimen@ii=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglshasparent{\@glo@label}%
      {%
        \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@label}@parent}}%
        \ifglshasparent{\@glo@parent}%
        {%
          \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@parent}@parent}}%
          \ifglshasparent{\@glo@parent}%
          {}%
          {%
            \glsmeasurewidth{\gls@tmplen}%
                {\glstreenamefmt{\glsentryname{\@glo@label}}}%
            \ifdim\gls@tmplen>\dimen@ii
              \dimen@ii=\gls@tmplen
              \eglssetwidest[2]{\glsentryname{\@glo@label}}}%
            \fi
          }%
        }%
        {%
          \glsmeasurewidth{\gls@tmplen}%
              {\glstreenamefmt{\glsentryname{\@glo@label}}}%
          \ifdim\gls@tmplen>\dimen@i
            \dimen@i=\gls@tmplen
            \eglssetwidest[1]{\glsentryname{\@glo@label}}}%
          \fi
        }%
      }%
      {%
        \glsmeasurewidth{\gls@tmplen}%
```

```
                              {\glstreenamefmt{\glsentryname{\@glo@label}}}%
                        \ifdim\gls@tmplen>\dimen@
                          \dimen@=\gls@tmplen
                          \eglssetwidest{\glsentryname{\@glo@label}}%
                        \fi
                    }%
                  }%
                }%
              }
```

\glsFindWidestUsedAnyNameSymbol 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.

```
\newrobustcmd*{\glsFindWidestUsedAnyNameSymbol}[2][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \glsmeasurewidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
        \glsmeasurewidth{\dimen@}%
         {\glsentrysymbol{\@glo@label}}%
        \ifdim\dimen@>#2\relax
          #2=\dimen@
        \fi
      }%
      {}%
    }%
  }%
}
```

\glsFindWidestAnyNameSymbol Like the above but doesn't check if the entry has been used.

```
\newrobustcmd*{\glsFindWidestAnyNameSymbol}[2][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
```

```
                      \glsmeasurewidth{\dimen@}%
                       {\glstreenamefmt{\glsentryname{\@glo@label}}}%
                      \ifdim\dimen@>\gls@tmplen
                        \gls@tmplen=\dimen@
                        \eglssetwidest{\glsentryname{\@glo@label}}%
                      \fi
                      \glsmeasurewidth{\dimen@}%
                       {\glsentrysymbol{\@glo@label}}%
                      \ifdim\dimen@>#2\relax
                        #2=\dimen@
                      \fi
                    }%
                  }%
                }
```

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.

```
\newrobustcmd*{\glsFindWidestUsedAnyNameSymbolLocation}[3][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  #3=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \glsmeasurewidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
        \glsmeasurewidth{\dimen@}%
         {\glsentrysymbol{\@glo@label}}%
        \ifdim\dimen@>#2\relax
          #2=\dimen@
        \fi
        \glsmeasurewidth{\dimen@}%
         {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
        \ifdim\dimen@>#3\relax
          #3=\dimen@
        \fi
      }%
      {}%
```

```
    }%
  }%
}
```

Like the `\glsFindWidestUsedAnyNameSymbol` but doesn't check if the entry has been used.

```
\newrobustcmd*{\glsFindWidestAnyNameSymbolLocation}[3][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  #3=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \glsmeasurewidth{\dimen@}%
       {\glstreenamefmt{\glsentryname{\@glo@label}}}%
      \ifdim\dimen@>\gls@tmplen
        \gls@tmplen=\dimen@
        \eglssetwidest{\glsentryname{\@glo@label}}%
      \fi
      \glsmeasurewidth{\dimen@}%
       {\glsentrysymbol{\@glo@label}}%
      \ifdim\dimen@>#2\relax
        #2=\dimen@
      \fi
      \glsmeasurewidth{\dimen@}%
        {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
      \ifdim\dimen@>#3\relax
        #3=\dimen@
      \fi
    }%
  }%
}
```

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.

```
\newrobustcmd*{\glsFindWidestUsedAnyNameLocation}[2][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \glsmeasurewidth{\dimen@}%
```

696

```
                    {\glstreenamefmt{\glsentryname{\@glo@label}}}%
                \ifdim\dimen@>\gls@tmplen
                  \gls@tmplen=\dimen@
                  \eglssetwidest{\glsentryname{\@glo@label}}%
                \fi
                \glsmeasurewidth{\dimen@}%
                 {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
                \ifdim\dimen@>#2\relax
                  #2=\dimen@
                \fi
            }%
            {}%
          }%
        }%
      }
```

\glsFindWidestAnyNameLocation Like the \glsFindWidestAnyNameLocation but doesn't check the first use flag.

```
      \newrobustcmd*{\glsFindWidestAnyNameLocation}[2][\@glo@types]{%
        \dimen@=0pt\relax
        \gls@tmplen=0pt\relax
        #2=0pt\relax
        \forallglossaries[#1]{\@gls@type}%
        {%
          \forglsentries[\@gls@type]{\@glo@label}%
          {%
            \glsmeasurewidth{\dimen@}%
             {\glstreenamefmt{\glsentryname{\@glo@label}}}%
            \ifdim\dimen@>\gls@tmplen
              \gls@tmplen=\dimen@
              \eglssetwidest{\glsentryname{\@glo@label}}%
            \fi
            \glsmeasurewidth{\dimen@}%
             {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
            \ifdim\dimen@>#2\relax
              #2=\dimen@
            \fi
          }%
        }%
      }
```

\glsxtrComputeTreeIndent 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.

```
      \newcommand*{\glsxtrComputeTreeIndent}[1]{%
        \glstreeindent=\glsxtrtreetopindent\relax
      }
```

<div style="border:1px solid #ccc; background:#ffffcc; border-radius:8px; padding:8px">

\glsxtrComputeTreeSubIndent{⟨*level*⟩}{⟨*label*⟩}{⟨*register*⟩}

</div>

\glsxtrComputeTreeSubIndent

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.

```
\newcommand*{\glsxtrComputeTreeSubIndent}[3]{%
  \ifcsundef{@glswidestname\romannumeral#1}%
  {%
    \glsmeasurewidth{#3}{\glstreenamefmt{\@glswidestname\space}}%
  }%
  {%
    \glsmeasurewidth{#3}{\glstreenamefmt{%
            \csname @glswidestname\romannumeral#1\endcsname\space}}%
  }%
}
```

\glsxtrAltTreeSetHangIndent Set \hangindent for top-level entries:

```
\newcommand*{\glsxtrAltTreeSetHangIndent}{\hangindent\glstreeindent}
```

sxtrAltTreeSetSubHangIndent Set \hangindent for sub-entries:

```
\newcommand*{\glsxtrAltTreeSetSubHangIndent}[1]{\hangindent\glstreeindent}
```

Redefine alttree:

```
\renewglossarystyle{alttree}{%
  \renewenvironment{theglossary}%
    {%
     \glsxtralttreeInit
     \def\@gls@prevlevel{-1}%
     \mbox{}\par}%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}[4]{}%
  \renewcommand{\glossentry}[2]{%
    \ifnum\@gls@prevlevel=0\relax
    \else
      \glsxtrComputeTreeIndent{##1}%
    \fi
    \parindent\glstreeindent
    \glsxtrAltTreeSetHangIndent
    \makebox[0pt][r]%
    {%
      \glstreenamebox{\glstreeindent}%
      {%
          \glsentryitem{##1}%
          \glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}%
      }%
    }%
    \glsxtralttreeSymbolDescLocation{##1}{##2}%
```

698

```
                    \def\@gls@prevlevel{0}%
                }
            \renewcommand{\subglossentry}[3]{%
              \ifnum##1=1\relax
                \glssubentryitem{##2}%
              \fi
              \ifnum\@gls@prevlevel=##1\relax
              \else
                \glsxtrComputeTreeSubIndent{##1}{##2}{\gls@tmplen}%
                \ifnum\@gls@prevlevel<##1\relax
                  \setlength\glstreeindent\gls@tmplen
                  \addtolength\glstreeindent\parindent
                  \parindent\glstreeindent
                \else
                  \ifnum\@gls@prevlevel=0\relax
                    \glsxtrComputeTreeIndent{##2}%
                  \else
                    \glsxtrComputeTreeSubIndent{\@gls@prevlevel}{##2}{\glstreeindent}%
                  \fi
                  \addtolength\parindent{-\glstreeindent}%
                  \setlength\glstreeindent\parindent
                \fi
              \fi
              \glsxtrAltTreeSetSubHangIndent{##1}%
              \makebox[0pt][r]{\glstreenamebox{\gls@tmplen}{%
                \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
              \glsxtralttreeSubSymbolDescLocation{##1}{##2}{##3}%
              \def\@gls@prevlevel{##1}%
            }%
            \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
          }
    }%
    {%
    }
```

Redefine alttreegroup so that it discourages a break after group headings.

```
    \ifdef{\@glsstyle@alttreegroup}
    {
```

\glsalttreesubgroupitem

> **\glsalttreesubgroupheader{⟨*previous group level*⟩}{⟨*group level*⟩}{⟨*parent label*⟩}{⟨*group label*⟩}{⟨*title*⟩}{⟨*width*⟩}**

```
    \newcommand*{\glsalttreesubgroupheader}[6]{%
      \par\smallskip
      \makebox[0pt][r]{\glstreenamebox{#6}%
        {\glstreegroupheaderfmt{#5}}}%
      \smallskip\par
    }
```

```
\renewglossarystyle{alttreegroup}{%
  \setglossarystyle{alttree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
    \glstreePreHeader{##1}{\glsxtr@grptitle}%
    \glstreegroupheaderfmt{\glsxtr@grptitle}%
```

Can't use `\@afterheading` here as it messes with the first item of the group.

```
    \glstreegroupheaderskip
  }%
```

Sub-groups are only supported with `\printunsrtglossary`.

```
  \renewcommand*{\glssubgroupheading}[4]{%
    \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
    \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
```

This is similar to `\subglossentry`

```
    \ifnum\@gls@prevlevel=##2\relax
    \else
      \ifcsundef{@glswidestname\romannumeral##2}%
      {%
        \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
      }%
      {%
        \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
              \csname @glswidestname\romannumeral##2\endcsname\space}}%
      }%
      \ifnum\@gls@prevlevel<##2\relax
        \setlength\glstreeindent\gls@tmplen
        \addtolength\glstreeindent\parindent
        \parindent\glstreeindent
      \else
        \ifnum\@gls@prevlevel=0\relax
          \glsxtrComputeTreeIndent{##2}%
        \else
          \ifcsundef{@glswidestname\romannumeral##2}%
          {%
            \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
          }%
          {%
            \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
                  \csname @glswidestname\romannumeral##2\endcsname\space}}%
          }%
        \fi
        \addtolength\parindent{-\glstreeindent}%
        \setlength\glstreeindent\parindent
      \fi
    \fi
```

```
        \glsxtrAltTreeSetSubHangIndent{##2}%
        \glsalttreesubgroupheader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}{\gls@tmplen}%
        \def\@gls@prevlevel{##2}%
        \par
      }%
    }%
  }%
  {%
  }
```

Similarly for alttreehypergroup.

```
  \ifdef{\@glsstyle@alttreehypergroup}
  {%
    \renewglossarystyle{alttreehypergroup}{%
      \setglossarystyle{alttree}%
      \renewcommand*{\glossaryheader}{%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreenavigationfmt{\glsnavigation}%
```

Can't use `\@afterheading` here as it messes with the first item of the group.

```
        \glstreegroupheaderskip
      }%
      \renewcommand*{\glsgroupheading}[1]{%
        \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
```

Can't use `\@afterheading` here as it messes with the first item of the group.

```
        \glstreegroupheaderskip
      }%
```

Sub-groups are only supported with `\printunsrtglossary`.

```
      \renewcommand*{\glssubgroupheading}[4]{%
        \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
        \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
```

This is similar to \subglossentry

```
        \ifnum\@gls@prevlevel=##2\relax
        \else
          \ifcsundef{@glswidestname\romannumeral##2}%
          {%
            \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
          }%
          {%
            \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
```

```
                \csname @glswidestname\romannumeral##2\endcsname\space}}%
          }%
          \ifnum\@gls@prevlevel<##2\relax
            \setlength\glstreeindent\gls@tmplen
            \addtolength\glstreeindent\parindent
            \parindent\glstreeindent
          \else
            \ifnum\@gls@prevlevel=0\relax
              \glsxtrComputeTreeIndent{##2}%
            \else
              \ifcsundef{@glswidestname\romannumeral##2}%
              {%
                \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
              }%
              {%
                \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
                        \csname @glswidestname\romannumeral##2\endcsname\space}}%
              }%
            \fi
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
          \fi
        \fi
        \glsxtrAltTreeSetSubHangIndent{##2}%
        \glsalttreesubgroupheader{##1}{##2}{##3}{##4}%
          {\glsnavhypertarget{##4}{\glsxtr@grptitle}}{\gls@tmplen}%
        \def\@gls@prevlevel{##2}%
        \par
      }%
    }
  }%
  {%
  }
```

4.9   Multicolumn Styles

Adjust mcolindexgroup to discourage page breaks after the group headings.

```
  \ifdef{\@glsstyle@mcolindexgroup}
  {%
    \renewglossarystyle{mcolindexgroup}{%
      \setglossarystyle{mcolindex}%
```

Group heading as indexgroup.

```
      \renewcommand*{\glsgroupheading}[1]{%
        \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \item\glstreegroupheaderfmt{\glsxtr@grptitle}%
        \glstreegroupheaderskip\@afterheading
      }%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}[4]{%
      \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
      \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
      \glsindexsubgroupitem{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
      \@afterheading
    }%
  }
}%
{%
}
```

Similarly for mcolindexhypergroup.

```
  \ifdef{\@glsstyle@mcolindexhypergroup}
  {%
    \renewglossarystyle{mcolindexhypergroup}{%
      \setglossarystyle{mcolindex}%
      \renewcommand*{\glossaryheader}{%
        \item\glstreenavigationfmt{\glsnavigation}%

        \glstreegroupheaderskip\@afterheading
      }%
```

Group heading.

```
      \renewcommand*{\glsgroupheading}[1]{%
        \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \item\glstreegroupheaderfmt
          {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
        \glstreegroupheaderskip\@afterheading
      }%
```

Sub-groups are only supported with \printunsrtglossary.

```
      \renewcommand*{\glssubgroupheading}[4]{%
        \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
        \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
        \glsindexsubgroupitem{##1}{##2}{##3}{##4}%
          {\glsnavhypertarget{##4}{\glsxtr@grptitle}}%
        \@afterheading
      }%
    }
  }%
}%
{%
}
```

Similarly for mcolindexspannav.

```
  \ifdef{\@glsstyle@mcolindexspannav}
  {%
    \renewglossarystyle{mcolindexspannav}{%
      \setglossarystyle{index}%
      \renewenvironment{theglossary}%
```

```
     {%
      \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]%
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem}%
     {\end{multicols}}%
```

Group heading.

```
     \renewcommand*{\glsgroupheading}[1]{%
       \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
       \glstreePreHeader{##1}{\glsxtr@grptitle}%
       \item\glstreegroupheaderfmt
         {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
       \glstreegroupheaderskip\@afterheading
     }%
```

Sub-groups are only supported with \printunsrtglossary.

```
     \renewcommand*{\glssubgroupheading}[4]{%
       \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
       \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
       \glsindexsubgroupitem{##1}{##2}{##3}{##4}%
         {\glsnavhypertarget{##4}{\glsxtr@grptitle}}%
       \@afterheading
     }%
   }
 }%
 {%
 }
```

Similarly for mcoltreegroup.

```
 \ifdef{\@glsstyle@mcoltreegroup}
 {%
   \renewglossarystyle{mcoltreegroup}{%
     \setglossarystyle{mcoltree}%
```

Group heading.

```
     \renewcommand{\glsgroupheading}[1]{%
       \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
       \glstreePreHeader{##1}{\glsxtr@grptitle}%
       \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
       \glstreegroupheaderskip\@afterheading
     }%
```

Sub-groups are only supported with \printunsrtglossary.

```
     \renewcommand*{\glssubgroupheading}[4]{%
      \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
      \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
      \glstreesubgroupitem{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
     }%
   }
 }%
 {%
```

```
  }
```
Similarly for mcoltreehypergroup.
```
\ifdef{\@glsstyle@mcoltreehypergroup}
{%
  \renewglossarystyle{mcoltreehypergroup}{%
    \setglossarystyle{mcoltree}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }%
```
Group heading.
```
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading
    }%
```
Sub-groups are only supported with \printunsrtglossary.
```
    \renewcommand*{\glssubgroupheading}[4]{%
      \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
      \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
      \glstreesubgroupitem{##1}{##2}{##3}{##4}%
        {\glsnavhypertarget{##4}{\glsxtr@grptitle}}%
    }%
  }
}%
{%
}
```
Similarly for mcoltreespannav.
```
\ifdef{\@glsstyle@mcoltreespannav}
{%
  \renewglossarystyle{mcoltreespannav}{%
    \setglossarystyle{tree}%
    \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}%
        [\noindent\glstreenavigationfmt{\glsnavigation}]%
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
```
Group heading.
```
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
```

```
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading
    }%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}[4]{%
      \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
      \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
      \glstreesubgroupitem{##1}{##2}{##3}{##4}%
        {\glsnavhypertarget{##4}{\glsxtr@grptitle}}%
    }%
  }
 }%
 {%
 }
```

Similarly for mcoltreenonamegroup. There are no sub-groups for this style as it doesn't show the name of the child entries.

```
 \ifdef{\@glsstyle@mcoltreenonamegroup}
 {%
   \renewglossarystyle{mcoltreenonamegroup}{%
     \setglossarystyle{mcoltreenoname}%
     \renewcommand{\glsgroupheading}[1]{%
       \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
       \glstreePreHeader{##1}{\glsxtr@grptitle}%
       \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
       \glstreegroupheaderskip\@afterheading
     }%
   }
 }%
 {%
 }
```

Similarly for mcoltreenonamehypergroup.

```
 \ifdef{\@glsstyle@mcoltreenonamehypergroup}
 {%
   \renewglossarystyle{mcoltreenonamehypergroup}{%
     \setglossarystyle{mcoltreenoname}%
     \renewcommand*{\glossaryheader}{%
       \par\noindent\glstreenavigationfmt{\glsnavigation}%
       \glstreegroupheaderskip
     }%
     \renewcommand*{\glsgroupheading}[1]{%
       \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
       \glstreePreHeader{##1}{\glsxtr@grptitle}%
       \par\noindent
       \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
       \glstreegroupheaderskip\@afterheading}%
   }
 }%
 {%
```

```
  }
```
Similarly for mcoltreenonamespannav.
```
 \ifdef{\@glsstyle@mcoltreenonamespannav}
 {%
   \renewglossarystyle{mcoltreenonamespannav}{%
     \setglossarystyle{treenoname}%
     \renewenvironment{theglossary}%
     {%
        \begin{multicols}{\glsmcols}%
          [\noindent\glstreenavigationfmt{\glsnavigation}]%
        \setlength{\parindent}{0pt}%
        \setlength{\parskip}{0pt plus 0.3pt}%
     }%
     {\end{multicols}}%
     \renewcommand*{\glsgroupheading}[1]{%
       \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
       \glstreePreHeader{##1}{\glsxtr@grptitle}%
       \par\noindent
       \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
       \glstreegroupheaderskip\@afterheading}%
   }
 }%
 {%
 }
```
mcolalttree needs adjusting so that it uses \glsxtralttreeInit This doesn't use \mbox{}\par which would unbalance the top of the columns.
```
 \ifdef{\@glsstyle@mcolalttree}
 {%
   \renewglossarystyle{mcolalttree}{%
     \setglossarystyle{alttree}%
     \renewenvironment{theglossary}%
     {%
        \glsxtralttreeInit
        \def\@gls@prevlevel{-1}%
        \begin{multicols}{\glsmcols}%
     }%
     {\par\end{multicols}}%
   }
 }%
 {%
 }
```
Redefine mcolalttreegroup to discourage page breaks after the group headings.
```
 \ifdef{\@glsstyle@mcolalttreegroup}
 {%
   \renewglossarystyle{mcolalttreegroup}{%
     \setglossarystyle{mcolalttree}%
     \renewcommand{\glsgroupheading}[1]{%
       \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
```

```
            \glstreePreHeader{##1}{\glsxtr@grptitle}%
            \par
            \def\@gls@prevlevel{-1}%
            \hangindent0pt\relax
            \parindent0pt\relax
            \glstreegroupheaderfmt{\glsxtr@grptitle}%
            \glstreegroupheaderskip
        }%
```

Sub-groups are only supported with \printunsrtglossary.

```
        \renewcommand*{\glssubgroupheading}[4]{%
        \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
        \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
```

This is similar to \subglossentry

```
        \ifnum\@gls@prevlevel=##2\relax
        \else
          \ifcsundef{@glswidestname\romannumeral##2}%
          {%
            \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
          }%
          {%
            \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
                    \csname @glswidestname\romannumeral##2\endcsname\space}}%
          }%
          \ifnum\@gls@prevlevel<##2\relax
            \setlength\glstreeindent\gls@tmplen
            \addtolength\glstreeindent\parindent
            \parindent\glstreeindent
          \else
            \ifnum\@gls@prevlevel=0\relax
              \glsxtrComputeTreeIndent{##2}%
            \else
              \ifcsundef{@glswidestname\romannumeral##2}%
              {%
                \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
              }%
              {%
                \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
                        \csname @glswidestname\romannumeral##2\endcsname\space}}%
              }%
            \fi
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
          \fi
        \fi
        \glsxtrAltTreeSetSubHangIndent{##2}%
        \glsalttreesubgroupheader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}{\gls@tmplen}%
        \def\@gls@prevlevel{##2}%
        \par
        }%
```

```
    }
 }%
 {%
 }
```

Similarly for mcolalttreehypergroup.

```
 \ifdef{\@glsstyle@mcolalttreehypergroup}
 {%
   \renewglossarystyle{mcolalttreehypergroup}{%
     \setglossarystyle{mcolalttree}%
     \renewcommand*{\glossaryheader}{%
       \par
       \def\@gls@prevlevel{-1}%
       \hangindent0pt\relax
       \parindent0pt\relax
       \glstreenavigationfmt{\glsnavigation}%
       \glstreegroupheaderskip
     }%
     \renewcommand*{\glsgroupheading}[1]{%
       \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
       \glstreePreHeader{##1}{\glsxtr@grptitle}%
       \par
       \def\@gls@prevlevel{-1}%
       \hangindent0pt\relax
       \parindent0pt\relax
       \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
       \glstreegroupheaderskip
     }%
```

Sub-groups are only supported with \printunsrtglossary.

```
     \renewcommand*{\glssubgroupheading}[4]{%
       \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
       \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
```

This is similar to \subglossentry

```
       \ifnum\@gls@prevlevel=##2\relax
       \else
         \ifcsundef{@glswidestname\romannumeral##2}%
         {%
           \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
         }%
         {%
           \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
                 \csname @glswidestname\romannumeral##2\endcsname\space}}%
         }%
         \ifnum\@gls@prevlevel<##2\relax
           \setlength\glstreeindent\gls@tmplen
           \addtolength\glstreeindent\parindent
           \parindent\glstreeindent
         \else
           \ifnum\@gls@prevlevel=0\relax
```

```
                \glsxtrComputeTreeIndent{##2}%
              \else
                \ifcsundef{@glswidestname\romannumeral##2}%
                {%
                  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
                }%
                {%
                  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
                          \csname @glswidestname\romannumeral##2\endcsname\space}}%
                }%
              \fi
              \addtolength\parindent{-\glstreeindent}%
              \setlength\glstreeindent\parindent
            \fi
        \fi
        \glsxtrAltTreeSetSubHangIndent{##2}%
        \glsalttreesubgroupheader{##1}{##2}{##3}{##4}%
          {\glsnavhypertarget{##4}{\glsxtr@grptitle}}{\gls@tmplen}%
        \def\@gls@prevlevel{##2}%
        \par
      }%
    }
  }%
  {%
  }
```

Similarly for mcolalttreespannav.

```
  \ifdef{\@glsstyle@mcolalttreespannav}
  {%
    \renewglossarystyle{mcolalttreespannav}{%
      \setglossarystyle{alttree}%
      \renewenvironment{theglossary}%
      {%
        \glsxtralttreeInit
        \def\@gls@prevlevel{-1}%
        \begin{multicols}{\glsmcols}%
            [\noindent\glstreenavigationfmt{\glsnavigation}]%
      }%
      {\par\end{multicols}}%
      \renewcommand*{\glsgroupheading}[1]{%
        \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
        \glstreegroupheaderskip
      }%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}[4]{%
      \glsxtrgetgrouptitle{##4}{\glsxtr@grptitle}%
      \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
```

This is similar to \subglossentry

```
      \ifnum\@gls@prevlevel=##2\relax
      \else
        \ifcsundef{@glswidestname\romannumeral##2}%
        {%
          \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
        }%
        {%
          \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
                \csname @glswidestname\romannumeral##2\endcsname\space}}%
        }%
        \ifnum\@gls@prevlevel<##2\relax
          \setlength\glstreeindent\gls@tmplen
          \addtolength\glstreeindent\parindent
          \parindent\glstreeindent
        \else
          \ifnum\@gls@prevlevel=0\relax
            \glsxtrComputeTreeIndent{##2}%
          \else
            \ifcsundef{@glswidestname\romannumeral##2}%
            {%
              \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
            }%
            {%
              \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
                    \csname @glswidestname\romannumeral##2\endcsname\space}}%
            }%
          \fi
          \addtolength\parindent{-\glstreeindent}%
          \setlength\glstreeindent\parindent
        \fi
      \fi
      \glsxtrAltTreeSetSubHangIndent{##2}%
      \glsalttreesubgroupheader{##1}{##2}{##3}{##4}%
        {\glsnavhypertarget{##4}{\glsxtr@grptitle}}{\gls@tmplen}%
      \def\@gls@prevlevel{##2}%
      \par
    }%
  }
}%
{%
}
```

Reset the default style

```
\ifx\@glossary@default@style\relax
```

```
  \else
    \setglossarystyle{\@glsxtr@current@style}
  \fi
```

5   bookindex style (**glossary-bookindex.sty**)

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:
```
\DeclareRelease{v1.48}{2021-11-22}{glossary-bookindex-2021-11-22.sty}
\DeclareCurrentRelease{v1.7}{2025-05-14}
```

Declare package:
```
\ProvidesPackage{glossary-bookindex}[2025/05/14 v1.7 (NLCT)]
```

Load required packages.
```
\RequirePackage{multicol}
\RequirePackage{glossary-tree}
```

\glsxtrbookindexcols Number of columns.
```
\newcommand{\glsxtrbookindexcols}{2}
```

\glsxtrbookindextarget Create the target for top-level items.
```
\newcommand*{\glsxtrbookindextarget}[2]{\glstarget{#1}{#2}}
```

\glsxtrbookindexsubtarget Create the target for child items.
```
\newcommand*{\glsxtrbookindexsubtarget}[2]{\glsxtrbookindextarget{#1}{#2}}
```

\glsxtrbookindexname Format used for top-level entries. (Argument is the label.)
```
\newcommand*{\glsxtrbookindexname}[1]{\glossentryname{#1}}
```

\glsxtrbookindexsubname Format used for sub entries.
```
\newcommand*{\glsxtrbookindexsubname}[1]{\glsxtrbookindexname{#1}}
```

\glsxtrprelocation Provide in case glossaries-stylemods isn't loaded.
```
\providecommand*{\glsxtrprelocation}{\space}
```

\glsxtrbookindexprelocation Separator used before location list for top-level entries. Version 1.22 has removed
the \ifglsnopostdot check since this style doesn't display the description.
```
\newcommand*{\glsxtrbookindexprelocation}[1]{%
  \glsxtrifhasfield{location}{#1}%
  {,\glsxtrprelocation}%
  {\glsxtrprelocation}%
}
```

sxtrbookindexsubprelocation Separator used before location list for sub-entries.
```
\newcommand*{\glsxtrbookindexsubprelocation}[1]{%
 \glsxtrbookindexprelocation{#1}%
}
```

`\glsxtrbookindexlocation`

> `\glsxtrbookindexlocation{`⟨`label`⟩`}{`⟨`location`⟩`}`

Displays the location.

```
\newcommand*{\glsxtrbookindexlocation}[2]{#2}
```

`\glsxtrbookindexsublocation`

> `\glsxtrbookindexlocation{`⟨`label`⟩`}{`⟨`location`⟩`}`

Displays the location for sub-entries.

```
\newcommand*{\glsxtrbookindexsublocation}{\glsxtrbookindexlocation}
```

`\glsxtrbookindexparentchildsep` Separator used between top-level parent and child entry.

```
\newcommand{\glsxtrbookindexparentchildsep}{\nopagebreak}
```

`\glsxtrbookindexparentsubchildsep` Separator used between sub-level parent and child entry.

```
\newcommand{\glsxtrbookindexparentsubchildsep}{\glsxtrbookindexparentchildsep}
```

`\glsxtrbookindexbetween` Between two top-level entries identified by the labels in the arguments.

```
\newcommand{\glsxtrbookindexbetween}[2]{}
```

`\glsxtrbookindexsubbetween` Between two level 1 entries identified by the labels in the arguments.

```
\newcommand{\glsxtrbookindexsubbetween}[2]{}
```

`\glsxtrbookindexsubsubbetween` Between two level 2 entries identified by the labels in the arguments.

```
\newcommand{\glsxtrbookindexsubsubbetween}[2]{}
```

`\glsxtrbookindexatendgroup` At the end of a letter group. The argument is the label of the last top-level entry.

```
\newcommand{\glsxtrbookindexatendgroup}[1]{}
```

`\glsxtrbookindexsubatendgroup` At the end of a letter group. The argument is the label of the last level 1 entry.

```
\newcommand{\glsxtrbookindexsubatendgroup}[1]{}
```

`\glsxtrbookindexsubsubatendgroup` At the end of a letter group. The argument is the label of the last level 2 entry.

```
\newcommand{\glsxtrbookindexsubsubatendgroup}[1]{}
```

`\glsxtrbookindexgroupskip` Group separator.

```
\newcommand{\glsxtrbookindexgroupskip}{\ifglsnogroupskip\else\indexspace\fi}
```

`\glsxtrbookindexpregroupskip` After group header. The argument is the skip that would normally be inserted if there wasn't a group header.

```
\newcommand{\glsxtrbookindexpregroupskip}[1]{#1}
```

`\glsxtrbookindexpostgroupskip` After group header.

```
\newcommand{\glsxtrbookindexpostgroupskip}{\indexspace}
```

713

sxtrbookindexpresubgroupskip

> \glsxtrbookindexpresubgroupskip{⟨*default*⟩}{⟨*prev group
> level*⟩}{⟨*group level*⟩}

Before sub-group separator. The first argument is the skip that would normally be used at this point if there wasn't a header.

```
\newcommand{\glsxtrbookindexpresubgroupskip}[3]{\par\medskip}
```

xtrbookindexpostsubgroupskip After sub-group separator.

```
\newcommand{\glsxtrbookindexpostsubgroupskip}[2]{\par\medskip}
```

\glsxtrbookindexsubsubitem Sub-sub item and lower. The argument is the level, which will be 2 or more.

```
\newcommand{\glsxtrbookindexsubsubitem}[1]{\glstreesubsubitem}
```

Format group title.

\glsxtrbookindexformatheader Group header.

```
\newcommand*{\glsxtrbookindexformatheader}[1]{%
  \par{\centering\glstreegroupheaderfmt{#1}\par}%
}
```

Format sub-group title.

sxtrbookindexformatsubheader Sub-group header. This defaults to the same format as the top-level group.

```
\newcommand*{\glsxtrbookindexformatsubheader}[5]{%
 \ifnum#2>1\relax
  \glsxtrbookindexsubsubitem{#2}\glstreegroupheaderfmt{#5}%
 \else
  \glstreesubitem\glstreegroupheaderfmt{#5}%
 \fi
}
```

\glsxtrbookindexbookmark Book mark group heading if supported.

```
\ifdef\pdfbookmark
{%
  \newcommand*{\glsxtrbookindexbookmark}[2]{%
    \ifdefstring{\@@glossarysec}{chapter}%
    {\pdfbookmark[1]{#1}{#2}}%
    {\pdfbookmark[2]{#1}{#2}}%
  }
}
{%
  \newcommand*{\glsxtrbookindexbookmark}[2]{}
}
```

\glsxtrbookindexsubbookmark Book mark sub-group heading if supported.

```
\ifdef\pdfbookmark
{%
  \newcommand*{\glsxtrbookindexsubbookmark}[3]{%
```

714

```
        \ifdefstring{\@@glossarysec}{chapter}%
        {\expandafter\pdfbookmark\expandafter[\number\numexpr#1+1]{#3}{#2}}%
        {\expandafter\pdfbookmark\expandafter[\number\numexpr#1+2]{#3}{#2}}%
     }
  }
  {%
     \newcommand*{\glsxtrbookindexsubbookmark}[3]{}
  }
```

\glsxtrbookindexbookmarkprefix Make the bookmark label prefix used for letter groups depend on the glossary
label (instead of original hardcoded "index.").

```
\newcommand*{\glsxtrbookindexbookmarkprefix}{\currentglossary.}
```

\glsxtrbookindexcolspread

```
\newcommand*{\glsxtrbookindexcolspread}{}
```

\glsxtrbookindexmulticolsenv

```
\newcommand*{\glsxtrbookindexmulticolsenv}{multicols}
```

bookindex Define the style.

```
\newglossarystyle{bookindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
  {%
    \ifnum\glsxtrbookindexcols>1\relax
      \ifdefempty\glsxtrbookindexcolspread
      {%
        \edef\glsxtr@beginbookindex{%
          \noexpand\begin{\glsxtrbookindexmulticolsenv}
            {\glsxtrbookindexcols}%
        }%
      }%
      {%
        \edef\glsxtr@beginbookindex{%
          \noexpand\begin{\glsxtrbookindexmulticolsenv}%
            {\glsxtrbookindexcols}[\glsxtrbookindexcolspread]%
        }%
      }%
    \else
      \def\glsxtr@beginbookindex{}%
    \fi
    \glsxtr@beginbookindex
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
    \let\@glsxtr@bookindex@sep\glsxtrbookindexparentchildsep
    \let\@glsxtr@bookindex@subsep\glsxtrbookindexparentsubchildsep
    \let\@glsxtr@bookindex@between\@gobble
    \let\@glsxtr@bookindex@subbetween\@gobble
    \let\@glsxtr@bookindex@subsubbetween\@gobble
```

715

```
    \let\@glsxtr@bookindex@atendgroup\relax
    \let\@glsxtr@bookindex@subatendgroup\relax
    \let\@glsxtr@bookindex@subsubatendgroup\relax
    \let\@glsxtr@bookindexgroupskip\relax
  }%
  {%
```

Do end group hooks.

```
      \@glsxtr@bookindex@subsubatendgroup
      \@glsxtr@bookindex@subatendgroup
      \@glsxtr@bookindex@atendgroup
```

End multicols environment.

```
    \ifnum\glsxtrbookindexcols>1\relax
      \edef\glsxtr@endbookindex{%
        \noexpand\end{\glsxtrbookindexmulticolsenv}%
      }%
    \else
      \def\glsxtr@endbookindex{}%
    \fi
    \glsxtr@endbookindex
  }%
```

Use ragged right as columns are likely to be narrow and indexes tend not to be
fully justified.

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

Top level entry format.

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

Do separator.

```
      \@glsxtr@bookindex@between{##1}%
```

Update separators.

```
      \let\@glsxtr@bookindex@sep\glsxtrbookindexparentchildsep
      \let\@glsxtr@bookindex@subsep\glsxtrbookindexparentsubchildsep
      \let\@glsxtr@bookindex@subbetween\@gobble
      \let\@glsxtr@bookindex@subsubbetween\@gobble
```

The second argument of \glsxtrbookindexbetween will be supplied as the
argument to \@glsxtr@bookindex@between.

```
      \protected@edef\@glsxtr@bookindex@between{%
        \noexpand\glsxtrbookindexbetween{##1}%
      }%
      \protected@edef\@glsxtr@bookindex@atendgroup{%
        \noexpand\glsxtrbookindexatendgroup{##1}%
      }%
      \let\@glsxtr@bookindex@subatendgroup\relax
      \let\@glsxtr@bookindex@subsubatendgroup\relax
```

Format entry.

```
      \glstreeitem
        \glsentryitem{##1}%
```

```
        \glsxtrbookindextarget{##1}{\glsxtrbookindexname{##1}}%
      \glsxtrbookindexprelocation{##1}%
      \glsxtrbookindexlocation{##1}{##2}%
    }%
    \renewcommand{\subglossentry}[3]{%
      \ifcase##1\relax
```

Level 0 (shouldn't happen as that's formatted with \glossentry).

```
        \glstreeitem
      \or
```

Level 1.

```
        \@glsxtr@bookindex@sep
        \@glsxtr@bookindex@subbetween{##2}%
        \let\@glsxtr@bookindex@sep\relax
```

Update separators.

```
        \let\@glsxtr@bookindex@subsubbetween\@gobble
        \let\@glsxtr@bookindex@subsep\glsxtrbookindexparentsubchildsep
        \edef\@glsxtr@bookindex@subbetween{%
            \noexpand\glsxtrbookindexsubbetween{##2}%
        }%
        \edef\@glsxtr@bookindex@atsubendgroup{%
          \noexpand\glsxtrbookindexatsubendgroup{##1}%
        }%
```

Start sub-item.

```
        \glstreesubitem
        \glssubentryitem{##2}%
      \else
```

All other levels.

```
        \@glsxtr@bookindex@subsep
        \@glsxtr@bookindex@subsubbetween{##2}%
```

Update separators.

```
        \let\@glsxtr@bookindex@subsep\relax
        \edef\@glsxtr@bookindex@subsubbetween{%
            \noexpand\glsxtrbookindexsubsubbetween{##2}%
        }%
        \edef\@glsxtr@bookindex@atsubsubendgroup{%
          \noexpand\glsxtrbookindexatsubsubendgroup{##1}%
        }%
```

Start sub-sub-item.

```
        \glsxtrbookindexsubsubitem{##1}%
      \fi
```

Format entry.

```
      \glsxtrbookindexsubtarget{##2}{\glsxtrbookindexsubname{##2}}%
      \glsxtrbookindexsubprelocation{##2}%
      \glsxtrbookindexsublocation{##2}{##3}%
    }%
```

The group skip is moved to the group heading to avoid interfering with the end letter group hooks.

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

Group heading format.

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

Do end group hooks.

```
    \@glsxtr@bookindex@subsubatendgroup
    \@glsxtr@bookindex@subatendgroup
    \@glsxtr@bookindex@atendgroup
    \glsxtrbookindexpregroupskip\@glsxtr@bookindexgroupskip
```

Update separators.

```
    \let\@glsxtr@bookindexgroupskip\glsxtrbookindexgroupskip
    \let\@glsxtr@bookindex@between\@gobble
    \let\@glsxtr@bookindex@atendgroup\relax
    \let\@glsxtr@bookindex@subatendgroup\relax
    \let\@glsxtr@bookindex@subsubatendgroup\relax
```

Fetch the group title from the label supplied in #1.

```
    \glsxtrgetgrouptitle{##1}{\glsxtrcurrentgrptitle}%
```

Do the PDF bookmark if supported.

```
    \glsxtrbookindexbookmark{\glsxtrcurrentgrptitle}{\glsxtrbookindexbookmarkprefix##1}%
```

Format the group title.

```
    \glsxtrbookindexformatheader{\glsxtrcurrentgrptitle}%
    \nopagebreak\glsxtrbookindexpostgroupskip\nopagebreak\@afterheading
  }%
```

Sub-groups are only supported with \printunsrtglossary.

```
\renewcommand*{\glssubgroupheading}[4]{%
```

Do end group hooks.

```
    \@glsxtr@bookindex@subsubatendgroup
    \@glsxtr@bookindex@subatendgroup
    \@glsxtr@bookindex@atendgroup
    \glsxtrbookindexpresubgroupskip\@glsxtr@bookindexgroupskip{##1}{##2}%
```

Update separators.

```
    \let\@glsxtr@bookindexgroupskip\glsxtrbookindexgroupskip
    \let\@glsxtr@bookindex@between\@gobble
    \let\@glsxtr@bookindex@atendgroup\relax
    \let\@glsxtr@bookindex@subatendgroup\relax
    \let\@glsxtr@bookindex@subsubatendgroup\relax
```

Get group title.

```
    \glsxtrgetgrouptitle{##4}{\glsxtrcurrentgrptitle}%
```

Do the PDF bookmark if supported.

```
    \glsxtrbookindexsubbookmark{##2}{##4}{\glsxtrcurrentgrptitle}%
```

Format the group title.

```
      \glsxtrbookindexformatsubheader{##1}{##2}{##3}{##4}{\glsxtrcurrentgrptitle}%
      \nopagebreak\glsxtrbookindexpostsubgroupskip{##1}{##2}\nopagebreak\@afterheading
   }
 }
```

Some supplementary commands that may be useful. These store the entry label for the current page. Since the page number is needed in the control sequence, this uses `\glsxtrbookindexthepage` instead of `\thepage` in case the page numbering has been set to something that contains formatting commands.

`\glsxtrbookindexthepage` The `\@printglossary` sets `\currentglossary` to the current glossary label. This is used as a prefix in case the page number is reset.

```
\newcommand{\glsxtrbookindexthepage}{%
 \ifdef\currentglossary{\currentglossary.\arabic{page}}{\arabic{page}}%
}
```

`\glsxtrbookindexmarkentry` Writes entry information to the `.aux` file. The argument is the entry label.

```
\newcommand*{\glsxtrbookindexmarkentry}[1]{%
  \protected@write\@auxout
  {\let\glsxtrbookindexthepage\relax}%
  {\string\glsxtr@setbookindexmark{\glsxtrbookindexthepage}{#1}}%
}
```

`\glsxtr@setbookindexmark`

```
\newcommand*{\glsxtr@setbookindexmark}[2]{%
 \ifcsundef{glsxtr@idxfirstmark@#1}%
 {\csgdef{glsxtr@idxfirstmark@#1}{#2}}%
 {}%
 \csgdef{glsxtr@idxlastmark@#1}{#2}%
}
```

`\glsxtrbookindexfirstmarkfmt`

```
\newcommand*{\glsxtrbookindexfirstmarkfmt}[1]{%
  \glsentryname{#1}%
}
```

`\glsxtrbookindexfirstmark`

```
\newcommand*{\glsxtrbookindexfirstmark}{%
 \letcs{\glsxtr@label}{glsxtr@idxfirstmark@\glsxtrbookindexthepage}%
 \ifdef\glsxtr@label
 {\glsxtrbookindexfirstmarkfmt{\glsxtr@label}}%
 {}%
}
```

`\glsxtrbookindexlastmarkfmt`

```
\newcommand*{\glsxtrbookindexlastmarkfmt}[1]{%
 \glsentryname{#1}%
}
```

`\glsxtrbookindexlastmark`

```
\newcommand*{\glsxtrbookindexlastmark}{%
 \letcs{\glsxtr@label}{glsxtr@idxlastmark@\glsxtrbookindexthepage}%
 \ifdef\glsxtr@label
 {\glsxtrbookindexlastmarkfmt{\glsxtr@label}}%
 {}%
}
```

6   longextra styles (**glossary-longextra.sty**)

Provides additional long styles.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v1.48}{2021-11-22}{glossary-longextra-2021-11-22.sty}
\DeclareCurrentRelease{v1.7}{2025-05-14}
```

Declare package:

```
\ProvidesPackage{glossary-longextra}[2025/05/14 v1.7 (NLCT)]
```

Load required packages.

```
\RequirePackage{glossary-longbooktabs}
```

`\glslongextraNameFmt`

> `\glslongextraNameFmt{`⟨*label*⟩`}`

Governs the way the name is displayed.

```
\newcommand{\glslongextraNameFmt}[1]{%
 \glsentryitem{#1}\glstarget{#1}{\glossentryname{#1}}%
}
```

`\glslongextraDescFmt`

> `\glslongextraDescFmt{`⟨*label*⟩`}`

Governs the way the description is displayed.

```
\newcommand{\glslongextraDescFmt}[1]{%
 \glossentrydesc{#1}\glspostdescription
}
```

`\glslongextraSymbolFmt`

> `\glslongextraSymbolFmt{`⟨*label*⟩`}`

Governs the way the symbol is displayed.

```
\newcommand{\glslongextraSymbolFmt}[1]{\glossentrysymbol{#1}}
```

\glslongextraSymbolTargetFmt

> \glslongextraSymbolTargetFmt{⟨*label*⟩}

Governs the way the symbol is displayed if it needs to include the target.

```
\newcommand{\glslongextraSymbolTargetFmt}[1]{%
 \glsentryitem{#1}\glstarget{#1}{\glslongextraSymbolFmt{#1}}}
```

\glslongextraSymbolOrName

> \glslongextraSymbolOrName{⟨*label*⟩}

Governs the way the symbol is displayed if it needs to include the target.

```
\newcommand{\glslongextraSymbolOrName}[1]{%
  \ifglshassymbol{#1}%
  {\glslongextraSymbolTargetFmt{#1}}%
  {\glslongextraNameFmt{#1}}%
}
```

\glslongextraLocationFmt

> \glslongextraLocationFmt{⟨*label*⟩}{⟨*location list*⟩}

Governs the way the location is displayed.

```
\newcommand{\glslongextraLocationFmt}[2]{#2}
```

\glslongextraShortTargetFmt

> \glslongextraShortTargetFmt{⟨*label*⟩}

Governs the way the short form is displayed if it needs to include the target.

```
\newcommand{\glslongextraShortTargetFmt}[1]{%
 \glsentryitem{#1}\glstarget{#1}{{\glsxtrshort[noindex,hyper=false]{#1}}}%
 \glsxtrpostnamehook{#1}%
}
```

\glslongextraLongFmt

> \glslongextraLongFmt{⟨*label*⟩}

Governs the way the long form is displayed.

```
\newcommand{\glslongextraLongFmt}[1]{%
 {\glsxtrlong[noindex,hyper=false]{#1}}\glspostdescription
}
```

\glslongextraSubNameFmt

> \glslongextraSubNameFmt{⟨*level*⟩}{⟨*label*⟩}

Governs the way the child name is displayed. Just does the sub-entry counter, if enabled, and the target.

```
\newcommand{\glslongextraSubNameFmt}[2]{%
```

```
\glssubentryitem{#2}\glstarget{#2}{\strut}%
}
```

`\glslongextraSubDescFmt{⟨level⟩}{⟨label⟩}`

`\glslongextraSubDescFmt`

Governs the way the child description is displayed.

```
\newcommand{\glslongextraSubDescFmt}[2]{%
  \glslongextraDescFmt{#2}%
}
```

`\glslongextraSubSymbolFmt{⟨level⟩}{⟨label⟩}`

`\glslongextraSubSymbolFmt`

Governs the way the child symbol is displayed.

```
\newcommand{\glslongextraSubSymbolFmt}[2]{%
  \glslongextraSymbolFmt{#2}%
}
```

`\glslongextraSubSymbolTargetFmt{⟨level⟩}{⟨label⟩}`

`\glslongextraSubSymbolTargetFmt`

Governs the way the child symbol is displayed if the target is required.

```
\newcommand{\glslongextraSubSymbolTargetFmt}[2]{%
 \glssubentryitem{#2}\glstarget{#2}{\glslongextraSymbolFmt{#2}}%
}
```

`\glslongextraSubSymbolOrName{⟨level⟩}{⟨label⟩}`

`\glslongextraSubSymbolOrName`

Shows the symbol or the name (if the symbol isn't set) as the target for sub-entries.

```
\newcommand{\glslongextraSubSymbolOrName}[2]{%
  \ifglshassymbol{#2}%
  {\glslongextraSubSymbolTargetFmt{#1}{#2}}%
  {\glslongextraSubNameFmt{#1}{#2}}%
}
```

`\glslongextraSubShortTargetFmt{⟨level⟩}{⟨label⟩}`

`\glslongextraSubShortTargetFmt`

Governs the way the short form is displayed if it needs to include the target.

```
\newcommand{\glslongextraSubShortTargetFmt}[2]{%
 \glssubentryitem{#2}\glstarget{#2}{{\glsxtrshort[noindex,hyper=false]{#2}}}%
 \glsxtrpostnamehook{#2}%
}
```

`\glslongextraSubLongFmt`

> `\glslongextraSubLongFmt{`⟨*label*⟩`}`

Governs the way the long form is displayed.

`\newcommand{\glslongextraSubLongFmt}[2]{\glslongextraLongFmt{#2}}`

`\glslongextraSubLocationFmt`

> `\glslongextraSubLocationFmt{`⟨*level*⟩`}{`⟨*label*⟩`}{`⟨*location*`
> `*list*⟩`}`

Governs the way the child location list is displayed.

`\newcommand{\glslongextraSubLocationFmt}[3]{#3}`

`\glslongextraNameAlign` Alignment for the name column.

`\newcommand{\glslongextraNameAlign}{l}`

`\glslongextraDescAlign` Alignment for the description column.

`\newcommand{\glslongextraDescAlign}{>{\raggedright}p{\glsdescwidth}}`

`\glslongextraSymbolAlign` Alignment for the symbol column.

`\newcommand{\glslongextraSymbolAlign}{c}`

`\glslongextraSymbolNameAlign` Alignment for the symbol column when it's being used instead of the name.

`\newcommand{\glslongextraSymbolNameAlign}{l}`

`\glslongextraLocationAlign` Alignment for the location column.

`\newcommand{\glslongextraLocationAlign}{>{\raggedright}p{\glspagelistwidth}}`

`\glslongextraGroupHeading` Used to format the letter group headings. The first argument is the number of columns in the table. The second is the group *label* (not the title).

`\newcommand{\glslongextraGroupHeading}[2]{}`

`\glslongextraSubGroupHeading`

> `\glslongextraSubGroupHeading{`⟨*number of columns*⟩`}{`⟨*prev*
> *group level*⟩`}{`⟨*group*
> *level*⟩`}{`⟨*parent entry*⟩`}{`⟨*group label*⟩`}`

`\newcommand*{\glslongextraSubGroupHeading}[5]{}`

`\glslongextraHeaderFormat` Format for the column headers.

`\newcommand{\glslongextraHeaderFmt}[1]{\textbf{#1}}`

`\glslongextraNameDescHeader`

`\newcommand{\glslongextraNameDescHeader}{%`
` \glslongextraNameDescTabularHeader\endhead`
` \glslongextraNameDescTabularFooter\endfoot`
`}`

```
\newcommand{\glslongextraNameDescTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\descriptionname\tabularnewline
 \midrule
}
```

```
\newcommand{\glslongextraNameDescTabularFooter}{%
 \bottomrule
}
```

Unlike the alttree style, there aren't different widths for the hierarchical levels.

Provide in case the tree styles haven't been loaded.

```
\newcommand*{\glslongextraSetWidest}[1]{%
 \def\@glslongextrawidestname{#1}%
}
```

Pick up the widest name from the alttree style if it has been set. (Will expand to nothing otherwise.)

```
\newcommand*{\@glslongextrawidestname}{\csuse{@glswidestname}}
```

```
\newcommand*{\glslongextraUpdateWidest}[1]{%
 \ifundef\@glslongextrawidestname
 {\def\@glslongextrawidestname{#1}}%
 {%
   \glsmeasurewidth{\dimen@}{\@glslongextrawidestname}%
   \glsmeasurewidth{\dimen@ii}{#1}%
   \ifdim\dimen@ii>\dimen@
    \def\@glslongextrawidestname{#1}%
   \fi
 }%
}
```

> $\text{\glslongextraUpdateWidestChild}\{\langle \textit{level} \rangle\}\{\langle \textit{text} \rangle\}$

Used by \glsxtrSetWidest in glossaries-extra-bib2gls. Does nothing by default, since the default action in these styles is to omit the child name. If the child name should be displayed, then this needs to be redefined to use \glslongextraUpdateWidest.

```
\newcommand*{\glslongextraUpdateWidestChild}[2]{}
```

`\glslongextraSetDescWidth` Computes the value of `\glsdescwidth` for the styles that only have name and description columns.

```
\newcommand{\glslongextraSetDescWidth}{%
 \glsmeasurewidth{\gls@tmplen}{\glslongextraHeaderFmt\entryname}%
```

Has the widest name been set.

```
 \glsmeasurewidth{\dimen@}{\glsnamefont{\@glslongextrawidestname}}%
 \ifdim\dimen@>\gls@tmplen
  \gls@tmplen=\dimen@
 \fi
```

Description width is `\linewidth` less 4`\tabcolsep` less the width of the name column.

```
 \setlength{\glsdescwidth}{\dimexpr\linewidth-4\tabcolsep-\gls@tmplen}%
 }
```

`\glslongextraSymSetDescWidth` Computes the value of `\glsdescwidth` for the styles that only have name, symbol and description columns.

```
\newcommand{\glslongextraSymSetDescWidth}{%
```

Work out the size for just the name and description style.

```
 \glslongextraSetDescWidth
```

Now work out the symbol column width. This is assuming that the column title will be the widest text in the column.

```
 \glsmeasurewidth{\gls@tmplen}{\glslongextraHeaderFmt\symbolname}%
```

Subtract 2`\tabcolsep` and the symbol header width.

```
 \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\gls@tmplen}%
 }
```

`\glslongextraSymNoNameSetDescWidth` Computes the value of `\glsdescwidth` for the styles that only have symbol and description columns.

```
\newcommand{\glslongextraSymNoNameSetDescWidth}{%
```

Now work out the symbol column width. This is assuming that the column title will be the widest text in the column.

```
 \glsmeasurewidth{\gls@tmplen}{\glslongextraHeaderFmt\symbolname}%
```

Subtract 4`\tabcolsep` and the symbol header width.

```
 \setlength{\glsdescwidth}{\dimexpr\linewidth-4\tabcolsep-\gls@tmplen}%
 }
```

`\glslongextraLocSetDescWidth` Computes the value of `\glsdescwidth` for the styles that only have name, location and description columns.

```
\newcommand{\glslongextraLocSetDescWidth}{%
```

Work out the size for just the name and description style.

```
 \glslongextraSetDescWidth
```

Subtract 2`\tabcolsep` and the location list column width.

```
 \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\glspagelistwidth}%
 }
```

725

slongextraSymLocSetDescWidth Computes the value of \glsdescwidth for the styles that have name, symbol, location and description columns.

```
\newcommand{\glslongextraSymLocSetDescWidth}{%
```

Work out the size for just the name, symbol and description style.

```
    \glslongextraSymSetDescWidth
```

Subtract 2\tabcolsep and the location list column width.

```
    \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\glspagelistwidth}%
}
```

extraShortNoNameSetDescWidth Computes the value of \glsdescwidth for the styles that only have short and long columns. The long form will essentially be treated like a description column.

```
\newcommand{\glslongextraShortNoNameSetDescWidth}{%
```

Now work out the short column width. This is assuming that the column title will be the widest text in the column.

```
    \glsmeasurewidth{\gls@tmplen}{\glslongextraHeaderFmt\glslongextraShortHeader}%
```

Subtract 4\tabcolsep and the above header width.

```
    \setlength{\glsdescwidth}{\dimexpr\linewidth-4\tabcolsep-\gls@tmplen}%
}
```

\ifGlsLongExtraUseTabular If true use tabular instead of longtable. Obviously only intended for short glossaries that can fit into a single page.

```
\newif\ifGlsLongExtraUseTabular
\GlsLongExtraUseTabularfalse
```

\glslongextraTabularVAlign Only used with the tabular setting.

```
\newcommand*{\glslongextraTabularVAlign}{c}
```

long-name-desc Two column style with multi-lined descriptions and header. This is similar to the longragged-booktabs style.

```
\newglossarystyle{long-name-desc}%
{%
  \ifGlsLongExtraUseTabular
  \renewenvironment{theglossary}%
   {%
     \glslongextraSetDescWidth
     \edef\@glslongextra@begintab{%
       \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
         \expandonce\glslongextraNameAlign
         \expandonce\glslongextraDescAlign}}%
     \@glslongextra@begintab
   }%
   {%
     \glslongextraNameDescTabularFooter
     \end{tabular}%
   }%
  \renewcommand*{\glossaryheader}{\glslongextraNameDescTabularHeader}%
```

```
  \else
   \renewenvironment{theglossary}%
    {%
       \glspatchLToutput
       \glslongextraSetDescWidth
       \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
             \expandonce\glslongextraNameAlign
             \expandonce\glslongextraDescAlign}}%
       \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraNameDescHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{2}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{2}}%
```

Top-level entry.

```
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraDescFmt{##1}\tabularnewline
  }%
```

Child entry.

```
  \renewcommand{\subglossentry}[3]{%
     \glslongextraSubNameFmt{##1}{##2}
     &
     \glslongextraSubDescFmt{##1}{##2}%
     \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
 }
```

```
\newcommand{\glslongextraNameDescLocationHeader}{%
 \glslongextraNameDescLocationTabularHeader\endhead
 \glslongextraNameDescLocationTabularFooter\endfoot
}
```

```
\newcommand{\glslongextraNameDescLocationTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\pagelistname\tabularnewline
```

```
                              \midrule
                              }
```

```
                              \newcommand{\glslongextraNameDescLocationTabularFooter}{%
                              \bottomrule
                              }
```

long-name-desc-loc Three columns: name, description and location list.

```
                              \newglossarystyle{long-name-desc-loc}%
                              {%
                                \ifGlsLongExtraUseTabular
                                 \renewenvironment{theglossary}%
                                  {%
                                    \glslongextraLocSetDescWidth
                                    \edef\@glslongextra@begintab{%
                                      \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                                        \expandonce\glslongextraNameAlign
                                        \expandonce\glslongextraDescAlign
                                        \expandonce\glslongextraLocationAlign
                                      }}%
                                      \@glslongextra@begintab
                                  }%
                                  {%
                                    \glslongextraNameDescLocationTabularFooter
                                    \end{tabular}%
                                  }%
                                 \renewcommand*{\glossaryheader}{\glslongextraNameDescLocationTabularHeader}%
                                \else
                                 \renewenvironment{theglossary}%
                                  {%
                                    \glspatchLToutput
                                    \glslongextraLocSetDescWidth
                                    \edef\@glslongextra@begintab{%
                                      \noexpand\begin{longtable}{%
                                        \expandonce\glslongextraNameAlign
                                        \expandonce\glslongextraDescAlign
                                        \expandonce\glslongextraLocationAlign
                                      }}%
                                      \@glslongextra@begintab
                                  }%
                                  {\end{longtable}}%
                                 \renewcommand*{\glossaryheader}{\glslongextraNameDescLocationHeader}%
                                \fi
                                \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
                                \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
                                \renewcommand{\glossentry}[2]{%
                                  \glslongextraNameFmt{##1} &
```

```
                              \glslongextraDescFmt{##1} &
                              \glslongextraLocationFmt{##1}{##2}\tabularnewline
                        }%
                        \renewcommand{\subglossentry}[3]{%
                           \glslongextraSubNameFmt{##1}{##2}&
                           \glslongextraSubDescFmt{##1}{##2}&
                           \glslongextraSubLocationFmt{##1}{##2}{##3}%
                           \tabularnewline
                        }%
                        \ifglsnogroupskip
                           \renewcommand*{\glsgroupskip}{}%
                        \else
                           \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                        \fi
                     }
```

\glslongextraDescNameHeader

```
                     \newcommand{\glslongextraDescNameHeader}{%
                      \glslongextraDescNameTabularHeader\endhead
                      \glslongextraDescNameTabularFooter\endfoot
                     }
```

gextraDescNameTabularHeader

```
                     \newcommand{\glslongextraDescNameTabularHeader}{%
                      \toprule
                      \glslongextraHeaderFmt\descriptionname&
                      \glslongextraHeaderFmt\entryname \tabularnewline
                      \midrule
                     }
```

gextraDescNameTabularFooter

```
                     \newcommand{\glslongextraDescNameTabularFooter}{%
                      \bottomrule
                     }
```

long-desc-name Like name-desc but swaps the columns.

```
                     \newglossarystyle{long-desc-name}%
                     {%
                       \ifGlsLongExtraUseTabular
                       \renewenvironment{theglossary}%
                       {%
                          \glslongextraSetDescWidth
                          \edef\@glslongextra@begintab{%
                             \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                                \expandonce\glslongextraDescAlign
                                \expandonce\glslongextraNameAlign}}%
                          \@glslongextra@begintab
                       }%
                       {%
```

729

```
            \glslongextraDescNameTabularFooter
            \end{tabular}%
        }%
        \renewcommand*{\glossaryheader}{\glslongextraDescNameTabularHeader}%
      \else
       \renewenvironment{theglossary}%
        {%
          \glspatchLToutput
          \glslongextraSetDescWidth
          \edef\@glslongextra@begintab{%
            \noexpand\begin{longtable}{%
              \expandonce\glslongextraDescAlign
              \expandonce\glslongextraNameAlign}}%
          \@glslongextra@begintab
        }%
        {\end{longtable}}%
        \renewcommand*{\glossaryheader}{\glslongextraDescNameHeader}%
      \fi
      \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{2}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
      \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{2}}%
      \renewcommand{\glossentry}[2]{%
        \glslongextraDescFmt{##1} &
        \glslongextraNameFmt{##1}\tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
        \glslongextraSubDescFmt{##1}{##2} &
        \glslongextraSubNameFmt{##1}{##2}\tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
      \fi
    }
```

extraLocationDescNameHeader

```
\newcommand{\glslongextraLocationDescNameHeader}{%
 \glslongextraLocationDescNameTabularHeader\endhead
 \glslongextraLocationDescNameTabularFooter\endfoot
}
```

cationDescNameTabularHeader

```
\newcommand{\glslongextraLocationDescNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\pagelistname&
 \glslongextraHeaderFmt\descriptionname&
 \glslongextraHeaderFmt\entryname \tabularnewline
 \midrule
```

730

```
                                     }
```
```
                                  \newcommand{\glslongextraLocationDescNameTabularFooter}{%
                                   \bottomrule
                                  }
```
long-loc-desc-name Three columns: location, description and name.
```
                                  \newglossarystyle{long-loc-desc-name}%
                                  {%
                                    \ifGlsLongExtraUseTabular
                                    {%
                                      \glslongextraLocSetDescWidth
                                      \edef\@glslongextra@begintab{%
                                        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                                          \expandonce\glslongextraLocationAlign
                                          \expandonce\glslongextraDescAlign
                                          \expandonce\glslongextraNameAlign}}%
                                      \@glslongextra@begintab
                                    }%
                                    {%
                                      \glslongextraLocationDescNameTabularFooter
                                      \end{tabular}%
                                    }%
                                    \renewcommand*{\glossaryheader}{\glslongextraLocationDescNameTabularHeader}%
                                    \else
                                    \renewenvironment{theglossary}%
                                    {%
                                      \glspatchLToutput
                                      \glslongextraLocSetDescWidth
                                      \edef\@glslongextra@begintab{%
                                        \noexpand\begin{longtable}{%
                                          \expandonce\glslongextraLocationAlign
                                          \expandonce\glslongextraDescAlign
                                          \expandonce\glslongextraNameAlign}}%
                                      \@glslongextra@begintab
                                    }%
                                    {\end{longtable}}%
                                    \renewcommand*{\glossaryheader}{\glslongextraLocationDescNameHeader}%
                                    \fi
                                    \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
                              Sub-groups are only supported with \printunsrtglossary.
                                    \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
                                    \renewcommand{\glossentry}[2]{%
                                      \glslongextraLocationFmt{##1}{##2} &
                                      \glslongextraDescFmt{##1} &
                                      \glslongextraNameFmt{##1}\tabularnewline
                                    }%
                                    \renewcommand{\subglossentry}[3]{%
```

```
                              \glslongextraSubLocationFmt{##1}{##2}{##3} &
                              \glslongextraSubDescFmt{##1}{##2} &
                              \glslongextraSubNameFmt{##1}{##2}\tabularnewline
                         }%
                         \ifglsnogroupskip
                           \renewcommand*{\glsgroupskip}{}%
                         \else
                           \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                         \fi
                       }
```

\glslongextraNameDescSymHeader

```
                       \newcommand{\glslongextraNameDescSymHeader}{%
                        \glslongextraNameDescSymTabularHeader\endhead
                        \glslongextraNameDescSymTabularFooter\endfoot
                       }
```

\glslongextraNameDescSymTabularHeader

```
                       \newcommand{\glslongextraNameDescSymTabularHeader}{%
                        \toprule
                        \glslongextraHeaderFmt\entryname &
                        \glslongextraHeaderFmt\descriptionname &
                        \glslongextraHeaderFmt\symbolname\tabularnewline
                        \midrule
                       }
```

\glslongextraNameDescSymTabularFooter

```
                       \newcommand{\glslongextraNameDescSymTabularFooter}{%
                        \bottomrule
                       }
```

long-name-desc-sym Three column style with symbol in the third column.

```
                       \newglossarystyle{long-name-desc-sym}%
                       {%
                         \ifGlsLongExtraUseTabular
                         \renewenvironment{theglossary}%
                          {%
                            \glslongextraSymSetDescWidth
                            \edef\@glslongextra@begintab{%
                              \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                                \expandonce\glslongextraNameAlign
                                \expandonce\glslongextraDescAlign
                                \expandonce\glslongextraSymbolAlign
                              }}%
                            \@glslongextra@begintab
                          }%
                          {%
                            \glslongextraNameDescSymTabularFooter
                            \end{tabular}%
```

```
        }%
      \renewcommand*{\glossaryheader}{\glslongextraNameDescSymTabularHeader}%
    \else
     \renewenvironment{theglossary}%
      {%
        \glspatchLToutput
        \glslongextraSymSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraDescAlign
            \expandonce\glslongextraSymbolAlign
          }}%
        \@glslongextra@begintab
      }%
      {\end{longtable}}%
     \renewcommand*{\glossaryheader}{\glslongextraNameDescSymHeader}%
    \fi
    \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
    \renewcommand{\glossentry}[2]{%
      \glslongextraNameFmt{##1} &
      \glslongextraDescFmt{##1} &
      \glslongextraSymbolFmt{##1}\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      \glslongextraSubNameFmt{##1}{##2} &
      \glslongextraSubDescFmt{##1}{##2} &
      \glslongextraSubSymbolFmt{##1}{##2}%
      \tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
    \fi
  }
```

raNameDescSymLocationHeader

```
\newcommand{\glslongextraNameDescSymLocationHeader}{%
 \glslongextraNameDescSymLocationTabularHeader\endhead
 \glslongextraNameDescSymLocationTabularFooter\endfoot
}
```

DescSymLocationTabularHeader

```
\newcommand{\glslongextraNameDescSymLocationTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
```

733

```
                              \glslongextraHeaderFmt\descriptionname &
                              \glslongextraHeaderFmt\symbolname &
                              \glslongextraHeaderFmt\pagelistname\tabularnewline
                              \midrule
                              }
```

DescSymLocationTabularFooter

```
                              \newcommand{\glslongextraNameDescSymLocationTabularFooter}{%
                              \bottomrule
                              }
```

long-name-desc-sym-loc Four columns: name, description and location

```
                              \newglossarystyle{long-name-desc-sym-loc}%
                              {%
                                \ifGlsLongExtraUseTabular
                                 \renewenvironment{theglossary}%
                                  {%
                                     \glslongextraSymLocSetDescWidth
                                     \edef\@glslongextra@begintab{%
                                       \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                                         \expandonce\glslongextraNameAlign
                                         \expandonce\glslongextraDescAlign
                                         \expandonce\glslongextraSymbolAlign
                                         \expandonce\glslongextraLocationAlign
                                       }}%
                                     \@glslongextra@begintab
                                  }%
                                  {%
                                     \glslongextraNameDescSymLocationTabularFooter
                                     \end{tabular}%
                                  }%
                                 \renewcommand*{\glossaryheader}{\glslongextraNameDescSymLocationTabularHeader}%
                                \else
                                 \renewenvironment{theglossary}%
                                  {%
                                     \glspatchLToutput
                                     \glslongextraSymLocSetDescWidth
                                     \edef\@glslongextra@begintab{%
                                       \noexpand\begin{longtable}{%
                                         \expandonce\glslongextraNameAlign
                                         \expandonce\glslongextraDescAlign
                                         \expandonce\glslongextraSymbolAlign
                                         \expandonce\glslongextraLocationAlign
                                       }}%
                                     \@glslongextra@begintab
                                  }%
                                  {\end{longtable}}%
                                 \renewcommand*{\glossaryheader}{\glslongextraNameDescSymLocationHeader}%
                                \fi
                                \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraSymbolFmt{##1}&
    \glslongextraLocationFmt{##1}{##2}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubNameFmt{##1}{##2} &
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubSymbolFmt{##1}{##2}&
    \glslongextraSubLocationFmt{##1}{##2}{##3}%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
```

```
\newcommand{\glslongextraNameSymDescHeader}{%
\glslongextraNameSymDescTabularHeader\endhead
\glslongextraNameSymDescTabularFooter\endfoot
}
```

```
\newcommand{\glslongextraNameSymDescTabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt\symbolname &
\glslongextraHeaderFmt\descriptionname\tabularnewline
\midrule
}
```

```
\newcommand{\glslongextraNameSymDescTabularFooter}{%
\bottomrule
}
```

long-name-sym-desc Three column style with symbol in the second column.

```
\newglossarystyle{long-name-sym-desc}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSymSetDescWidth
```

735

```
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraDescAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraNameSymDescTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraNameSymDescTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraDescAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraNameSymDescHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraSymbolFmt{##1} &
    \glslongextraDescFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
     \glslongextraSubNameFmt{##1}{##2} &
     \glslongextraSubSymbolFmt{##1}{##2} &
     \glslongextraSubDescFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
```

```
\newcommand{\glslongextraNameSymDescLocationHeader}{%
 \glslongextraNameSymDescLocationTabularHeader\endhead
 \glslongextraNameSymDescLocationTabularFooter\endfoot
}
```

```
\newcommand{\glslongextraNameSymDescLocationTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\pagelistname\tabularnewline
 \midrule
}
```

```
\newcommand{\glslongextraNameSymDescLocationTabularFooter}{%
 \bottomrule
}
```

long-name-sym-desc-loc Four column style with symbol in the second column.

```
\newglossarystyle{long-name-sym-desc-loc}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
     {%
       \glslongextraSymLocSetDescWidth
       \edef\@glslongextra@begintab{%
         \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
           \expandonce\glslongextraNameAlign
           \expandonce\glslongextraSymbolAlign
           \expandonce\glslongextraDescAlign
           \expandonce\glslongextraLocationAlign
         }}%
       \@glslongextra@begintab
     }%
     {%
       \glslongextraNameSymDescLocationTabularFooter
       \end{tabular}%
     }%
    \renewcommand*{\glossaryheader}{\glslongextraNameSymDescLocationTabularHeader}%
  \else
    \renewenvironment{theglossary}%
     {%
       \glspatchLToutput
       \glslongextraSymLocSetDescWidth
       \edef\@glslongextra@begintab{%
         \noexpand\begin{longtable}{%
```

```
                \expandonce\glslongextraNameAlign
                \expandonce\glslongextraSymbolAlign
                \expandonce\glslongextraDescAlign
                \expandonce\glslongextraLocationAlign
            }}%
          \@glslongextra@begintab
        }%
        {\end{longtable}}%
      \renewcommand*{\glossaryheader}{\glslongextraNameSymDescLocationHeader}%
    \fi
    \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```
Sub-groups are only supported with \printunsrtglossary.
```
    \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
    \renewcommand{\glossentry}[2]{%
      \glslongextraNameFmt{##1} &
      \glslongextraSymbolFmt{##1} &
      \glslongextraDescFmt{##1} &
      \glslongextraLocationFmt{##1}{##2}\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      \glslongextraSubNameFmt{##1}{##2} &
      \glslongextraSubSymbolFmt{##1}{##2} &
      \glslongextraSubDescFmt{##1}{##2} &
      \glslongextraSubLocationFmt{##1}{##2}{##3}\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
    \fi
  }
```

slongextraSymDescNameHeader

```
\newcommand{\glslongextraSymDescNameHeader}{%
 \glslongextraSymDescNameTabularHeader\endhead
 \glslongextraSymDescNameTabularFooter\endfoot
}
```

traSymDescNameTabularHeader

```
\newcommand{\glslongextraSymDescNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\entryname\tabularnewline
 \midrule
}
```

traSymDescNameTabularFooter

```
\newcommand{\glslongextraSymDescNameTabularFooter}{%
```

```
                 \bottomrule
                }
```

long-sym-desc-name Three column style with symbol in the first column, description in the second
                    and name in the third.

```
\newglossarystyle{long-sym-desc-name}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSymSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraSymDescNameTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraSymDescNameTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraSymDescNameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraSymbolFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
```

```
                              \glslongextraSubSymbolFmt{##1}{##2} &
                              \glslongextraSubDescFmt{##1}{##2} &
                              \glslongextraSubNameFmt{##1}{##2}\tabularnewline
                          }%
                          \ifglsnogroupskip
                            \renewcommand*{\glsgroupskip}{}%
                          \else
                            \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                          \fi
                        }
```

traLocationSymDescNameHeader
```
                        \newcommand{\glslongextraLocationSymDescNameHeader}{%
                         \glslongextraLocationSymDescNameTabularHeader\endhead
                         \glslongextraLocationSymDescNameTabularFooter\endfoot
                        }
```

ionSymDescNameTabularHeader
```
                        \newcommand{\glslongextraLocationSymDescNameTabularHeader}{%
                         \toprule
                         \glslongextraHeaderFmt\pagelistname &
                         \glslongextraHeaderFmt\symbolname &
                         \glslongextraHeaderFmt\descriptionname &
                         \glslongextraHeaderFmt\entryname\tabularnewline
                         \midrule
                        }
```

ionSymDescNameTabularFooter
```
                        \newcommand{\glslongextraLocationSymDescNameTabularFooter}{%
                         \bottomrule
                        }
```

long-loc-sym-desc-name Four column style with location list, symbol, description and name.
```
                        \newglossarystyle{long-loc-sym-desc-name}%
                        {%
                          \ifGlsLongExtraUseTabular
                           \renewenvironment{theglossary}%
                           {%
                             \glslongextraSymLocSetDescWidth
                             \edef\@glslongextra@begintab{%
                               \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                                  \expandonce\glslongextraLocationAlign
                                  \expandonce\glslongextraSymbolAlign
                                  \expandonce\glslongextraDescAlign
                                  \expandonce\glslongextraNameAlign
                               }}%
                             \@glslongextra@begintab
                           }%
                           {%
```

```
        \glslongextraLocationSymDescNameTabularFooter
        \end{tabular}%
     }%
     \renewcommand*{\glossaryheader}{\glslongextraLocationSymDescNameTabularHeader}%
   \else
    \renewenvironment{theglossary}%
     {%
        \glspatchLToutput
        \glslongextraSymLocSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
             \expandonce\glslongextraLocationAlign
             \expandonce\glslongextraSymbolAlign
             \expandonce\glslongextraDescAlign
             \expandonce\glslongextraNameAlign
          }}%
        \@glslongextra@begintab
      }%
      {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraLocationSymDescNameHeader}%
   \fi
   \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
   \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
   \renewcommand{\glossentry}[2]{%
     \glslongextraLocationFmt{##1}{##2} &
     \glslongextraSymbolFmt{##1} &
     \glslongextraDescFmt{##1} &
     \glslongextraNameFmt{##1}\tabularnewline
   }%
   \renewcommand{\subglossentry}[3]{%
     \glslongextraSubLocationFmt{##1}{##2}{##3} &
     \glslongextraSubSymbolFmt{##1}{##2} &
     \glslongextraSubDescFmt{##1}{##2} &
     \glslongextraSubNameFmt{##1}{##2}\tabularnewline
   }%
   \ifglsnogroupskip
     \renewcommand*{\glsgroupskip}{}%
   \else
     \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
   \fi
 }
```

slongextraDescSymNameHeader

```
 \newcommand{\glslongextraDescSymNameHeader}{%
  \glslongextraDescSymNameTabularHeader\endhead
  \glslongextraDescSymNameTabularFooter\endfoot
 }
```

```
\newcommand{\glslongextraDescSymNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\entryname\tabularnewline
 \midrule
}
```

```
\newcommand{\glslongextraDescSymNameTabularFooter}{%
 \bottomrule
}
```

long-desc-sym-name Three column style with description in the first column, symbol in the second and name in the third.

```
\newglossarystyle{long-desc-sym-name}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSymSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraDescSymNameTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraDescSymNameTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
```

```
                              \renewcommand*{\glossaryheader}{\glslongextraDescSymNameHeader}%
                              \fi
                              \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```
Sub-groups are only supported with \printunsrtglossary.
```
                              \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
                              \renewcommand{\glossentry}[2]{%
                                \glslongextraDescFmt{##1} &
                                \glslongextraSymbolFmt{##1} &
                                \glslongextraNameFmt{##1}\tabularnewline
                              }%
                              \renewcommand{\subglossentry}[3]{%
                                \glslongextraSubDescFmt{##1}{##2} &
                                \glslongextraSubSymbolFmt{##1}{##2} &
                                \glslongextraSubNameFmt{##1}{##2}\tabularnewline
                              }%
                              \ifglsnogroupskip
                                \renewcommand*{\glsgroupskip}{}%
                              \else
                                \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                              \fi
                            }
```

traLocationDescSymNameHeader

```
                            \newcommand{\glslongextraLocationDescSymNameHeader}{%
                             \glslongextraLocationDescSymNameTabularHeader\endhead
                             \glslongextraLocationDescSymNameTabularFooter\endfoot
                            }
```

tionDescSymNameTabularHeader

```
                            \newcommand{\glslongextraLocationDescSymNameTabularHeader}{%
                             \toprule
                             \glslongextraHeaderFmt\pagelistname &
                             \glslongextraHeaderFmt\descriptionname &
                             \glslongextraHeaderFmt\symbolname &
                             \glslongextraHeaderFmt\entryname\tabularnewline
                             \midrule
                            }
```

tionDescSymNameTabularFooter

```
                            \newcommand{\glslongextraLocationDescSymNameTabularFooter}{%
                             \bottomrule
                            }
```

long-loc-desc-sym-name Four column style with location list, description, symbol and name.
```
                            \newglossarystyle{long-loc-desc-sym-name}%
                            {%
                              \ifGlsLongExtraUseTabular
                               \renewenvironment{theglossary}%
                                {%
```

```
      \glslongextraSymLocSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraLocationAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraLocationDescSymNameTabularFooter
      \end{tabular}%
    }%
  \renewcommand*{\glossaryheader}{\glslongextraLocationDescSymNameTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymLocSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraLocationAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraLocationDescSymNameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```
Sub-groups are only supported with \printunsrtglossary.
```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraLocationFmt{##1}{##2} &
    \glslongextraDescFmt{##1} &
    \glslongextraSymbolFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubLocationFmt{##1}{##2}{##3} &
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubSymbolFmt{##1}{##2} &
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
```

```
        \else
          \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
        \fi
      }
```

long-sym-desc Two column style with symbol in the first column and description in the second.
               The name isn't shown unless the symbol is missing.

```
\newglossarystyle{long-sym-desc}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
    {%
      \glslongextraSymNoNameSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraSymbolNameAlign
          \expandonce\glslongextraDescAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraSymDescTabularFooter
      \end{tabular}%
    }%
    \renewcommand*{\glossaryheader}{\glslongextraSymDescTabularHeader}}%
  \else
    \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymNoNameSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraSymbolNameAlign
          \expandonce\glslongextraDescAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraSymDescHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraSymbolOrName{##1} &
    \glslongextraDescFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubSymbolOrName{##1}{##2} &
```

745

```
                              \glslongextraSubDescFmt{##1}{##2}\tabularnewline
                          }%
                          \ifglsnogroupskip
                            \renewcommand*{\glsgroupskip}{}%
                          \else
                            \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                          \fi
                      }
```

\glslongextraSymDescHeader

```
                      \newcommand{\glslongextraSymDescHeader}{%
                       \glslongextraSymDescTabularHeader\endhead
                       \glslongextraSymDescTabularFooter\endfoot
                      }
```

ongextraSymDescTabularHeader

```
                      \newcommand{\glslongextraSymDescTabularHeader}{%
                       \toprule
                       \glslongextraHeaderFmt\symbolname &
                       \glslongextraHeaderFmt\descriptionname\tabularnewline
                       \midrule
                      }
```

ongextraSymDescTabularFooter

```
                      \newcommand{\glslongextraSymDescTabularFooter}{%
                       \bottomrule
                      }
```

long-desc-sym Two column style with description in the first column and symbol in the second.
The name isn't shown.

```
                      \newglossarystyle{long-desc-sym}%
                      {%
                        \ifGlsLongExtraUseTabular
                        \renewenvironment{theglossary}%
                         {%
                           \glslongextraSymNoNameSetDescWidth
                           \edef\@glslongextra@begintab{%
                             \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                               \expandonce\glslongextraDescAlign
                               \expandonce\glslongextraSymbolNameAlign
                             }}%
                           \@glslongextra@begintab
                         }%
                         {%
                           \glslongextraDescSymTabularFooter
                           \end{tabular}%
                         }%
                         \renewcommand*{\glossaryheader}{\glslongextraDescSymTabularHeader}%
                        \else
```

```
                            \renewenvironment{theglossary}%
                             {%
                                \glspatchLToutput
                                \glslongextraSymNoNameSetDescWidth
                                \edef\@glslongextra@begintab{%
                                   \noexpand\begin{longtable}{%
                                      \expandonce\glslongextraDescAlign
                                      \expandonce\glslongextraSymbolNameAlign
                                   }}%
                                \@glslongextra@begintab
                             }%
                             {\end{longtable}}%
                            \renewcommand*{\glossaryheader}{\glslongextraDescSymHeader}%
                          \fi
                          \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
                          \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
                          \renewcommand{\glossentry}[2]{%
                             \glslongextraDescFmt{##1} &
                             \glslongextraSymbolOrName{##1}\tabularnewline
                          }%
                          \renewcommand{\subglossentry}[3]{%
                             \glslongextraSubDescFmt{##1}{##2} &
                             \glslongextraSubSymbolOrName{##1}{##2}\tabularnewline
                          }%
                          \ifglsnogroupskip
                             \renewcommand*{\glsgroupskip}{}%
                          \else
                             \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                          \fi
                        }
```

\glslongextraDescSymHeader

```
                          \newcommand{\glslongextraDescSymHeader}{%
                          \glslongextraDescSymTabularHeader\endhead
                          \glslongextraDescSymTabularFooter\endfoot
                          }
```

ongextraDescSymTabularHeader

```
                          \newcommand{\glslongextraDescSymTabularHeader}{%
                          \toprule
                          \glslongextraHeaderFmt\descriptionname &
                          \glslongextraHeaderFmt\symbolname\tabularnewline
                          \midrule
                          }
```

ongextraDescSymTabularFooter

```
                          \newcommand{\glslongextraDescSymTabularFooter}{%
                          \bottomrule
```

```
      }
```

abbr-short-long Two column style with the short field in the first column and the long field in the second. The name, symbol and description aren't shown (although the abbreviation style may mean that they will happen to be shown if they are the same as the short or long field).

```
\newglossarystyle{abbr-short-long}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraShortNoNameSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraDescAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraShortLongTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraShortLongTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraShortNoNameSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraDescAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraShortLongHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \ifglshasshort{##1}%
    {\glslongextraShortTargetFmt{##1}}%
    {\glslongextraNameFmt{##1}}%
     &
    \ifglshaslong{##1}%
    {\glslongextraLongFmt{##1}}%
```

748

```
          {\glslongextraDescFmt{##1}}%
          \tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
          \ifglshasshort{##2}%
          {\glslongextraSubShortTargetFmt{##1}{##2}}%
          {\glslongextraSubNameFmt{##1}{##2}}%
           &
          \ifglshaslong{##2}%
          {\glslongextraSubLongFmt{##1}{##2}}%
          {\glslongextraSubDescFmt{##1}{##2}}%
          \tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
      \fi
    }
```

\glslongextraShortLongHeader

```
    \newcommand{\glslongextraShortLongHeader}{%
     \glslongextraShortLongTabularHeader\endhead
     \glslongextraShortLongTabularFooter\endfoot
    }
```

\glslongextraShortHeader

```
    \newcommand{\glslongextraShortHeader}{\entryname}
```

\glslongextraLongHeader

```
    \newcommand{\glslongextraLongHeader}{\descriptionname}
```

\glslongextraShortLongTabularHeader

```
    \newcommand{\glslongextraShortLongTabularHeader}{%
     \toprule
     \glslongextraHeaderFmt\glslongextraShortHeader &
     \glslongextraHeaderFmt\glslongextraLongHeader\tabularnewline
     \midrule
    }
```

\glslongextraShortLongTabularFooter

```
    \newcommand{\glslongextraShortLongTabularFooter}{%
     \bottomrule
    }
```

abbr-long-short   Two column style with the short field in the first column and the long field
                  in the second. The name, symbol and description aren't shown (although the
                  abbreviation style may mean that they will happen to be shown if they are the
                  same as the short or long field).

```
\newglossarystyle{abbr-long-short}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
    {%
      \glslongextraShortNoNameSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraLongShortTabularFooter
      \end{tabular}%
    }%
    \renewcommand*{\glossaryheader}{\glslongextraLongShortTabularHeader}%
  \else
    \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraShortNoNameSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraLongShortHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```
Sub-groups are only supported with \printunsrtglossary.
```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \ifglshaslong{##1}%
    {\glslongextraLongFmt{##1}}%
    {\glslongextraDescFmt{##1}}%
    &
    \ifglshasshort{##1}%
    {\glslongextraShortTargetFmt{##1}}%
    {\glslongextraNameFmt{##1}}%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifglshaslong{##2}%
    {\glslongextraSubLongFmt{##1}{##2}}%
```

```
                        {\glslongextraSubDescFmt{##1}{##2}}%
                         &
                        \ifglshasshort{##2}%
                        {\glslongextraSubShortTargetFmt{##1}{##2}}%
                        {\glslongextraSubNameFmt{##1}{##2}}%
                        \tabularnewline
                    }%
                    \ifglsnogroupskip
                      \renewcommand*{\glsgroupskip}{}%
                    \else
                      \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                    \fi
                }
```

`\glslongextraCustomIIField`

```
\newcommand{\glslongextraCustomIIField}{userii}
```

`\glslongextraCustomIIHeader`

```
\newcommand{\glslongextraCustomIIHeader}{%
 \MFUsentencecase{\glslongextraCustomIIField}}
```

`\glslongextraCustomIIFmt`

```
\newcommand{\glslongextraCustomIIFmt}[1]{%
  \glsxtrusefield{#1}{\glslongextraCustomIIField}%
}
```

`\glslongextraSubCustomIIFmt`

```
\newcommand{\glslongextraSubCustomIIFmt}[2]{%
 \glslongextraCustomIIFmt{#2}%
}
```

`\glslongextraCustomIIIField`

```
\newcommand{\glslongextraCustomIIIField}{useriii}
```

`\glslongextraCustomIIIHeader`

```
\newcommand{\glslongextraCustomIIIHeader}{%
 \MFUsentencecase{\glslongextraCustomIIIField}}
```

`\glslongextraCustomIIIFmt`

```
\newcommand{\glslongextraCustomIIIFmt}[1]{%
  \glsxtrusefield{#1}{\glslongextraCustomIIIField}%
}
```

`\glslongextraSubCustomIIIFmt`

```
\newcommand{\glslongextraSubCustomIIIFmt}[2]{%
 \glslongextraCustomIIIFmt{#2}%
}
```

`\glslongextraCustomIAlign` Alignment for the custom1 column.

```
\newcommand{\glslongextraCustomIAlign}{l}
```

`\glslongextraCustomIIAlign` Alignment for the custom2 column.

```
\newcommand{\glslongextraCustomIIAlign}{l}
```

`\glslongextraCustomIIIAlign` Alignment for the custom3 column.

```
\newcommand{\glslongextraCustomIIIAlign}{l}
```

`\glslongextraCustomTabularFooter`

```
\newcommand{\glslongextraCustomTabularFooter}{%
 \bottomrule
}
```

752

glslongextraNameCustomIHeader

```
\newcommand{\glslongextraNameCustomIHeader}{%
\glslongextraNameCustomITabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

extraNameCustomITabularHeader

```
\newcommand{\glslongextraNameCustomITabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt{\glslongextraCustomIHeader}%
\tabularnewline\midrule
}
```

lslongextraCustomINameHeader

```
\newcommand{\glslongextraCustomINameHeader}{%
\glslongextraCustomINameTabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

traCustomINameTabularHeader

```
\newcommand{\glslongextraCustomINameTabularHeader}{%
\toprule
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt\entryname
\tabularnewline\midrule
}
```

slongextraNameCustomIIHeader

```
\newcommand{\glslongextraNameCustomIIHeader}{%
\glslongextraNameCustomIITabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

raNameCustomIITabularHeader

```
\newcommand{\glslongextraNameCustomIITabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIHeader}%
\tabularnewline\midrule
}
```

slongextraCustomIINameHeader

```
\newcommand{\glslongextraCustomIINameHeader}{%
\glslongextraCustomIINameTabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

```
\newcommand{\glslongextraCustomIINameTabularHeader}{%
\toprule
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIHeader} &
\glslongextraHeaderFmt\entryname
\tabularnewline\midrule
}
```

```
\newcommand{\glslongextraNameCustomIIIHeader}{%
\glslongextraNameCustomIIITabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

```
\newcommand{\glslongextraNameCustomIIITabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIIHeader}%
\tabularnewline\midrule
}
```

```
\newcommand{\glslongextraCustomIIINameHeader}{%
\glslongextraCustomIIINameTabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

```
\newcommand{\glslongextraCustomIIINameTabularHeader}{%
\toprule
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIIHeader} &
\glslongextraHeaderFmt\entryname
\tabularnewline\midrule
}
```

long-name-custom1 Two column style with custom 1 field in the second column.

```
\newglossarystyle{long-name-custom1}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
    {%
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
```

```
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraCustomIAlign
          }}%
        \@glslongextra@begintab
      }%
      {%
        \glslongextraCustomTabularFooter
        \end{tabular}%
      }%
      \renewcommand*{\glossaryheader}{\glslongextraNameCustomITabularHeader}%
    \else
      \renewenvironment{theglossary}%
      {%
        \glspatchLToutput
        \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraCustomIAlign
          }}%
        \@glslongextra@begintab
      }%
      {\end{longtable}}%
      \renewcommand*{\glossaryheader}{\glslongextraNameCustomIHeader}%
    \fi
    \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{2}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{2}}%
    \renewcommand{\glossentry}[2]{%
      \glslongextraNameFmt{##1} &
      \glslongextraCustomIFmt{##1}\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      \glslongextraSubNameFmt{##1}{##2} &
      \glslongextraSubCustomIFmt{##1}{##2}\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
    \fi
  }
```

long-custom1-name Two column style with custom 1 field in the first column.

```
    \newglossarystyle{long-custom1-name}%
    {%
      \ifGlsLongExtraUseTabular
      \renewenvironment{theglossary}%
      {%
        \edef\@glslongextra@begintab{%
```

```
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraCustomTabularFooter
      \end{tabular}%
    }%
    \renewcommand*{\glossaryheader}{\glslongextraCustomINameTabularHeader}%
  \else
    \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraCustomINameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{2}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{2}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraCustomIFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubCustomIFmt{##1}{##2} &
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
```

long-name-custom2 Three column style with custom 1 field in the second column and custom 2 field
in the third column.

```
\newglossarystyle{long-name-custom2}%
{%
  \ifGlsLongExtraUseTabular
  \renewenvironment{theglossary}%
```

```
    {%
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraCustomTabularFooter
      \end{tabular}%
    }%
  \renewcommand*{\glossaryheader}{\glslongextraNameCustomIITabularHeader}%
  \else
  \renewenvironment{theglossary}%
   {%
      \glspatchLToutput
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
        }}%
      \@glslongextra@begintab
   }%
   {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\glslongextraNameCustomIIHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraCustomIFmt{##1}&
    \glslongextraCustomIIFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
     \glslongextraSubNameFmt{##1}{##2} &
     \glslongextraSubCustomIFmt{##1}{##2} &
     \glslongextraSubCustomIIFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
```

long-custom2-name As long-name-custom2 but with the name column at the end.

```
\newglossarystyle{long-custom2-name}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraCustomTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraCustomIINameTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraCustomIINameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```
Sub-groups are only supported with \printunsrtglossary.
```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraCustomIFmt{##1}&
    \glslongextraCustomIIFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubCustomIFmt{##1}{##2} &
    \glslongextraSubCustomIIFmt{##1}{##2} &
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
   \renewcommand*{\glsgroupskip}{}%
```

758

```
      \else
        \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
      \fi
   }
```

long-name-custom3 Four column style with custom 1 field in the second column, custom 2 field in
the third column and custom 3 field in the fourth column.

```
\newglossarystyle{long-name-custom3}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
          \expandonce\glslongextraCustomIIAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraCustomTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraNameCustomIIITabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
          \expandonce\glslongextraCustomIIIAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraNameCustomIIIHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraCustomIFmt{##1}&
    \glslongextraCustomIIFmt{##1}&
```

759

```
              \glslongextraCustomIIIFmt{##1}\tabularnewline
          }%
          \renewcommand{\subglossentry}[3]{%
              \glslongextraSubNameFmt{##1}{##2} &
              \glslongextraSubCustomIFmt{##1}{##2} &
              \glslongextraSubCustomIIFmt{##1}{##2} &
              \glslongextraSubCustomIIIFmt{##1}{##2}\tabularnewline
          }%
          \ifglsnogroupskip
            \renewcommand*{\glsgroupskip}{}%
          \else
            \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
          \fi
      }
```

long-custom3-name As long-name-custom3 but with the name in the end column.

```
\newglossarystyle{long-custom3-name}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
    {%
        \edef\@glslongextra@begintab{%
          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
              \expandonce\glslongextraCustomIAlign
              \expandonce\glslongextraCustomIIAlign
              \expandonce\glslongextraCustomIIAlign
              \expandonce\glslongextraNameAlign
            }}%
        \@glslongextra@begintab
    }%
    {%
        \glslongextraCustomTabularFooter
        \end{tabular}%
    }%
    \renewcommand*{\glossaryheader}{\glslongextraCustomIIINameTabularHeader}%
  \else
    \renewenvironment{theglossary}%
    {%
        \glspatchLToutput
        \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
              \expandonce\glslongextraCustomIAlign
              \expandonce\glslongextraCustomIIAlign
              \expandonce\glslongextraCustomIIIAlign
              \expandonce\glslongextraNameAlign
            }}%
        \@glslongextra@begintab
    }%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraCustomIIINameHeader}%
```

```
     \fi
     \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
     \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
     \renewcommand{\glossentry}[2]{%
       \glslongextraCustomIFmt{##1}&
       \glslongextraCustomIIFmt{##1}&
       \glslongextraCustomIIIFmt{##1} &
       \glslongextraNameFmt{##1}\tabularnewline
     }%
     \renewcommand{\subglossentry}[3]{%
       \glslongextraSubCustomIFmt{##1}{##2} &
       \glslongextraSubCustomIIFmt{##1}{##2} &
       \glslongextraSubCustomIIIFmt{##1}{##2} &
       \glslongextraSubNameFmt{##1}{##2}\tabularnewline
     }%
     \ifglsnogroupskip
       \renewcommand*{\glsgroupskip}{}%
     \else
       \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
     \fi
   }
```

\glslongextraCustomISetDescWidth Computes the value of \glsdescwidth for the styles that have name, custom1
and description columns.

```
   \newcommand{\glslongextraCustomISetDescWidth}{%
```

Work out the size for just the name and description style.

```
   \glslongextraSetDescWidth
```

Now work out the custom1 column width. This is assuming that the column
title will be the widest text in the column.

```
   \glsmeasurewidth{\gls@tmplen}{\glslongextraHeaderFmt\glslongextraCustomIHeader}%
```

Subtract 2\tabcolsep and the custom1 header width.

```
   \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\gls@tmplen}%
   }
```

\glslongextraCustomIISetDescWidth Computes the value of \glsdescwidth for the styles that have name, custom1,
custom2 and description columns.

```
   \newcommand{\glslongextraCustomIISetDescWidth}{%
   \glslongextraCustomISetDescWidth
   \glsmeasurewidth{\gls@tmplen}{\glslongextraHeaderFmt\glslongextraCustomIIHeader}%
   \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\gls@tmplen}%
   }
```

\glslongextraCustomIIISetDescWidth Computes the value of \glsdescwidth for the styles that have name, custom1,
custom2 and description columns.

```
   \newcommand{\glslongextraCustomIIISetDescWidth}{%
   \glslongextraCustomIISetDescWidth
```

```
                              \glsmeasurewidth{\gls@tmplen}{\glslongextraHeaderFmt\glslongextraCustomIIIHeader}%
                              \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\gls@tmplen}%
                            }
```

gextraNameCustomIDescHeader

```
\newcommand{\glslongextraNameCustomIDescHeader}{%
\glslongextraNameCustomIDescTabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

NameCustomIDescTabularHeader

```
\newcommand{\glslongextraNameCustomIDescTabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt\descriptionname
\tabularnewline\midrule
}
```

gextraNameCustomIIDescHeader

```
\newcommand{\glslongextraNameCustomIIDescHeader}{%
\glslongextraNameCustomIIDescTabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

meCustomIIDescTabularHeader

```
\newcommand{\glslongextraNameCustomIIDescTabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIHeader} &
\glslongextraHeaderFmt\descriptionname
\tabularnewline\midrule
}
```

extraNameCustomIIIDescHeader

```
\newcommand{\glslongextraNameCustomIIIDescHeader}{%
\glslongextraNameCustomIIIDescTabularHeader\endhead
\glslongextraCustomTabularFooter\endfoot
}
```

neCustomIIIDescTabularHeader

```
\newcommand{\glslongextraNameCustomIIIDescTabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt{\glslongextraCustomIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIHeader} &
\glslongextraHeaderFmt{\glslongextraCustomIIIHeader} &
\glslongextraHeaderFmt\descriptionname
```

```
                    \tabularnewline\midrule
                   }
```

long-name-custom1-desc Three column style with custom 1 field in the second column and the description
in the third.

```
\newglossarystyle{long-name-custom1-desc}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraCustomISetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraDescAlign
         }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraCustomTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraNameCustomIDescTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraCustomISetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraDescAlign
         }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraNameCustomIDescHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraCustomIFmt{##1}&
    \glslongextraDescFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
```

```
          \glslongextraSubNameFmt{##1}{##2} &
          \glslongextraSubCustomIFmt{##1}{##2}&
          \glslongextraSubDescFmt{##1}{##2}\tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
      \fi
    }
```

long-name-custom2-desc   Four column style with custom 1 field in the second column, custom 2 field in
                         the third column and the description in the fourth.

```
      \newglossarystyle{long-name-custom2-desc}%
      {%
        \ifGlsLongExtraUseTabular
         \renewenvironment{theglossary}%
          {%
            \glslongextraCustomIISetDescWidth
            \edef\@glslongextra@begintab{%
              \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                \expandonce\glslongextraNameAlign
                \expandonce\glslongextraCustomIAlign
                \expandonce\glslongextraCustomIIAlign
                \expandonce\glslongextraDescAlign
              }}%
            \@glslongextra@begintab
          }%
          {%
            \glslongextraCustomTabularFooter
            \end{tabular}%
          }%
         \renewcommand*{\glossaryheader}{\glslongextraNameCustomIIDescTabularHeader}%
        \else
         \renewenvironment{theglossary}%
          {%
            \glspatchLToutput
            \glslongextraCustomIISetDescWidth
            \edef\@glslongextra@begintab{%
              \noexpand\begin{longtable}{%
                \expandonce\glslongextraNameAlign
                \expandonce\glslongextraCustomIAlign
                \expandonce\glslongextraCustomIIAlign
                \expandonce\glslongextraDescAlign
              }}%
            \@glslongextra@begintab
          }%
          {\end{longtable}}%
         \renewcommand*{\glossaryheader}{\glslongextraNameCustomIIDescHeader}%
        \fi
```

```
            \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```
Sub-groups are only supported with \printunsrtglossary.
```
        \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
        \renewcommand{\glossentry}[2]{%
          \glslongextraNameFmt{##1} &
          \glslongextraCustomIFmt{##1}&
          \glslongextraCustomIIFmt{##1}&
          \glslongextraDescFmt{##1}\tabularnewline
        }%
        \renewcommand{\subglossentry}[3]{%
          \glslongextraSubNameFmt{##1}{##2} &
          \glslongextraSubCustomIFmt{##1}{##2}&
          \glslongextraSubCustomIIFmt{##1}{##2}&
          \glslongextraSubDescFmt{##1}{##2}\tabularnewline
        }%
        \ifglsnogroupskip
          \renewcommand*{\glsgroupskip}{}%
        \else
          \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
        \fi
     }
```

long-name-custom3-desc Five column style with custom 1 field in the second column, custom 2 field in
                        the third column, custom 3 field in the fourth column, and the description in
                        the fifth.
```
  \newglossarystyle{long-name-custom3-desc}%
  {%
    \ifGlsLongExtraUseTabular
      \renewenvironment{theglossary}%
      {%
        \glslongextraCustomIIISetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraCustomIAlign
            \expandonce\glslongextraCustomIIAlign
            \expandonce\glslongextraCustomIIIAlign
            \expandonce\glslongextraDescAlign
          }}%
        \@glslongextra@begintab
      }%
      {%
        \glslongextraCustomTabularFooter
        \end{tabular}%
      }%
      \renewcommand*{\glossaryheader}{\glslongextraNameCustomIIIDescTabularHeader}%
    \else
      \renewenvironment{theglossary}%
      {%
```

765

```
            \glspatchLToutput
            \glslongextraCustomIIISetDescWidth
            \edef\@glslongextra@begintab{%
              \noexpand\begin{longtable}{%
                \expandonce\glslongextraNameAlign
                \expandonce\glslongextraCustomIAlign
                \expandonce\glslongextraCustomIIAlign
                \expandonce\glslongextraCustomIIIAlign
                \expandonce\glslongextraDescAlign
              }}%
            \@glslongextra@begintab
          }%
          {\end{longtable}}%
        \renewcommand*{\glossaryheader}{\glslongextraNameCustomIIIDescHeader}%
        \fi
        \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{5}{##1}}%
```
Sub-groups are only supported with \printunsrtglossary.
```
        \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{5}}%
        \renewcommand{\glossentry}[2]{%
          \glslongextraNameFmt{##1} &
          \glslongextraCustomIFmt{##1}&
          \glslongextraCustomIIFmt{##1}&
          \glslongextraCustomIIIFmt{##1}&
          \glslongextraDescFmt{##1}\tabularnewline
        }%
        \renewcommand{\subglossentry}[3]{%
          \glslongextraSubNameFmt{##1}{##2} &
          \glslongextraSubCustomIFmt{##1}{##2}&
          \glslongextraSubCustomIIFmt{##1}{##2}&
          \glslongextraSubCustomIIIFmt{##1}{##2}&
          \glslongextraSubDescFmt{##1}{##2}\tabularnewline
        }%
        \ifglsnogroupskip
          \renewcommand*{\glsgroupskip}{}%
        \else
          \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
        \fi
      }
```

gextraDescCustomINameHeader

```
      \newcommand{\glslongextraDescCustomINameHeader}{%
       \glslongextraDescCustomINameTabularHeader\endhead
       \glslongextraCustomTabularFooter\endfoot
      }
```

DescCustomINameTabularHeader

```
      \newcommand{\glslongextraDescCustomINameTabularHeader}{%
       \toprule
       \glslongextraHeaderFmt\descriptionname &
```

```
                                  \glslongextraHeaderFmt{\glslongextraCustomIHeader} &
                                  \glslongextraHeaderFmt\entryname
                                  \tabularnewline\midrule
                                  }
```

```
                                  \newcommand{\glslongextraDescCustomIINameHeader}{%
                                  \glslongextraDescCustomIINameTabularHeader\endhead
                                  \glslongextraCustomTabularFooter\endfoot
                                  }
```

```
                                  \newcommand{\glslongextraDescCustomIINameTabularHeader}{%
                                  \toprule
                                  \glslongextraHeaderFmt\descriptionname &
                                  \glslongextraHeaderFmt{\glslongextraCustomIHeader} &
                                  \glslongextraHeaderFmt{\glslongextraCustomIIHeader} &
                                  \glslongextraHeaderFmt\entryname
                                  \tabularnewline\midrule
                                  }
```

```
                                  \newcommand{\glslongextraDescCustomIIINameHeader}{%
                                  \glslongextraDescCustomIIINameTabularHeader\endhead
                                  \glslongextraCustomTabularFooter\endfoot
                                  }
```

```
                                  \newcommand{\glslongextraDescCustomIIINameTabularHeader}{%
                                  \toprule
                                  \glslongextraHeaderFmt\descriptionname &
                                  \glslongextraHeaderFmt{\glslongextraCustomIHeader} &
                                  \glslongextraHeaderFmt{\glslongextraCustomIIHeader} &
                                  \glslongextraHeaderFmt{\glslongextraCustomIIIHeader} &
                                  \glslongextraHeaderFmt\entryname
                                  \tabularnewline\midrule
                                  }
```

long-desc-custom1-name As long-name-custom1-desc but with the name and description columns the
other way around.

```
                                  \newglossarystyle{long-desc-custom1-name}%
                                  {%
                                    \ifGlsLongExtraUseTabular
                                      \renewenvironment{theglossary}%
                                      {%
                                        \glslongextraCustomISetDescWidth
                                        \edef\@glslongextra@begintab{%
                                          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                                            \expandonce\glslongextraDescAlign
```

```
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraCustomTabularFooter
      \end{tabular}%
    }%
  \renewcommand*{\glossaryheader}{\glslongextraDescCustomINameTabularHeader}%
  \else
  \renewenvironment{theglossary}%
  {%
    \glspatchLToutput
    \glslongextraCustomISetDescWidth
    \edef\@glslongextra@begintab{%
      \noexpand\begin{longtable}{%
        \expandonce\glslongextraDescAlign
        \expandonce\glslongextraCustomIAlign
        \expandonce\glslongextraNameAlign
      }}%
    \@glslongextra@begintab
  }%
  {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\glslongextraDescCustomINameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{3}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraDescFmt{##1} &
    \glslongextraCustomIFmt{##1}&
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubCustomIFmt{##1}{##2}&
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
```

long-desc-custom2-name As long-name-custom2-desc but with the name and description columns the other way around.

```
  \newglossarystyle{long-desc-custom2-name}%
```

```
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
    {%
      \glslongextraCustomIISetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraCustomTabularFooter
      \end{tabular}%
    }%
    \renewcommand*{\glossaryheader}{\glslongextraDescCustomIINameTabularHeader}%
  \else
    \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraCustomIISetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraCustomIAlign
          \expandonce\glslongextraCustomIIAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraDescCustomIINameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
```

Sub-groups are only supported with \printunsrtglossary.

```
  \renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{4}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraDescFmt{##1} &
    \glslongextraCustomIFmt{##1}&
    \glslongextraCustomIIFmt{##1}&
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubCustomIFmt{##1}{##2}&
    \glslongextraSubCustomIIFmt{##1}{##2}&
```

```
                      \glslongextraSubNameFmt{##1}{##2}\tabularnewline
                    }%
                    \ifglsnogroupskip
                      \renewcommand*{\glsgroupskip}{}%
                    \else
                      \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
                    \fi
                  }
```

long-desc-custom3-name As long-name-custom-desc but with the name and description columns switched.

```
                  \newglossarystyle{long-desc-custom3-name}%
                  {%
                    \ifGlsLongExtraUseTabular
                     \renewenvironment{theglossary}%
                     {%
                        \glslongextraCustomIIISetDescWidth
                        \edef\@glslongextra@begintab{%
                          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
                            \expandonce\glslongextraDescAlign
                            \expandonce\glslongextraCustomIAlign
                            \expandonce\glslongextraCustomIIAlign
                            \expandonce\glslongextraCustomIIIAlign
                            \expandonce\glslongextraNameAlign
                          }}%
                        \@glslongextra@begintab
                     }%
                     {%
                        \glslongextraCustomTabularFooter
                        \end{tabular}%
                     }%
                     \renewcommand*{\glossaryheader}{\glslongextraDescCustomIIINameTabularHeader}%
                    \else
                     \renewenvironment{theglossary}%
                     {%
                        \glspatchLToutput
                        \glslongextraCustomIIISetDescWidth
                        \edef\@glslongextra@begintab{%
                          \noexpand\begin{longtable}{%
                            \expandonce\glslongextraDescAlign
                            \expandonce\glslongextraCustomIAlign
                            \expandonce\glslongextraCustomIIAlign
                            \expandonce\glslongextraCustomIIIAlign
                            \expandonce\glslongextraNameAlign
                          }}%
                        \@glslongextra@begintab
                     }%
                     {\end{longtable}}%
                     \renewcommand*{\glossaryheader}{\glslongextraDescCustomIIINameHeader}%
                    \fi
                    \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{5}{##1}}%
```

Sub-groups are only supported with `\printunsrtglossary`.

```
\renewcommand*{\glssubgroupheading}{\glslongextraSubGroupHeading{5}}%
\renewcommand{\glossentry}[2]{%
  \glslongextraDescFmt{##1} &
  \glslongextraCustomIFmt{##1}&
  \glslongextraCustomIIFmt{##1}&
  \glslongextraCustomIIIFmt{##1}&
  \glslongextraNameFmt{##1}\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
  \glslongextraSubDescFmt{##1}{##2} &
  \glslongextraSubCustomIFmt{##1}{##2}&
  \glslongextraSubCustomIIFmt{##1}{##2}&
  \glslongextraSubCustomIIIFmt{##1}{##2}&
  \glslongextraSubNameFmt{##1}{##2}\tabularnewline
}%
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

7   topic styles (**glossary-topic.sty**)

Provides "topic" styles where top-level entries are considered a topic.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v1.48}{2021-11-22}{glossary-topic-2021-11-22.sty}
\DeclareCurrentRelease{v1.7}{2025-05-14}
```

Declare package:

```
\ProvidesPackage{glossary-topic}[2025/05/14 v1.7 (NLCT)]
```

Load required package.

```
\RequirePackage{multicol}
```

The top-level entries act like headers. If the top-level entry has a description it's placed below the name.

topic

```
\newglossarystyle{topic}{%
  \renewenvironment{theglossary}%
  {%
    \glstopicInit
    \def\glstopic@prechildren{}%
    \def\glstopic@prevlevel{-1}%
  }%
  {\par}%
```

```
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{%
      \def\glstopic@prevlevel{-1}%
      \glstopicGroupHeading{##1}%
    }%
```

Sub-groups are only supported with \printunsrtglossary.

```
    \renewcommand*{\glssubgroupheading}{\glstopicSubGroupHeading}%
    \renewcommand{\glossentry}[2]{%
      \hangindent0pt\relax
      \parindent\glstopicParIndent\relax
      \glstopicItem{##1}{##2}%
```

If there isn't a description, penalise a page break.

```
      \ifglshasdesc{##1}%
      {%
        \def\glstopic@prechildren{}%
      }%
      {%
        \def\glstopic@prechildren{\nopagebreak}%
      }%
    }%
    \renewcommand{\subglossentry}[3]{%
      \ifnum\glstopic@prevlevel=0\relax\glstopic@prechildren\fi
      \def\glstopic@prevlevel{##1}%
```

Grouping is added to scope the effect of \everypar.

```
      \begingroup
      \glstopicAssignSubIndent{##1}%
      \glstopicSubItem{##1}{##2}{##3}%

      \par
      \endgroup
    }%
    \renewcommand*{\glsgroupskip}{}%
  }
```

\glstopicGroupHeading

> \glstopicGroupHeading{⟨*group label*⟩}

May be redefined if letter group headings are required. For example:

```
%\renewcommand*{\glstopicGroupHeading}[1]{%
%  \glsxtrgetgrouptitle{#1}{\thisgrptitle}%
%  \section*{\thisgrptitle}%
%}
%

  \newcommand*{\glstopicGroupHeading}[1]{}
```

\glstopicSubGroupHeading

> \glstopicSubGroupHeading{⟨*prev group level*⟩}{⟨*group level*⟩}{⟨*parent entry*⟩}{⟨*group label*⟩}

```
\newcommand*{\glstopicSubGroupHeading}[4]{%
 \begingroup
  \glspar\glstopicPreSkip\glspar\noindent
  \glsxtrgetgrouptitle{#4}{\glstopicsubgrouptitle}%
  \glstopicAssignSubIndent{#2}%
  \glstopicSubItemBox{#2}{\glstopicTitleFont{\glstopicsubgrouptitle}}%
  \glstopicSubItemSep
  \glspar\nobreak\glstopicPostSkip
  \par
 \endgroup
}
```

\glstopicItem

> \glstopicItem{⟨*label*⟩}{⟨*location list*⟩}

```
\newcommand*{\glstopicItem}[2]{%
 \glspar\glstopicPreSkip\glspar\noindent
 \glstopicMarker{#1}%
 \glstopicTitleFont
 {%
   \glsentryitem{#1}\glstarget{#1}{\glstopicTitle{#1}}%
 }%
 \ifglshasdesc{#1}%
 {\glspar\nobreak\glstopicMidSkip\glspar\nobreak
  \@afterheading\glstopicDesc{#1}\glspar\glstopicPostSkip
 }%
 {\glspar\nobreak\glstopicPostSkip}%
 \glstopicLoc{#1}{#2}%
}
```

\glstopicMarker May be used to insert a bookmark etc if required.

```
\newcommand*{\glstopicMarker}[1]{}
```

\glstopicName

```
\newcommand*{\glstopicTitle}[1]{\Glossentryname{#1}%
 \ifglshassymbol{#1}{\space(\glossentrysymbol{#1})}{}%
}
```

\glstopicTitleFont

```
\newcommand*{\glstopicTitleFont}[1]{\textbf{\large #1}}
```

\glstopicDesc

```
\newcommand*{\glstopicDesc}[1]{\Glossentrydesc{#1}\glspostdescription}
```

`\glstopicLoc`

```
\newcommand*{\glstopicLoc}[2]{}
```

`\glstopicParIndent`

```
\newlength\glstopicParIndent
\setlength\glstopicParIndent{20pt}
```

`\glstopicSubIndent`

```
\newlength\glstopicSubIndent
\setlength\glstopicSubIndent{20pt}
```

`\glstopicInit`

```
\newcommand{\glstopicInit}{}
```

`\glstopicAssignSubIndent`

> `\glstopicAssignSubIndent{⟨level⟩}`

Used to set the indentation for sub-levels.

```
\newcommand*{\glstopicAssignSubIndent}[1]{%

  \par
  \parindent\dimexpr#1\glstopicSubIndent-\glstopicSubIndent\relax
  \glstopicAssignWidest{#1}%
  \glstopicsubitemhangindent=\dimexpr\parindent+\glstopicwidest\relax
  \hangindent\glstopicsubitemhangindent\relax

  \everypar{\hangindent\glstopicsubitemhangindent\relax
    \parindent\dimexpr\glstopicSubItemParIndent+\glstopicsubitemhangindent\relax}%
}
```

`\glstopicsubitemhangindent`

```
\newlength\glstopicsubitemhangindent
```

`\glstopicSubItemParIndent`

```
\newlength\glstopicSubItemParIndent
\glstopicSubItemParIndent\parindent
```

`\glstopicwidest`

```
\newlength\glstopicwidest
```

`\glstopicAssignWidest`

> `\glstopicAssignWidest{⟨level⟩}`

Used in the definition of `\glstopicAssignSubIndent` to set the indentation from the widest name for the given level. This will require glossary-tree to set the values.

```
\newcommand*{\glstopicAssignWidest}[1]{%
```

```
                  \ifcsundef{@glswidestlength\romannumeral#1}%
                  {%
                    \ifcsdef{@glswidestname\romannumeral#1}%
                    {%
                      \glsmeasurewidth{\glstopicwidest}{%
                       \glstopicSubNameFont{\csuse{@glswidestname\romannumeral#1}}%
                       \glstopicSubItemSep
                      }%
                    }%
                    {\setlength{\glstopicwidest}{0pt}}%
```
Save the value so that it doesn't have to keep being recalculated.
```
                    \csedef{@glswidestlength\romannumeral#1}{\the\glstopicwidest}%
                  }%
                  {\setlength{\glstopicwidest}{\csuse{@glswidestlength\romannumeral#1}}}%
                }
```

\glstopicPreSkip
```
                  \newcommand*{\glstopicPreSkip}{\medskip}
```

\glstopicMidSkip
```
                  \newcommand*{\glstopicMidSkip}{\smallskip}
```

\glstopicPostSkip
```
                  \newcommand*{\glstopicPostSkip}{\smallskip}
```

$\boxed{\texttt{\textbackslash glstopicSubItem}\{\langle level \rangle\}\{\langle label \rangle\}\{\langle location\ list \rangle\}}$

\glstopicSubItem
```
                  \newcommand*{\glstopicSubItem}[3]{%

                    \glstopicSubItemBox{#1}{\glstopicSubNameFont{\glsentryitem{#2}%
                      \glstarget{#2}{\glossentryname{#2}}}%
                      \glstopicSubItemSep
                    }%
                    \ifglshassymbol{#2}{(\glossentrysymbol{#2})\space}{}%

                    \ifglshasdesc{#2}%
                     {\glossentrydesc{#2}\glspostdescription\glstopicSubPreLocSep}{}%
                    \glstopicSubLoc{#2}{#3}%
                  }
```

\glstopicSubItemSep
```
                  \newcommand*{\glstopicSubItemSep}{\quad}
```

$\boxed{\texttt{\textbackslash glstopicSubItemBox}\{\langle level \rangle\}\{\langle text \rangle\}}$

\glstopicSubItemBox
```
                  \newcommand*{\glstopicSubItemBox}[2]{%
                    \ifdim\glstopicwidest>0pt\relax\makebox[\glstopicwidest][l]{#2}\else#2\fi
                  }
```

```

\glstopicSubNameFont

         \newcommand*{\glstopicSubNameFont}[1]{\textbf{#1}}

\glstopicSubPreLocSep

         \newcommand*{\glstopicSubPreLocSep}{\space}

\glstopicSubLoc

         \newcommand*{\glstopicSubLoc}[2]{#2}

\glstopicCols

         \newcommand*{\glstopicCols}{2}

\glstopicColsEnv

         \newcommand*{\glstopicColsEnv}{multicols}

topicmcols

         \newglossarystyle{topicmcols}{%
           \renewenvironment{theglossary}%
           {%
             \glstopicInit
             \def\glstopic@prechildren{}%
             \def\glstopic@postchildren{}%
             \def\glstopic@prevlevel{-1}%
           }%
           {%
             \ifnum\glstopic@prevlevel>0\relax\glstopic@postchildren\fi
             \par
           }%
           \renewcommand*{\glossaryheader}{}%
           \renewcommand*{\glsgroupheading}[1]{%
             \ifnum\glstopic@prevlevel>0\relax\glstopic@postchildren\fi
             \def\glstopic@prevlevel{-1}%
             \glstopicGroupHeading{##1}%
           }%

Sub-groups are only supported with \printunsrtglossary.

           \renewcommand*{\glssubgroupheading}{\glstopicSubGroupHeading}%
           \renewcommand{\glossentry}[2]{%
             \ifnum\glstopic@prevlevel>0\relax\glstopic@postchildren\fi
             \def\glstopic@prevlevel{0}%
             \hangindent0pt\relax
             \parindent\glstopicParIndent\relax
             \glstopicItem{##1}{##2}%
             \ifnum\glstopicCols>1\relax

If there isn't a description, penalise a page break.

             \ifglshasdesc{##1}%
             {%
               \edef\glstopic@prechildren{%
                 \noexpand\begin{\glstopicColsEnv}{\glstopicCols}%

776

```
      }%
    }%
    {%
      \edef\glstopic@prechildren{%
        \noexpand\nopagebreak
        \noexpand\begin{\glstopicColsEnv}{\glstopicCols}%
      }%
    }%
    \edef\glstopic@postchildren{\noexpand\end{\glstopicColsEnv}}%
  \fi
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum\glstopic@prevlevel=0\relax\glstopic@prechildren\fi
  \def\glstopic@prevlevel{##1}%
  \glstopicAssignSubIndent{##1}%
  \glstopicSubItem{##1}{##2}{##3}%
}%
\renewcommand*{\glsgroupskip}{}%
}
```

8   table styles (**glossary-table.sty**)

Intended for use with `bib2gls`. This is still experimental.

```
\NeedsTeXFormat{LaTeX2e}
```

Declare package:

```
\ProvidesPackage{glossary-table}[2025/05/14 v1.7 (NLCT)]
```

Load required packages.

```
\RequirePackage{longtable}
\RequirePackage{array}
\RequirePackage{booktabs}
```

Check if `\gls@start@measuring` has been defined (introduced to glossaries v4.51). This package also requires `\ifglsfieldvoid` which was added to glossaries v4.50.

```
\ifdef\gls@start@measuring
{}
{\PackageError{glossary-table}%
 {glossaries.sty v4.51+ required. Please update glossaries.sty}
 {Your version of glossaries.sty is too old. Minimum version 4.51 required}
}
```

\glstableblockperrowcount Number of blocks (entries) per row.

```
\newcount\glstableblockperrowcount
\glstableblockperrowcount=2\relax
```

Add a key to allow this value to be changed.

```
\define@key{printglosstable}{blocks}{\glstableblockperrowcount=#1\relax}
```

\glstablecurrentblockindex Keep track of current block (entry) index.

```
\newcount\glstablecurrentblockindex
```

\glstabletotalcols Total number of columns. This will be updated at the start of \printunsrttable, but is a user level command so that it can be used in any hooks.

```
\newcount\glstabletotalcols
\glstabletotalcols=4\relax
```

\glstablenameheader

```
\newcommand{\glstablenameheader}{\entryname}
```

\glstabledescheader

```
\newcommand{\glstabledescheader}{\descriptionname}
```

\glstableotherheader

```
\newcommand{\glstableotherheader}{\MFUsentencecase{\glstableotherfield}}
```

\glstablesymbolheader

```
\newcommand{\glstablesymbolheader}{\symbolname}
```

Provide boolean option to suppress header.

```
\define@boolkey{printglosstable}{header}[true]{}
\KV@printglosstable@headertrue
```

Provide boolean option to suppress rules.

```
\define@boolkey{printglosstable}{rules}[true]{}
\KV@printglosstable@rulestrue
```

Provide boolean option to suppress caption.

```
\define@boolkey{printglosstable}{caption}[true]{}
\KV@printglosstable@captiontrue
```

```
\define@key{printglosstable}{blocksep}{\renewcommand{\glstable@blockalignsep}{#1}}
```

\glstable@blockalignsep Alignment spec between blocks.

```
\newcommand{\glstable@blockalignsep}{|}
```

\glstablesubentryalign

```
\newcommand{\glstablesubentryalign}{%
 \glstableleftalign{\dimexpr\glstablesubentrywidth-\tabcolsep}@{}}
```

\glstablesubentrywidth

```
\newcommand{\glstablesubentrywidth}{\glstabledescwidth}
```

glstablesubentries (*env.*)

```
\newenvironment{glstablesubentries}%
 {%
   \protected@edef\@gls@dotabular{%
     \noexpand\begin{tabular}[t]{\glstablesubentryalign}}%
   \@gls@dotabular
 }%
 {\end{tabular}}
```

`\glstablePreChildren`

```
\newcommand{\glstablePreChildren}{\glstableifpar{\par}}
```

`\glstableblocksubentrysep`

```
\newcommand{\glstableblocksubentrysep}{\glstablenewline}
```

Provide boolean option to allow paragraph cells.

```
\define@choicekey{printglosstable}{par}
[\@glstable@par@val\@glstable@par@n]
{false,justified,ragged}
{%
  \ifcase\@glstable@par@n
    \renewcommand{\glstable@parcase}[3]{##1}%
  \or
    \renewcommand{\glstable@parcase}[3]{##2}%
  \or
    \renewcommand{\glstable@parcase}[3]{##3}%
  \fi
}
```

`\glstable@parcase`

```
\newcommand{\glstable@parcase}[3]{#1}
```

`\glstableifpar`

```
\newcommand{\glstableifpar}[1]{\glstable@parcase{}{#1}{#1}}
```

`\glstableleftalign`

```
\newcommand{\glstableleftalign}[1]{%
 \glstable@parcase{l}{p{#1}}{>{\protect\raggedright}p{#1}}%
}
```

`\glstablerightalign`

```
\newcommand{\glstablerightalign}[1]{%
 \glstable@parcase{r}{p{#1}}{>{\protect\raggedleft}p{#1}}%
}
```

`\glstablecenteralign`

```
\newcommand{\glstablecenteralign}[1]{%
 \glstable@parcase{c}{p{#1}}{>{\protect\centering}p{#1}}%
}
```

`\glstablenamecolalign` The alignment for the name column.

```
\newcommand{\glstablenamecolalign}{\glstableleftalign{\glstablenamewidth}}
```

`\glstabledesccolalign` The alignment for the description column.

```
\newcommand{\glstabledesccolalign}{\glstableleftalign{\glstabledescwidth}}
```

`\glstableothercolalign` The alignment for the description column.

```
\newcommand{\glstableothercolalign}{\glstableleftalign{\glstableotherwidth}}
```

779

`\glstablesymbolcolalign` The alignment for the symbol column.

```
\newcommand{\glstablesymbolcolalign}{\glstablecenteralign{\glstablesymbolwidth}}
```

`\glstableNameTarget`

```
\newcommand{\glstableNameTarget}[1]{%
 \glstarget{#1}{\glstableName{#1}}%
}
```

`\glstableNameFmt`

```
\newcommand{\glstableNameFmt}[1]{#1}
```

`\glstableName` Entry item needs to be included in measuring to ensure there's enough room for it as well.

```
\newcommand{\glstableName}[1]{%
 \glsentryitem{#1}%
 \glstableNameFmt{\glossentryname{#1}}}
```

`\glstableSubNameTarget`

```
\newcommand{\glstableSubNameTarget}[1]{%
 \glstarget{#1}{\glstableSubName{#1}}%
}
```

`\glstableSubNameFmt`

```
\newcommand{\glstableSubNameFmt}[1]{}
```

`\glstableSubName`

```
\newcommand{\glstableSubName}[1]{%
 \glssubentryitem{#1}%
 \glstableSubNameFmt{\glossentryname{#1}}%
}
```

`\glstableotherfield`

```
\newcommand{\glstableotherfield}{}
```

`\glstableifhasotherfield`

```
\newcommand{\glstableifhasotherfield}[3]{%
  \ifdefvoid\glstableotherfield
  {#3}%
  {%
    \ifglsfieldvoid{\glstableotherfield}{#1}{#3}{#2}%
  }%
}
```

Add an extra key to allow this value to be changed.

```
\define@key{printglosstable}{other}{\renewcommand{\glstableotherfield}{#1}}
```

`\glstableOther`

```
\newcommand{\glstableOther}[1]{%
 \glstableOtherFmt{\glsxtrusefield{#1}{\glstableotherfield}}}
```

\glstableOtherFmt

```
\newcommand{\glstableOtherFmt}[1]{#1}
```

\glstableSubOther

```
\newcommand{\glstableSubOther}[1]{\glstableOther{#1}}
```

\glstableOtherWithSep

```
\newcommand{\glstableOtherWithSep}[3]{%
 \glstableifhasotherfield{#2}%
 {#1\glstableOther{#2}#3}%
 {}%
}
```

\glstableSubOtherWithSep

```
\newcommand{\glstableSubOtherWithSep}[3]{%
 \glstableifhasotherfield{#2}%
 {#1\glstableSubOther{#2}#3}%
 {}%
}
```

\glstableNameSingleFmt

```
\newcommand{\glstableNameSingleFmt}[1]{%
 \glstableNameTarget{#1}%
 \ifglshasdesc{#1}%
 {%
```

Has description.

```
   \glstableNameSinglePostName
   \glstableNameSingleSuppl
   {%
     \ifglshassymbol{#1}%
     {\glstableSymbol{#1}\glstableNameSingleSymSep}%
     {}%

     \glstableOtherWithSep{}{#1}{\glstableOtherSep}%
     \glstableDesc{#1}%
   }%
 }%
 {%
```

No description.

```
   \ifglshassymbol{#1}%
   {%
```

Has symbol

```
     \glstableNameSinglePostName
     \glstableNameSingleSuppl
     {%
       \glstableSymbol{#1}%
```

781

```
                    \glstableifhasotherfield{#1}%
                    {%

                      \glstableNameSingleSymSep\glstableOther{#1}%
                    }%
                    {}%
                  }%
                }%
                {%

No description or symbol.
                  \glstableifhasotherfield{#1}%
                  {%

Has other but no description or symbol
                    \glstableNameSinglePostName
                    \glstableNameSingleSuppl{\glstableOther{#1}}%
                  }%
                  {%

No description, symbol or other.
                  }%
                }%
              }%
            }
```

\glstableNameSingleSuppl

```
\newcommand{\glstableNameSingleSuppl}[1]{(#1)}
```

\glstableNameSinglePostName

```
\newcommand{\glstableNameSinglePostName}{ }
```

\glstableNameSingleSymSep

```
\newcommand{\glstableNameSingleSymSep}{ }
```

\glstableOtherSep

```
\newcommand{\glstableOtherSep}{, }
```

\glstableSubOtherSep

```
\newcommand{\glstableSubOtherSep}{\glstableOtherSep}
```

\glstableSubDescSep

```
\newcommand{\glstableSubDescSep}{\glstableSubOtherSep}
```

\glstableSubNameSingleFmt

```
\newcommand{\glstableSubNameSingleFmt}[1]{%
\glstableSubNameTarget{#1}%

\ifglshasdesc{#1}%
{%
```

```
    \ifglshassymbol{#1}%
    {%
      \glstableifhasotherfield{#1}%
      {%
```

Description, symbol and other

```
        \glstableNameSinglePostSubName
        \glstableNameSingleSubSuppl
        {%
          \glstableSubSymbol{#1}%
          \glstableNameSingleSymSep
          \glstableSubOtherWithSep{}{#1}{\glstableSubOtherSep}%
          \glstableSubDesc{#1}%
        }%
      }%
      {%
```

Description and symbol but no other.

```
        \glstableNameSinglePostSubName
        \glstableNameSingleSubSuppl
        {%
          \glstableSubSymbol{#1}%
          \glstableNameSingleSymSep
          \glstableSubDesc{#1}%
        }%
     }%
    }%
    {%
```

Description but no symbol.

```
      \glstableNameSinglePostSubName
      \glstableNameSingleSubSuppl
      {%
        \glstableSubOtherWithSep{}{#1}{\glstableSubOtherSep}%
        \glstableSubDesc{#1}%
      }%
    }%
  }%
  {%
```

No description.

```
    \ifglshassymbol{#1}%
    {%
```

No description but has symbol.

```
      \glstableNameSinglePostSubName
      \glstableNameSingleSubSuppl
      {%
        \glstableifhasotherfield{#1}%
        {%
```

No description, but has symbol and other.

```
                    \glstableSubSymbol{#1}\glstableNameSingleSymSep
                    \glstableSubOther{#1}%
                }%
                {%
```
No description or other but has symbol.
```
                    \glstableSubSymbol{#1}%
                }%
              }%
            }%
            {%
              \glstableifhasotherfield{#1}%
              {%
```
No description or symbol but has other.
```
                \glstableNameSinglePostSubName
                \glstableNameSingleSubSuppl{\glstableSubOther{#1}}%
              }%
              {%
```
No description, symbol or other.
```
              }%
            }%
         }%
        }
```

```
                    \newcommand{\glstableNameSingleSubSuppl}[1]{#1}
```

```
                    \newcommand{\glstableNameSinglePostSubName}{ }
```

```
                    \newcommand{\glstableSubSep}{\space}
```

```
                    \newcommand{\glstableSubNameSep}{}
```

```
                    \newcommand{\glstableNameNoDesc}[1]{%
                      \glstableNameTarget{#1}%
                      \glstableOtherWithSep{\glstableSubNameSep}{##1}{}%
                    }
```

```
                    \newcommand{\glstableSubNameNoDesc}[1]{%
                      \glstableSubNameTarget{#1}%
                      \glstableSubOtherWithSep{\glstableSubNameSep}{#1}{}%
                    }
```

```
\newcommand{\glstableSubNameSymbolNoDesc}[1]{%
  \glstableSubNameTarget{#1}%
  \glstableifhasotherfield{#1}%
  {%
    \glstableSubOther{#1}%
    \ifglshassymbol{#1}%
    {\glstableSubOtherSep\glstableSubSymbol{#1}}%
    {}%
  }%
  {%
    \ifglshassymbol{#1}%
    {\glstableSubSymbol{#1}}%
    {}%
  }%
}
```

\glstableSymbolFmt

```
\newcommand{\glstableSymbolFmt}[1]{#1}
```

\glstableSymbol

```
\newcommand{\glstableSymbol}[1]{\glstableSymbolFmt{\glossentrysymbol{#1}}}
```

\glstableSubSymbolFmt

```
\newcommand{\glstableSubSymbolFmt}[1]{\glstableSymbolFmt{#1}}
```

\glstableSubSymbol

```
\newcommand{\glstableSubSymbol}[1]{\glstableSubSymbolFmt{\glossentrysymbol{#1}}}
```

\glstableSubSymbolWithSep

```
\newcommand{\glstableSubSymbolWithSep}[3]{%
 \ifglshassymbol{#2}%
 {#1\glstableSubSymbol{#2}#3}%
 {}%
}
```

\glstableSymbolNameTarget Where the symbol takes place of the name.

```
\newcommand{\glstableSymbolNameTarget}[1]{%
 \glstarget{#1}{\glstableSymbolName{#1}}%
}
```

\glstableSymbolNameFmt

```
\newcommand{\glstableSymbolNameFmt}[1]{%
 \glstableSymbolFmt{#1}%
}
```

\glstableSymbolName

```
\newcommand{\glstableSymbolName}[1]{%
 \glsentryitem{#1}\glstableSymbolNameFmt{\glossentrysymbol{#1}}%
}
```

Where the symbol takes place of the name.

```
\newcommand{\glstableSubSymbolNameTarget}[1]{%
\glstarget{#1}{\glstableSubSymbolName{#1}}%
}
```

\glstableSubSymbolNameFmt

```
\newcommand{\glstableSubSymbolNameFmt}[1]{}
```

\glstableSubSymbolName

```
\newcommand{\glstableSubSymbolName}[1]{%
\glssubentryitem{#1}\glstableSubSymbolNameFmt{\glossentrysymbol{#1}}%
}
```

\glstableDesc

```
\newcommand{\glstableDesc}[1]{%
\glstableDescFmt{\glossentrydesc{#1}\glspostdescription}%
}
```

\glstableDescFmt

```
\newcommand{\glstableDescFmt}[1]{#1}
```

\glstableDescWithOther

```
\newcommand{\glstableDescWithOther}[1]{%
  \glstableifhasotherfield{#1}%
  {%
    \glstableOther{#1}%
    \ifglshasdesc{#1}{\glstableOtherSep\glstableDesc{#1}}{}%
  }%
  {%
    \ifglshasdesc{#1}{\glstableDesc{#1}}{}%
  }%
}
```

\glstableSubDescFmt

```
\newcommand{\glstableSubDescFmt}[1]{\glstableDescFmt{#1}}
```

\glstableSubDesc

```
\newcommand{\glstableSubDesc}[1]{%
\glstableSubDescFmt{\glossentrydesc{#1}\glspostdescription}%
}
```

\glstableSubDescWithOther

```
\newcommand{\glstableSubDescWithOther}[1]{\glstableDescWithOther{#1}}
```

\glstableSubDescSymbolOther

```
\newcommand{\glstableSubDescSymbolOther}[1]{%
\ifglshasdesc{#1}%
{%
```

```
        \glstableSubDesc{#1}%
        \ifglshassymbol{#1}%
        {%
          \glstableSubDescSep
          \glstableSubSymbol{#1}%
          \glstableSubOtherWithSep{\glstableSubSep}{#1}{}%
        }%
        {%
          \glstableSubOtherWithSep{\glstableSubOtherSep}{#1}{}%
        }%
      }%
      {%
        \ifglshassymbol{#1}%
        {%
          \glstableSubSymbol{#1}%
          \glstableSubOtherWithSep{\glstableSubSep}{#1}{}%
        }%
        {\glstableSubOther{#1}}%
      }%
    }
```

\glstableOtherNoDesc

```
\newcommand{\glstableOtherNoDesc}[1]{%
 \glstableOtherIfSet{#1}%
}
```

\glstableOtherIfSet

```
\newcommand{\glstableOtherIfSet}[1]{%
  \glstableifhasotherfield{#1}{\glstableOther{#1}}{}%
}
```

\glstableSubOtherNoDesc

```
\newcommand{\glstableSubOtherNoDesc}[1]{%
 \glstableOtherNoDesc{#1}%
}
```

\glstableSubOtherIfSet

```
\newcommand{\glstableSubOtherIfSet}[1]{%
 \glstableOtherIfSet{#1}%
}
```

\glstableHeaderFmt

```
\newcommand{\glstableHeaderFmt}[1]{\textbf{#1}}

\define@key{printglosstable}{block-style}
{\glstablesetstyle{#1}}
```

\glstablecolsperblock   Number of columns per block (entry). Assigned by block style.

```
\newcount\glstablecolsperblock
\glstablecolsperblock=2\relax
```

787

`\glstableblockheader` The column header, which may cover multiple columns. Redefined by block style.

```
\newcommand{\glstableblockheader}{}
```

`\glstableblockalign` The column alignment specs for the block. Redefined by the block style.

```
\newcommand{\glstableblockalign}{}
```

`\glstableblockentry` The entry item, which may cover multiple columns. Redefined by block style.

```
\newcommand{\glstableblockentry}[1]{}
```

`\glstableblocksubentry` The sub-entry is in a single column of the block (requires children to be saved) Redefined by block style.

```
\newcommand{\glstableblocksubentry}[1]{}
```

`\glstableinitlengthupdates` Block style command.

```
\newcommand{\glstableinitlengthupdates}{}
```

`\glstablelengthupdate` Block style command.

```
\newcommand{\glstablelengthupdate}[1]{}
```

`\glstablefinishlengthupdates` Block style command.

```
\newcommand{\glstablefinishlengthupdates}{}
```

`\glstablesetstyle`

```
\newcommand{\glstablesetstyle}[1]{%
  \ifcsdef{@glstable@style@#1}%
  {\csuse{@glstable@style@#1}}%
  {\PackageError{glossary-table}{Unknown style '#1'}{}}%
}
```

`\glstablenewstyle`

```
\newcommand{\glstablenewstyle}[2]{%
  \ifcsdef{@glstable@style@#1}%
  {\PackageError{glossary-table}{style '#1' already defined}{}}%
  {\csdef{@glstable@style@#1}{#2}}%
}
```

Provide some common layouts.

`name-desc`

```
\glstablenewstyle{name-desc}{%
```
2 columns per block (name, description).
```
\glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max name width if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablenamewidth}%
      {\glstableHeaderFmt\glstablenameheader}%
  \else
    \setlength{\glstablenamewidth}{0pt}%
  \fi
  \setlength{\glstabledescwidth}{0pt}%
}%
```

Update width in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
}%
```

Finally set the description width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
   - \glstablenamewidth}%
  \ifdim\glstabledescwidth<0pt\relax
    \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstabledescwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameTarget{##1} &
  \glstableDescWithOther{##1}%
  \glstableChildEntries{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameTarget{##1}\glstableSubNameSep
  \glstableSubDescWithOther{##1}%
}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstabledescwidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablenameheader &
  \glstableHeaderFmt\glstabledescheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstablenamecolalign\glstabledesccolalign}%
}
```

Set the default style.

```
\glstablesetstyle{name-desc}
```

```
\glstablenewstyle{name}{%
```

1 columns per block (name optionally with symbol and description).

```
\glstablecolsperblock=1\relax
```

Initialise length registers (no calculation required, column width same as block width).

```
\renewcommand{\glstableinitlengthupdates}{}%
```

No measuring required.

```
\renewcommand{\glstablelengthupdate}[1]{}%
```

Set the name width to the amount available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstablenamewidth}{\glstableblockwidth}%
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameSingleFmt{##1}%
  \glstableChildEntries{##1}%
 }%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameSingleFmt{##1}}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstableblockwidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablenameheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstablenamecolalign}%
 }
```

```
\glstablenewstyle{name-symbol}{%
```

2 columns per block (name and symbol).

```
\glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max symbol width if par align). This assumes the symbol requires the minimal width and any leftover can be assigned to the name.

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablesymbolwidth}%
      {\glstableHeaderFmt\glstablesymbolheader}%
  \else
```

790

```
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
  \setlength{\glstablenamewidth}{0pt}%
}%
```

Update widths in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbol{##1}}%
}%
```

Finally set the name width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstablenamewidth}{\dimexpr\glstableblockwidth
   - \glstablesymbolwidth}%
  \ifdim\glstablenamewidth<0pt\relax
    \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstablesymbolwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block. v1.50 child entries now in name
column.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameNoDesc{##1}%
  \glstableChildEntries{##1}%
& \glstableSymbol{##1}}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameSymbolNoDesc{##1}}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstablenamewidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablenameheader &
  \glstableHeaderFmt\glstablesymbolheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstablenamecolalign\glstablesymbolcolalign}%
 }
```

```
\glstablenewstyle{desc-name}{%
```

2 columns per block (description, name).

```
\glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max name width if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablenamewidth}%
```

```
      {\glstableHeaderFmt\glstablenameheader}%
    \else
      \setlength{\glstablenamewidth}{0pt}%
    \fi
    \setlength{\glstabledescwidth}{0pt}%
  }%
```

Update width in unsrt hook.

```
  \renewcommand{\glstablelengthupdate}[1]{%
    \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
  }%
```

Finally set the description width to the remaining available.

```
  \renewcommand{\glstablefinishlengthupdates}{%
    \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
      - \glstablenamewidth}%
    \ifdim\glstabledescwidth<0pt\relax
      \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
      \setlength{\glstabledescwidth}{\glstablenamewidth}%
    \fi
  }%
```

How to format the top-level entry in the block. v1.50 child entries now in
description column

```
  \renewcommand{\glstableblockentry}[1]{%
    \glstableDescWithOther{##1}%
    \glstableChildEntries{##1}%
  &
    \glstableNameTarget{##1}%
  }%
```

How to format the entry's children.

```
  \renewcommand{\glstableblocksubentry}[1]{%
    \glstableSubDescWithOther{##1}\glstableSubNameSep
    \glstableSubNameTarget{##1}%
  }%
```

Available width for child entries.

```
  \renewcommand{\glstablesubentrywidth}{\glstabledescwidth}
```

How to format the block's header row, if required.

```
  \renewcommand{\glstableblockheader}{%
    \glstableHeaderFmt\glstabledescheader &
    \glstableHeaderFmt\glstablenameheader
  }%
```

Set the block's column alignments.

```
  \renewcommand{\glstableblockalign}{\glstabledesccolalign\glstablenamecolalign}%
 }
```

symbol-name

```
  \glstablenewstyle{symbol-name}{%
```

2 columns per block (symbol, name).

```
\glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max symbol width if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablesymbolwidth}%
      {\glstableHeaderFmt\glstablesymbolheader}%
  \else
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
  \setlength{\glstablenamewidth}{0pt}%
}%
```

Update widths in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbol{##1}}%
}%
```

Finally set the name width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstablenamewidth}{\dimexpr\glstableblockwidth
   - \glstablesymbolwidth}%
  \ifdim\glstablenamewidth<0pt\relax
    \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstablesymbolwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableSymbol{##1} &
  \glstableNameNoDesc{##1}%
  \glstableChildEntries{##1}%
%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubSymbolWithSep{}{##1}{\glstableSubSep}%
  \glstableSubNameNoDesc{##1}%
}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstablenamewidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablesymbolheader &
  \glstableHeaderFmt\glstablenameheader
}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstablesymbolcolalign\glstablenamecolalign}%
}
```

name-symbol-desc

```
\glstablenewstyle{name-symbol-desc}{%
```

3 columns per block (name, symbol, description).

```
\glstablecolsperblock=3\relax
```

Initialise length registers (need to calculate max name and symbol widths if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablenamewidth}%
      {\glstableHeaderFmt\glstablenameheader}%
    \glsmeasurewidth{\glstablesymbolwidth}%
      {\glstableHeaderFmt\glstablesymbolheader}%
  \else
    \setlength{\glstablenamewidth}{0pt}%
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
  \setlength{\glstabledescwidth}{0pt}%
}%
```

Update widths in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbol{##1}}%
}%
```

Finally set the description width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
   - \glstablesymbolwidth - \glstablenamewidth}%
  \ifdim\glstabledescwidth<0pt\relax
    \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth
   - 0.5\glstablesymbolwidth}%
    \setlength{\glstabledescwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameTarget{##1} &
  \glstableSymbol{##1} &
  \glstableDescWithOther{##1}%
  \glstableChildEntries{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
```

```
    \glstableSubNameTarget{##1}\glstableSubNameSep
    \glstableSubSymbolWithSep{}{##1}{\glstableSubSep}%
    \glstableSubDescWithOther{##1}%
  }%
```

Available width for child entries.

```
  \renewcommand{\glstablesubentrywidth}{\glstabledescwidth}
```

How to format the block's header row, if required.

```
  \renewcommand{\glstableblockheader}{%
    \glstableHeaderFmt\glstablenameheader &
    \glstableHeaderFmt\glstablesymbolheader &
    \glstableHeaderFmt\glstabledescheader}%
```

Set the block's column alignments.

```
  \renewcommand{\glstableblockalign}{%
    \glstablenamecolalign\glstablesymbolcolalign\glstabledesccolalign}%
  }
```

```
  \glstablenewstyle{name-other-desc}{%
```

3 columns per block (name, other, description).

```
  \glstablecolsperblock=3\relax
```

Initialise length registers (need to calculate max name and other widths if par align).

```
  \renewcommand{\glstableinitlengthupdates}{%
    \ifKV@printglosstable@header
      \glsmeasurewidth{\glstablenamewidth}%
        {\glstableHeaderFmt\glstablenameheader}%
      \glsmeasurewidth{\glstableotherwidth}%
        {\glstableHeaderFmt\glstableotherheader}%
    \else
      \setlength{\glstablenamewidth}{0pt}%
      \setlength{\glstableotherwidth}{0pt}%
    \fi
    \setlength{\glstabledescwidth}{0pt}%
  }%
```

Update widths in unsrt hook.

```
  \renewcommand{\glstablelengthupdate}[1]{%
    \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
    \glstablemeasureandupdate{\glstableotherwidth}{\glstableOther{##1}}%
  }%
```

Finally set the description width to the remaining available.

```
  \renewcommand{\glstablefinishlengthupdates}{%
    \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
      - \glstableotherwidth - \glstablenamewidth}%
    \ifdim\glstabledescwidth<0pt\relax
      \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth
```

```
        - 0.5\glstableotherwidth}%
         \setlength{\glstabledescwidth}{\glstablenamewidth}%
       \fi
    }%
```

How to format the top-level entry in the block.

```
  \renewcommand{\glstableblockentry}[1]{%
    \glstableNameTarget{##1} &
    \glstableOther{##1} &
    \glstableDesc{##1}%
    \glstableChildEntries{##1}%
 }%
```

How to format the entry's children.

```
  \renewcommand{\glstableblocksubentry}[1]{%
    \glstableSubNameTarget{##1}\glstableSubNameSep
    \glstableSubOtherWithSep{}{##1}{\glstableSubOtherSep}%
    \glstableSubDesc{##1}}%
```

Available width for child entries.

```
  \renewcommand{\glstablesubentrywidth}{\glstabledescwidth}
```

How to format the block's header row, if required.

```
  \renewcommand{\glstableblockheader}{%
    \glstableHeaderFmt\glstablenameheader &
    \glstableHeaderFmt\glstableotherheader &
    \glstableHeaderFmt\glstabledescheader}%
```

Set the block's column alignments.

```
  \renewcommand{\glstableblockalign}{%
    \glstablenamecolalign\glstableothercolalign\glstabledesccolalign}%
 }
```

desc-other-name As name-other-desc but with the end columns switched.

```
  \glstablenewstyle{desc-other-name}{%
    \glstablesetstyle{name-other-desc}%
```

How to format the top-level entry in the block.

```
  \renewcommand{\glstableblockentry}[1]{%
    \glstableDesc{##1}%
    \glstableChildEntries{##1} &
    \glstableOther{##1} &
    \glstableNameTarget{##1}%
  }%
```

How to format the entry's children.

```
  \renewcommand{\glstableblocksubentry}[1]{%
    \glstableSubDesc{##1}%
    \glstableSubOtherWithSep{\glstableSubOtherSep}{##1}{}%
    \glstableSubNameSep
    \glstableSubNameTarget{##1}%
  }%
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstabledescheader &
  \glstableHeaderFmt\glstableotherheader &
  \glstableHeaderFmt\glstablenameheader
}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{%
  \glstabledesccolalign
  \glstableothercolalign
  \glstablenamecolalign
}%
}
```

name-symbol-other-desc

```
\glstablenewstyle{name-symbol-other-desc}{%
```

4 columns per block (name, symbol, other, description).

```
\glstablecolsperblock=4\relax
```

Initialise length registers (need to calculate max name, symbol and other widths if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablenamewidth}%
      {\glstableHeaderFmt\glstablenameheader}%
    \glsmeasurewidth{\glstablesymbolwidth}%
      {\glstableHeaderFmt\glstablesymbolheader}%
    \glsmeasurewidth{\glstableotherwidth}%
      {\glstableHeaderFmt\glstableotherheader}%
  \else
    \setlength{\glstablenamewidth}{0pt}%
    \setlength{\glstablesymbolwidth}{0pt}%
    \setlength{\glstableotherwidth}{0pt}%
  \fi
  \setlength{\glstabledescwidth}{0pt}%
}%
```

Update widths in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbol{##1}}%
  \glstablemeasureandupdate{\glstableotherwidth}{\glstableOther{##1}}%
}%
```

Finally set the description width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
    - \glstablesymbolwidth - \glstablenamewidth - \glstableotherwidth}%
  \ifdim\glstabledescwidth<0pt\relax
```

Not enough room so balance them out evenly.

```
    \setlength{\glstablenamewidth}{\dimexpr0.25\glstableblockwidth}%
    \setlength{\glstablesymbolwidth}{\glstablenamewidth}%
    \setlength{\glstableotherwidth}{\glstablenamewidth}%
    \setlength{\glstabledescwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameTarget{##1} &
  \glstableSymbol{##1} &
  \glstableOther{##1} &
  \glstableDesc{##1}%
  \glstableChildEntries{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameTarget{##1}\glstableSubNameSep
  \glstableSubSymbolWithSep{}{##1}{\glstableSubSep}%
  \glstableSubOtherWithSep{}{##1}{\glstableSubOtherSep}%
  \glstableSubDesc{##1}}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstabledescwidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablenameheader &
  \glstableHeaderFmt\glstablesymbolheader &
  \glstableHeaderFmt\glstableotherheader &
  \glstableHeaderFmt\glstabledescheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{%
  \glstablenamecolalign
  \glstablesymbolcolalign
  \glstableothercolalign
  \glstabledesccolalign}%
}
```

name-desc-symbol

```
\glstablenewstyle{name-desc-symbol}{%
```

3 columns per block (name, description, symbol).

```
\glstablecolsperblock=3\relax
```

Initialise length registers (need to calculate max name and symbol widths if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
```

798

```
    \glsmeasurewidth{\glstablenamewidth}%
      {\glstableHeaderFmt\glstablenameheader}%
    \glsmeasurewidth{\glstablesymbolwidth}%
      {\glstableHeaderFmt\glstablesymbolheader}%
  \else
    \setlength{\glstablenamewidth}{0pt}%
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
  \setlength{\glstabledescwidth}{0pt}%
}%
```

Update widths in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbol{##1}}%
}%
```

Finally set the description width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
    - \glstablesymbolwidth - \glstablenamewidth}%
  \ifdim\glstabledescwidth<0pt\relax
    \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth
    - 0.5\glstablesymbolwidth}%
    \setlength{\glstabledescwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block. v1.50 child entries in description column.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameTarget{##1} &
  \glstableDescWithOther{##1}%
  \glstableChildEntries{##1}%
&
  \glstableSymbol{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameTarget{##1}\glstableSubNameSep
  \glstableSubDescWithOther{##1}%
  \glstableSubSymbolWithSep{\glstableSubSep}{##1}{}%
}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstabledescwidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablenameheader &
  \glstableHeaderFmt\glstabledescheader &
```

```
    \glstableHeaderFmt\glstablesymbolheader
  }%
```

Set the block's column alignments.

```
  \renewcommand{\glstableblockalign}{%
    \glstablenamecolalign\glstabledesccolalign\glstablesymbolcolalign}%
}
```

desc-symbol-other-name As name-symbol-other-desc but with the end columns switched.

```
\glstablenewstyle{desc-symbol-other-name}{%
  \glstablesetstyle{name-symbol-other-desc}%
```

How to format the top-level entry in the block.

```
  \renewcommand{\glstableblockentry}[1]{%
    \glstableDesc{##1}%
    \glstableChildEntries{##1} &
    \glstableSymbol{##1} &
    \glstableOther{##1} &
    \glstableNameTarget{##1}%
  }%
```

How to format the entry's children.

```
  \renewcommand{\glstableblocksubentry}[1]{%
    \glstableSubDescSymbolOther{##1}%
    \glstableSubNameSep
    \glstableSubNameTarget{##1}%
  }%
```

How to format the block's header row, if required.

```
  \renewcommand{\glstableblockheader}{%
    \glstableHeaderFmt\glstabledescheader &
    \glstableHeaderFmt\glstablesymbolheader &
    \glstableHeaderFmt\glstableotherheader &
    \glstableHeaderFmt\glstablenameheader
  }%
```

Set the block's column alignments.

```
  \renewcommand{\glstableblockalign}{%
    \glstabledesccolalign
    \glstablesymbolcolalign
    \glstableothercolalign
    \glstablenamecolalign
  }%
}
```

desc-other-symbol-name As name-symbol-other-desc but column order is description, other, symbol and name.

```
\glstablenewstyle{desc-other-symbol-name}{%
  \glstablesetstyle{name-symbol-other-desc}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableDesc{##1}%
  \glstableChildEntries{##1} &
  \glstableOther{##1} &
  \glstableSymbol{##1} &
  \glstableNameTarget{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubDesc{##1}%
  \glstableSubOtherWithSep{\glstableSubOtherSep}{##1}{}%
  \glstableSubSymbolWithSep{\glstableSubSep}{##1}{}%
  \glstableSubNameSep
  \glstableSubNameTarget{##1}%
}%
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstabledescheader &
  \glstableHeaderFmt\glstableotherheader &
  \glstableHeaderFmt\glstablesymbolheader &
  \glstableHeaderFmt\glstablenameheader
}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{%
  \glstabledesccolalign
  \glstableothercolalign
  \glstablesymbolcolalign
  \glstablenamecolalign
}%
}
```

`name-other-symbol-desc` As name-symbol-other-desc but column order is name, other, symbol and description.

```
\glstablenewstyle{name-other-symbol-desc}{%
  \glstablesetstyle{name-symbol-other-desc}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameTarget{##1} &
  \glstableOther{##1} &
  \glstableSymbol{##1} &
  \glstableDesc{##1}%
  \glstableChildEntries{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
```

```
      \glstableSubNameTarget{##1}\glstableSubNameSep
      \glstableSubOtherWithSep{}{##1}{\glstableSubOtherSep}%
      \glstableSubSymbolWithSep{}{##1}{\glstableSubSep}%
      \glstableSubDesc{##1}%
    }%
```

How to format the block's header row, if required.

```
    \renewcommand{\glstableblockheader}{%
      \glstableHeaderFmt\glstablenameheader &
      \glstableHeaderFmt\glstableotherheader &
      \glstableHeaderFmt\glstablesymbolheader &
      \glstableHeaderFmt\glstabledescheader
    }%
```

Set the block's column alignments.

```
    \renewcommand{\glstableblockalign}{%
      \glstablenamecolalign
      \glstableothercolalign
      \glstablesymbolcolalign
      \glstabledesccolalign
    }%
  }
```

name-other As name-desc but the other field is put in the description column.

```
    \glstablenewstyle{name-other}{%
```

2 columns per block (name, other).

```
    \glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max name width if par align).

```
    \renewcommand{\glstableinitlengthupdates}{%
      \ifKV@printglosstable@header
        \glsmeasurewidth{\glstablenamewidth}%
          {\glstableHeaderFmt\glstablenameheader}%
      \else
        \setlength{\glstablenamewidth}{0pt}%
      \fi
      \setlength{\glstableotherwidth}{0pt}%
    }%
```

Update width in unsrt hook.

```
    \renewcommand{\glstablelengthupdate}[1]{%
      \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
    }%
```

Finally set the other width to the remaining available.

```
    \renewcommand{\glstablefinishlengthupdates}{%
      \setlength{\glstableotherwidth}{\dimexpr\glstableblockwidth
        - \glstablenamewidth}%
      \ifdim\glstableotherwidth<0pt\relax
        \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
        \setlength{\glstableotherwidth}{\glstablenamewidth}%
```

```
    \fi
  }%
```

How to format the top-level entry in the block.

```
  \renewcommand{\glstableblockentry}[1]{%
    \glstableNameTarget{##1} & \glstableOtherNoDesc{##1}%
    \glstableChildEntries{##1}%
 }%
```

How to format the entry's children.

```
  \renewcommand{\glstableblocksubentry}[1]{%
    \glstableSubNameTarget{##1}\glstableSubNameSep \glstableSubOtherNoDesc{##1}}%
```

Available width for child entries.

```
  \renewcommand{\glstablesubentrywidth}{\glstableotherwidth}
```

How to format the block's header row, if required.

```
  \renewcommand{\glstableblockheader}{%
    \glstableHeaderFmt\glstablenameheader &
    \glstableHeaderFmt\glstableotherheader}%
```

Set the block's column alignments.

```
  \renewcommand{\glstableblockalign}{\glstablenamecolalign\glstableothercolalign}%
 }
```

other-name

```
  \glstablenewstyle{other-name}{%
```

2 columns per block (other, name).

```
  \glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max name width if par align).

```
  \renewcommand{\glstableinitlengthupdates}{%
    \ifKV@printglosstable@header
      \glsmeasurewidth{\glstablenamewidth}%
        {\glstableHeaderFmt\glstablenameheader}%
    \else
      \setlength{\glstablenamewidth}{0pt}%
    \fi
    \setlength{\glstableotherwidth}{0pt}%
  }%
```

Update width in unsrt hook.

```
  \renewcommand{\glstablelengthupdate}[1]{%
    \glstablemeasureandupdate{\glstablenamewidth}{\glstableName{##1}}%
  }%
```

Finally set the other width to the remaining available.

```
  \renewcommand{\glstablefinishlengthupdates}{%
    \setlength{\glstableotherwidth}{\dimexpr\glstableblockwidth
      - \glstablenamewidth}%
    \ifdim\glstableotherwidth<0pt\relax
      \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
```

```
    \setlength{\glstableotherwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block. v1.50 child entries in other column.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableOtherNoDesc{##1}%
  \glstableChildEntries{##1}%
&
  \glstableNameTarget{##1}}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubOtherNoDesc{##1}\glstableSubNameSep
  \glstableSubNameTarget{##1}}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstableotherwidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstableotherheader &
  \glstableHeaderFmt\glstablenameheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstableothercolalign\glstablenamecolalign}%
 }
```

`symbol-other` As name-other but the use the symbol in place of the name.

```
\glstablenewstyle{symbol-other}{%
```

2 columns per block (symbol, other).

```
\glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max symbol width if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablesymbolwidth}%
      {\glstableHeaderFmt\glstablesymbolheader}%
  \else
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
  \setlength{\glstableotherwidth}{0pt}%
}%
```

Update width in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolName{##1}}%
}%
```

Finally set the other width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstableotherwidth}{\dimexpr\glstableblockwidth
   - \glstablesymbolwidth}%
  \ifdim\glstableotherwidth<0pt\relax
    \setlength{\glstablesymbolwidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstableotherwidth}{\glstablesymbolwidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableSymbolNameTarget{##1} & \glstableOtherNoDesc{##1}%
  \glstableChildEntries{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubSymbolNameTarget{##1}\glstableSubNameSep
  \glstableSubOtherNoDesc{##1}}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstableotherwidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablesymbolheader &
  \glstableHeaderFmt\glstableotherheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstablesymbolcolalign\glstableothercolalign}%
}
```

other-symbol

```
\glstablenewstyle{other-symbol}{%
```

2 columns per block (other-symbol).

```
\glstablecolsperblock=2\relax
```

Initialise length registers (need to calculate max symbol width if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
    \glsmeasurewidth{\glstablesymbolwidth}%
      {\glstableHeaderFmt\glstablesymbolheader}%
  \else
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
  \setlength{\glstableotherwidth}{0pt}%
}%
```

Update width in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolName{##1}}%
}%
```

Finally set the other width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstableotherwidth}{\dimexpr\glstableblockwidth
   - \glstablesymbolwidth}%
  \ifdim\glstableotherwidth<0pt\relax
    \setlength{\glstablesymbolwidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstableotherwidth}{\glstablesymbolwidth}%
  \fi
}%
```

How to format the top-level entry in the block. v1.50 child entries in other column.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableOtherNoDesc{##1}%
  \glstableChildEntries{##1}%
& \glstableSymbolNameTarget{##1}}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubOtherWithSep{}{##1}{\glstableSubSep}%
  \glstableSubSymbol{##1}%
  \glstableSubNameSep
  \glstableSubSymbolNameTarget{##1}}%
```

Available width for child entries.

```
\renewcommand{\glstablesubentrywidth}{\glstableotherwidth}
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstableotherheader &
  \glstableHeaderFmt\glstablesymbolheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstableothercolalign\glstablesymbolcolalign}%
}
```

\glstablecaption

> $\glstablecaption\{\langle toc\ title\rangle\}\{\langle title\rangle\}\{\langle label\ code\rangle\}$

The $\langle label\ code\rangle$ will be \@@glossaryseclabel.

```
\newcommand{\glstablecaption}[3]{%
\caption[#1]{#3#2}%
}
```

\glstablepostnextcaption

```
\newcommand{\glstablepostnextcaption}{ (\MFUsentencecase{\glsxtrcontinuedname})}
```

\glstablenextcaption

> $\glstablenextcaption\{\langle toc\ title\rangle\}\{\langle title\rangle\}$

```
\newcommand{\glstablenextcaption}[2]{%
 \caption[]{#1\glstablepostnextcaption}%
}
```

\glstablefoot

> \glstablefoot{⟨*postamble*⟩}

```
\newcommand{\glstablefoot}[1]{}
```

\glstablelastfoot

> \glstablelastfoot{⟨*postamble*⟩}

```
\newcommand{\glstablelastfoot}[1]{\glstablerowspan{#1}}
```

\glstablehead

> \glstablehead{⟨*preamble*⟩}

```
\newcommand{\glstablehead}[1]{}
```

\glstablefirsthead

> \glstablefirsthead{⟨*preamble*⟩}

```
\newcommand{\glstablefirsthead}[1]{\glstablerowspan{#1}}
```

\glstablepostpreambleskip

```
\newlength\glstablepostpreambleskip
\setlength\glstablepostpreambleskip{5pt}
```

\glstableprepostambleskip

```
\newlength\glstableprepostambleskip
\setlength\glstableprepostambleskip{5pt}
```

\glstablefootstrut

```
\newcommand{\glstablefootstrut}{%
 \rule{0pt}{\dimexpr\baselineskip+\glstableprepostambleskip}%
}
```

\glstablerowspan

> \glstablerowspan{⟨*text*⟩}

```
\newcommand{\glstablerowspan}[1]{%
 \multicolumn{\glstabletotalcols}{c}{\parbox{\glstablespanwidth}{#1}}%
}
```

\glstablespanwidth This will be updated if column widths are measured. This width doesn't include \tabcolsep on either side. The default is to use \LTcapwidth, which may not be the same size as the table.

```
\newcommand{\glstablespanwidth}{\LTcapwidth}
```

\glstable@begin

```
\newcommand{\glstable@begin}{%
 \PackageError{glossary-table}{table style can only be used with
  \string\printunsrttable}{}%
}
```

\glstable@filter Filter all child entries, but take level offset into account and apply custom handler.

```
\newcommand{\glstable@filter}[1]{%
  \ifnum\glscurrententrylevel>0\relax
    \printunsrtglossaryskipentry
  \else
    \glstableiffilter{#1}%
    {\printunsrtglossaryskipentry}%
    {%
      \glstable@calclengths{\glstablelengthupdate{#1}}%
    }%
  \fi
}
```

\glstableiffilter

```
\newcommand{\glstableiffilter}[3]{#3}
```

\glstablenamewidth

```
\newlength\glstablenamewidth
```

\glstableblockwidth Maximum width available for each block.

```
\newlength\glstableblockwidth
```

\glstabledescwidth

```
\newlength\glstabledescwidth
```

\glstablesymbolwidth

```
\newlength\glstablesymbolwidth
```

\glstableotherwidth

```
\newlength\glstableotherwidth
```

\glstableifmeasuring

$\boxed{\texttt{\textbackslash glstableifmeasuring\{}\langle \textit{true} \rangle\texttt{\}\{}\langle \textit{false} \rangle\texttt{\}}}$

```
\newcommand{\glstableifmeasuring}[2]{#2}
```

`\glstable@stepentry`

```
\newcommand{\glstable@stepentry}[1]{%
 \ifglsentrycounter
  \stepcounter{glossaryentry}%
 \fi
}
```

`\glstable@stepsubentry`

```
\newcommand{\glstable@stepsubentry}[1]{%
 \ifglssubentrycounter
  \stepcounter{glossarysubentry}%
 \fi
}
```

> `\glstablemeasureandupdate{⟨len reg⟩}{⟨text⟩}`

`\glstablemeasureandupdate`

```
\newcommand{\glstablemeasureandupdate}[2]{%
```
Measure.
```
   \glsmeasurewidth{\dimen@}{#2}%
```
Update if wider.
```
   \ifdim\dimen@>#1\relax
     \setlength{#1}{\dimen@}%
   \fi
}
```

`\glstable@ifhaspreamble`

```
\newcommand{\glstable@ifhaspreamble}[2]{%
 \ifdefempty\glossarypreamble
 {#2}%
 {%
  \ifx\@glstable@defaultpreamble\glossarypreamble
    \ifcsvoid{@glossarypreamble@\currentglossary}{#2}{#1}%
  \else
    #1%
  \fi
 }%
}
```

Need the type, preamble and postamble.
```
\define@key{printglosstable}{type}{\renewcommand{\@glo@type}{#1}}
\define@key{printglosstable}{preamble}{\renewcommand{\glossarypreamble}{#1}}
\define@key{printglosstable}{postamble}{\renewcommand{\glossarypostamble}{#1}}
```
Allow localised initialisation.

`\glstable@init`

```
\newcommand\glstable@init{}
```

```
\define@cmdkey{printglosstable}[glstable@]{init}{}
```

The default setting is groups=false, unlike the usual default for \printunsrtglossary.
Support for groups isn't fully implemented.

```
\define@choicekey{printglosstable}{groups}
[\@glstable@groups@val\@glstable@groups@n]
{false,true,noskip,addskip}[true]%
{%
 \ifcase\@glstable@groups@n\relax
  \let\glstable@groupheading\@gobble
  \glsxtr@printgloss@groupsfalse
 \or
  \let\glstable@groupheading\glstablegroupheading
  \glsxtr@printgloss@groupstrue
 \or
  \let\glstable@groupheading\glstablegroupheading
  \glsxtr@printgloss@groupstrue
  \glsnogroupskiptrue
 \or
  \let\glstable@groupheading\glstablegroupheading
  \glsxtr@printgloss@groupstrue
  \glsnogroupskipfalse
 \fi
}
```

\glstable@groupheading

```
\newcommand{\glstable@groupheading}[1]{}
```

\glstablegroupheading This isn't quite working as it puts a spurious line above if it occurs at the start
of a new row.

```
\newcommand{\glstablegroupheading}[1]{%
 \multicolumn{\glstabletotalcols}{c}{%
   \glsxtrgetgrouptitle{#1}{\glsxtrcurrentgrptitle}%
   \glstableGroupHeaderFmt\glsxtrcurrentgrptitle
 }%
 \glstablePostGroupNewLine
}
```

\glstablePostGroupNewLine

```
\newcommand{\glstablePostGroupNewLine}{\glstablenewline*}
```

\glstableGroupHeaderFmt

```
\newcommand{\glstableGroupHeaderFmt}{\glstableHeaderFmt}
```

\glstable@preentryhook

```
\newcommand{\glstable@preentryhook}[1]{%
 \if@glstable@afterheading
 \else
  \advance\glstablecurrentblockindex by 1\relax
```

```
                           \ifnum\glstablecurrentblockindex<\glstableblockperrowcount
                             \appto#1{&}%
                           \else
                             \appto#1{\glstablenewline}%
                           \fi
                         \fi
                       }

        \glstablenewline
                       \newcommand{\glstablenewline}{\tabularnewline}

  \glstable@postentryhook
                       \newcommand{\glstable@postentryhook}[1]{%
                        \ifnum\glstableblockperrowcount=\glstablecurrentblockindex
                          \glstablecurrentblockindex=0\relax
                        \fi
                        \@glstable@afterheadingfalse
                       }

       \glstable@grouphook
                       \newcommand{\glstable@grouphook}[1]{%
                         \if@glstable@afterheading
                         \else
                           \preto#1{\glstablenewline}%
                           \advance\glstablecurrentblockindex by 1\relax
                           \ifnum\glstablecurrentblockindex<\glstableblockperrowcount\relax
                             \expandafter\glstable@n@to@amps\expandafter
                             {\numexpr\glstableblockperrowcount-\glstablecurrentblockindex}%
                             {\preto}{#1}%
                           \fi
                         \fi
                         \glstablecurrentblockindex=0\relax
                         \@glstable@afterheadingtrue
                       }

         \glstable@finish
                       \newcommand{\glstable@finish}[1]{%
                         \if@glstable@afterheading
                         \else
                           \advance\glstablecurrentblockindex by 1\relax
                           \ifnum\glstablecurrentblockindex<\glstableblockperrowcount\relax
                             \expandafter\glstable@n@to@amps\expandafter
                             {\numexpr\glstableblockperrowcount-\glstablecurrentblockindex}%
                             {\appto}{#1}%
                           \fi
                         \fi
                       }

\@glstable@defaultpreamble
                       \let\@glstable@defaultpreamble\glossarypreamble
```

`\@glstable@clearpage`

```
\newcommand{\@glstable@clearpage}{}
```

`\@glstable@clearpage@iflt` Clear page if less than given length available.

```
\newcommand{\@glstable@clearpage@iflt}[1]{%
  \par
  \ifdim #1>\dimexr\pagegoal-\pagetotal\relax
   \clearpage
  \fi
}
```

Allow `\clearpage` to be inserted.

```
\define@key{printglosstable}{clearpage}[true]{%
  \ifstrequal{#1}{true}%
  {%
    \renewcommand{\@glstable@clearpage}{\clearpage}%
  }%
  {%
    \ifstrequal{#1}{false}%
    {%
      \renewcommand{\@glstable@clearpage}{}%
    }%
    {%
      \renewcommand{\@glstable@clearpage}{\@glstable@clearpage@iflt{#1}}%
    }%
  }%
}
```

`\if@glstable@afterheading`

```
\newif\if@glstable@afterheading
```

`\printunsrttable`

```
\NewDocumentCommand\printunsrttable{O{}}{%
 \bgroup
```
Initialise glossary type.
```
 \def\@glo@type{\glsdefaulttype}%
```
Initialise title.
```
 \def\glossarytitle{%
  \ifcsdef{@glotype@\@glo@type @title}%
  {\csuse{@glotype@\@glo@type @title}}%
  {\glossaryname}%
 }%
 \def\glossarytoctitle{\glossarytitle}%
```
Initialise preamble.
```
 \let\glossarypreamble\@glstable@defaultpreamble
```
Initialise groups=false.
```
 \glsxtr@printgloss@groupsfalse
```

Initialise nogroupskip=true.

```
\glsnogroupskiptrue
```

Set table keys.

```
\setkeys*{printglosstable}{#1}%
%\changes{1.50}{2022-11-08}{added check for caption and floats options}
```

If this table should have a caption, check the floats package option to determine whether or not to switch counter. Can be counteracted by redefining `\glscounter` in init code.

```
\ifKV@printglosstable@caption
 \if@glsxtr@floats
  \renewcommand{\glscounter}{table}%
 \fi
\fi
```

Initialisation hook.

```
\glstable@init
```

Should lengths be calculated?

```
\let\glstable@calclengths\glstableifpar
```

Has nogroupskip=false been used?

```
\ifglsnogroupskip
\else
 \ifundef\glspenaltygroupskip
 {%
   \PackageError{glossary-table}{\string\printunsrttable[nogroupskip=false]
   requires glossary-longbooktabs.sty}%
   {You need to load glossary-longbooktabs.sty in addition to
   loading glossary-table.sty if you want the group skip}%
   \glsnogroupskiptrue
 }%
 {\glspatchLToutput}%
\fi
\let\currentglossary\@glo@type
\protected@edef\glstable@opts{type=\@glo@type,style=table}%
\ifdefempty\XKV@rm{}{\epreto\glstable@opts{\expandonce\XKV@rm,}}%
```

Calculate the total number of columns.

```
\glstabletotalcols=\numexpr\glstablecolsperblock*\glstableblockperrowcount\relax
```

If the widest name is non-void, calculate the remaining width available for the blocks. 1pt is subtracted to allow for rounding errors.

```
\glstable@calclengths
{%
  \edef\glstablespanwidth{\dimexpr\linewidth-2\tabcolsep-1pt}%
  \glstableblockwidth=\dimexpr
   (\linewidth-\glstabletotalcols\tabcolsep-\glstabletotalcols\tabcolsep)
    /\glstableblockperrowcount-1pt
    \relax
  \glstableinitlengthupdates
}%
```

Build the header row.

```
\def\glstable@alignment{}%
\ifKV@printglosstable@rules
  \def\glstable@header{\toprule}%
\else
  \def\glstable@header{}%
\fi
\global\glstablecurrentblockindex=0\relax
\loop
```

Add to alignment spec.

```
  \ifnum\glstablecurrentblockindex>0\relax
   \protected@eappto\glstable@alignment{\glstable@blockalignsep}%
  \fi
  \protected@eappto\glstable@alignment{\glstableblockalign}%
  \ifKV@printglosstable@header
```

Add to header.

```
   \ifnum\glstablecurrentblockindex>0\relax
    \appto\glstable@header{&}%
   \fi
   \eappto\glstable@header{\expandonce\glstableblockheader}%
  \fi
```

Increment loop counter

```
   \advance\glstablecurrentblockindex by 1\relax
  \ifnum\glstablecurrentblockindex<\glstableblockperrowcount
  \repeat
  \ifKV@printglosstable@header
```

Append cr to header.

```
  \appto\glstable@header{\glstablenewline}%
  \ifKV@printglosstable@rules
    \appto\glstable@header{\midrule}%
  \fi
\fi
\protected@edef\glstable@begin{%
  \noexpand\begin{longtable}{\expandonce\glstable@alignment}%
}%
```

Use `\expandafter` after to allow an empty `\glossarytoctitle` to prevent the caption from being added to the table of contents.

```
\ifKV@printglosstable@caption
  \appto\glstable@begin{%
    \expandafter\glstablecaption\expandafter
      {\glossarytoctitle}{\glossarytitle}%
      {\@@glossaryseclabel}%
    \glstablenewline
  }%
\fi
```

Add preamble if set.

```
\glstable@ifhaspreamble
{%
  \eappto\glstable@begin{%
    \noexpand\glstablefirsthead
     {\expandonce\glossarypreamble}%
    \noexpand\glstablenewline[\glstablepostpreambleskip]%
    \expandonce\glstable@header
    \noexpand\endfirsthead
  }%
  \ifKV@printglosstable@caption
    \appto\glstable@begin{%
      \expandafter\glstablenextcaption\expandafter
       {\glossarytoctitle}{\glossarytitle}%
      \glstablenewline
    }%
  \fi
  \ifx\glstablehead\@gobble
  \else
    \eappto\glstable@begin{%
      \noexpand\glstablehead{\expandonce\glossarypreamble}%
      \noexpand\glstablenewline[\glstablepostpreambleskip]%
    }%
  \fi
}%
{%
  \eappto\glstable@begin{%
    \expandonce\glstable@header
    \noexpand\endfirsthead
  }%
  \ifKV@printglosstable@caption
    \appto\glstable@begin{%
      \expandafter\glstablenextcaption\expandafter
       {\glossarytoctitle}{\glossarytitle}%
      \glstablenewline
    }%
  \fi
}%
\eappto\glstable@begin{%
  \expandonce\glstable@header
  \noexpand\endhead
}%
```

Add postamble if set.

```
\ifdefvoid\glossarypostamble
{%
```

Just add rule, if required.

```
\ifKV@printglosstable@rules
  \appto\glstable@begin{\bottomrule\endfoot}%
```

```
    \fi
}
{%
  \ifKV@printglosstable@rules
    \eappto\glstable@begin{%
     \noexpand\bottomrule
     \noexpand\glstablefoot
       {\noexpand\glstablefootstrut\expandonce\glossarypostamble}%
     \noexpand\glstablenewline
     \noexpand\endfoot
     \noexpand\bottomrule
     \noexpand\glstablelastfoot
       {\noexpand\glstablefootstrut\expandonce\glossarypostamble}%
     \noexpand\glstablenewline
     \noexpand\endlastfoot
    }%
  \else
    \eappto\glstable@begin{%
     \noexpand\glstablefoot{\expandonce\glossarypostamble}%
     \noexpand\glstablenewline[\glstableprepostambleskip]%
     \noexpand\endfoot
     \noexpand\glstablelastfoot{\expandonce\glossarypostamble}%
     \noexpand\glstablenewline[\glstableprepostambleskip]%
     \noexpand\endlastfoot
    }%
  \fi
}%
```

Set up filtering.

```
\let\printunsrtglossaryentryprocesshook\glstable@filter
```

Use the hooks to add tab and new lines to avoid awkward conditionals within
longtable.

```
\renewcommand{\printunsrtglossarypreentryprocesshook}{%
  \glstable@preentryhook
}%
\renewcommand{\printunsrtglossarypostentryprocesshook}{%
  \glstable@postentryhook
}%
\renewcommand{\printunsrtglossarygrouphook}{%
  \glstable@grouphook
}%
\renewcommand{\printunsrtglossarypreend}{%
  \glstable@finish
}%
```

Disable preamble and postamble commands as their content has already been
added to the table specs.

```
\let\glossarypostamble\relax
\let\glossarypreamble\relax
```

Disable the section command as the title and toc title are now in the caption.

```
\renewcommand{\glossarysection}[2][]{}%
```

Used in hooks.

```
\glstablecurrentblockindex=0\relax
\@glstable@afterheadingtrue
```

Clear page if required.

```
\@glstable@clearpage
```

Finish updating lengths in hook.

```
\let\glstable@org@predoglossary\printunsrtglossarypredoglossary
\renewcommand{\printunsrtglossarypredoglossary}
{%
  \glstable@calclengths{\glstablefinishlengthupdates}%
  \glstable@org@predoglossary
}%
```

The glossary will be empty on the first LaTeX run as the entries won't be defined until `bib2gls` has selected them.

```
\glsxtrifemptyglossary{\currentglossary}
{%
  \GlossariesExtraWarning{Glossary '\currentglossary' is empty}%
```

Just do the table header and footer to allow it to be added to the list of tables and have the label added to the aux file.

```
  \edef\@glsxtr@tmp{\noexpand\setkeys{printgloss}{\expandonce\glstable@opts}}%
  \@glsxtr@tmp
  \glstable@begin% \begin{longtable}{specs}
  \end{longtable}%
}%
{%
  \expandafter\printunsrtglossary\expandafter[\glstable@opts]\relax
}%
\egroup
}
```

\glstableiffilterchild

```
\newcommand{\glstableiffilterchild}[3]{#3}
```

\glstable@child

```
\newcommand{\glstable@child}[1]{%
\glstableiffilterchild{#1}{}%
{%
  \ifdefempty\glstable@dochildren{}%
  {\appto\glstable@dochildren{\glstableblocksubentrysep}}%
  \appto\glstable@dochildren{\glstableblocksubentry{#1}}%
}%
}
```

\glstableChildEntries

```
\newcommand{\glstableChildEntries}[1]{%
  \def\glstable@dochildren{}%
  \GlsXtrIfFieldNonZero*{childcount}{#1}%
  {%
    \glsxtrfieldforlistloop{#1}{childlist}{\glstable@child}%
    \ifdefempty\glstable@dochildren
    {}%
    {%
      \preto\glstable@dochildren{%
        \glstablePreChildren
        \begin{glstablesubentries}%
      }%
      \appto\glstable@dochildren{\end{glstablesubentries}}%
    }%
  }%
  {}%
  \glstable@dochildren
}
```

\glstable@n@amps Removed.

\glstable@n@to@amps

```
\newcommand{\glstable@n@to@amps}[3]{%
 \ifnum#1>0\relax
   \count@=0\relax
   \loop
     \advance\count@ by 1\relax
     #2#3{&}%
   \ifnum\count@<#1
   \repeat
 \fi
}
```

\glstablefinishrow Removed in v1.50.

table

```
\newglossarystyle{table}%
{%
  \renewenvironment{theglossary}%
  {%
   \glstable@begin
  }
  {%
    \end{longtable}%
  }%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{\glstable@groupheading{##1}}%
  \renewcommand*{\glssubgroupheading}[4]{}%
  \ifglsnogroupskip
```

```
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
  \renewcommand{\glossentry}[2]{%
    \glstableblockentry{##1}%
```
v1.50: `\glstableChildEntries` moved to block style and conditionals moved to processing hooks.
```
    }%
    \renewcommand{\subglossentry}[3]{}%
}
```

9 Rollback Files

9.1 Rollback v1.48 (glossaries-extra-2021-11-22.sty)

Version 1.48 preserved for rollback.
```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-extra}[2021/11/22 v1.48 (NLCT)]
\RequirePackage{xkeyval}
\RequirePackage{etoolbox}
\@ifpackageloaded{glossaries}
{%
  \newcommand{\glsxtr@dooption}[1]{\setupglossaries{#1}}%
  \let\@glsxtr@declareoption\@gls@declareoption
}
{%
  \newcommand{\glsxtr@dooption}[1]{%
    \PassOptionsToPackage{#1}{glossaries}%
  }%
  \PassOptionsToPackage{toc}{glossaries}
  \PassOptionsToPackage{nopostdot}{glossaries}
  \PassOptionsToPackage{noredefwarn}{glossaries}
  \@ifpackageloaded{polyglossia}%
  {}%
  {%
    \@ifpackageloaded{babel}%
    {\PassOptionsToPackage{translate=babel}{glossaries}}%
    {}%
  }%
  \newcommand*{\@glsxtr@declareoption}[2]{%
    \DeclareOptionX{#1}{#2}%
    \DeclareOption{#1}{#2}%
  }
}
\newcommand*{\glsxtrundefaction}[2]{%
  \@glsxtrundeftag\PackageError{glossaries-extra}{#1}{#2}%
}
```

```
\newcommand*{\glsxtr@warnonexistsordo}[1]{}
\newcommand*{\glsxtrundeftag}{??}
\newcommand*{\@glsxtrundeftag}{}
\newcommand*{\@glsxtr@warn@undefaction}[2]{%
  \@glsxtrundeftag\GlossariesExtraWarning{#1}%
}
\newcommand*{\@glsxtr@err@undefaction}[2]{%
  \@glsxtrundeftag\PackageError{glossaries-extra}{#1}{#2}%
}
\newcommand*{\@glsxtr@warn@onexistsordo}[1]{%
  \GlossariesExtraWarning{\string#1\space hasn't been defined, so
  some errors won't be converted to warnings.
  (This most likely means your version of
  glossaries.sty is below version 4.19.)}%
}

\newcommand*{\@glsxtr@redef@forglsentries}{}
\newcommand*{\@glsxtr@do@redef@forglsentries}{%
  \renewcommand*{\forglsentries}[3][\glsdefaulttype]{%
    \protected@edef\@@glo@list{\csname glolist@##1\endcsname}%
    \ifdefstring{\@@glo@list}{,}%
    {%
      \GlossariesExtraWarning{No entries defined in glossary '##1'}%
    }%
    {%
      \@for##2:=\@@glo@list\do
      {%
        \ifdefempty{##2}{}{##3}%
      }%
    }%
  }%
}%
\define@choicekey{glossaries-extra.sty}{undefaction}%
 [\glsxtr@undefaction@val\glsxtr@undefaction@nr]%
 {warn,error}%
 {%
   \ifcase\glsxtr@undefaction@nr\relax
     \let\glsxtrundefaction\@glsxtr@warn@undefaction
     \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
     \let\@glsxtr@redef@forglsentries\@glsxtr@do@redef@forglsentries
   \or
     \let\glsxtrundefaction\@glsxtr@err@undefaction
     \let\glsxtr@warnonexistsordo\@gobble
     \let\@glsxtr@redef@forglsentries\relax
   \fi
 }
\newcommand*{\@glsxtr@record}[3]{}
\newcommand*{\glsxtr@recordsee}[2]{}
\newcommand*{\@glsxtr@defaultnumberformat}{glsnumberformat}%
\newcommand*{\GlsXtrSetDefaultNumberFormat}[1]{%
```

```
    \renewcommand*{\@glsxtr@defaultnumberformat}{#1}%
}%
\newcommand*{\@glsxtr@do@record@wrglossary}[1]{%
 \begingroup
   \ifKV@glslink@noindex
   \else
     \protected@edef\@gls@label{\glsdetoklabel{#1}}%
     \let\glslabel\@gls@label
     \glswriteentry{#1}%
     {%
       \ifdefempty{\@glsxtr@thevalue}%
       {%
         \ifx\@glsxtr@org@theHvalue\@glsxtr@theHvalue
         \else
           \let\theHglsentrycounter\@glsxtr@theHvalue
         \fi
         \glsxtr@saveentrycounter
         \let\@@do@@wrglossary\@glsxtr@dorecord
       }%
       {%
         \let\theglsentrycounter\@glsxtr@thevalue
         \let\theHglsentrycounter\@glsxtr@theHvalue
         \let\@@do@@wrglossary\@glsxtr@dorecordnodefer
       }%
       \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
         \glsxtr@@do@wrglossary{#1}%
       \else
         \@@glsxtrwrglossmark
         \glsxtr@inc@wrglossaryctr{#1}%
         \@@do@@wrglossary
       \fi
     }%
   \fi
 \endgroup
}
\newcommand*{\glsxtr@do@alsoindex@wrglossary}[1]{%
  \glsxtr@@do@wrglossary{#1}%
  \@glsxtr@do@record@wrglossary{#1}%
}
\newcommand*{\@@glsxtr@record}[3]{%
 \protected@edef\@gls@label{\glsdetoklabel{#2}}%
 \let\glslabel\@gls@label
 \ifglsentryexists{#2}{}%
 {%
   \@@glsxtrwrglossmark
   \begingroup
     \let\@glsnumberformat\@glsxtr@defaultnumberformat
     \def\@glsxtr@thevalue{}%
     \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
     \let\@glsxtr@org@theHvalue\@glsxtr@theHvalue
```

```
        \let\@gls@counter\glscounter
        \if@glsxtr@equations
          \@glsxtr@use@equation@counter
        \fi
        \@gls@setdefault@glslink@opts
        \csuse{@glsxtr@#3@prekeys}%
        \setkeys{#3}{#1}%
        \glsxtr@do@autoadd{#3}%
        \csuse{@glsxtr@#3@postkeys}%
        \glsxtr@inc@wrglossaryctr{#2}%
        \ifKV@glslink@noindex
        \else
          \glswriteentry{#2}%
          {%
            \ifdefempty{\@glsxtr@thevalue}%
            {%
              \ifx\@glsxtr@org@theHvalue\@glsxtr@theHvalue
              \else
                \let\theHglsentrycounter\@glsxtr@theHvalue
              \fi
              \glsxtr@saveentrycounter
              \let\@@do@@wrglossary\@glsxtr@dorecord
            }%
            {%
              \let\theglsentrycounter\@glsxtr@thevalue
              \let\theHglsentrycounter\@glsxtr@theHvalue
              \let\@@do@@wrglossary\@glsxtr@dorecordnodefer
            }%
            \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
              \glsxtr@@do@wrglossary{#2}%
            \else
              \@@do@@wrglossary
            \fi
          }%
        \fi
      \endgroup
 }%
}
\newcommand{\@glsxtr@glslink@prekeys}{\glslinkpresetkeys}
\newcommand{\@glsxtr@glslink@postkeys}{\glslinkpostsetkeys}
\newcommand{\@glsxtr@glossadd@prekeys}{\glsaddpresetkeys}
\newcommand{\@glsxtr@glossadd@postkeys}{\glsaddpostsetkeys}
\newcommand*\@glsxtr@dorecord{%
    \global\let\@glsrecordlocref\theglsentrycounter
    \let\@glsxtr@orgprefix\@glo@counterprefix
    \ifx\theglsentrycounter\theHglsentrycounter
      \def\@glo@counterprefix{}%
    \else
      \protected@edef\@glsxtr@theentrycounter{\theglsentrycounter}%
      \protected@edef\@glsxtr@theHentrycounter{\theHglsentrycounter}%
```

```
    \@onelevel@sanitize\@glsxtr@theentrycounter
    \@onelevel@sanitize\@glsxtr@theHentrycounter
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glsxtr@theentrycounter}{\@glsxtr@theHentrycounter}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
    \@glsxtr@do@nameref@record
     {\@gls@label}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}%
     {\@glsrecordlocref}%
  \else
    \protected@write\@auxout{}{\string\glsxtr@record
       {\@gls@label}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}%
       {\@glsrecordlocref}}%
  \fi
  \@glsxtr@counterrecordhook
  \let\@glo@counterprefix\@glsxtr@orgprefix
}
\newcommand*\@glsxtr@dorecordnodefer{%
  \ifx\theglsentrycounter\theHglsentrycounter
    \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
      \@glsxtr@do@nameref@record
         {\@gls@label}{}{\@gls@counter}{\@glsnumberformat}%
         {\theglsentrycounter}%
    \else
      \protected@write\@auxout{}{\string\glsxtr@record
         {\@gls@label}{}{\@gls@counter}{\@glsnumberformat}%
         {\theglsentrycounter}}%
    \fi
  \else
    \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\theglsentrycounter}{\theHglsentrycounter}%
    }%
    \@do@gls@getcounterprefix
    \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
      \@glsxtr@do@nameref@record
         {\@gls@label}{\@glo@counterprefix}{\@gls@counter}%
         {\@glsnumberformat}{\theglsentrycounter}%
    \else
      \protected@write\@auxout{}{\string\glsxtr@record
         {\@gls@label}{\@glo@counterprefix}{\@gls@counter}{\@glsnumberformat}%
         {\theglsentrycounter}}%
    \fi
  \fi
  \@glsxtr@counterrecordhook
}
\newcommand{\@glsxtr@ifnum@mmode}[2]{%
 \ifmmode
   \ifst@rred
```

```
      #2%
    \else
      \if@display #1\else #2\fi
    \fi
 \else
    #2%
 \fi
}
\newcommand*{\@glsxtr@do@nameref@record}[5]{%
  \gls@ifnotmeasuring
  {%
    \protected@write\@auxout{}{\string\glsxtr@record@nameref
      {#1}{#2}{#3}{#4}{#5}%
      {\csuse{@currentlabelname}}{\csuse{@currentHref}}%
      {\theHglsentrycounter}}%
  }%
}
\newcommand*{\@@glsxtr@recordcounter}{%
  \@glsxtr@noop@recordcounter
}
\newcommand*{\@glsxtr@noop@recordcounter}[1]{%
  \PackageError{glossaries-extra}{\string\GlsXtrRecordCounter\space
    requires record=only or record=hybrid package option}{}%
}
\newcommand*{\@glsxtr@op@recordcounter}[1]{%
  \protected@eappto\@glsxtr@counterrecordhook{\noexpand\@glsxtr@docounterrecord{#1}}%
}
\newcommand*{\@glsxtr@recordsee}[2]{%
 \@@glsxtrwrglossmark
 \def\@gls@xref{#2}%
 \@onelevel@sanitize\@gls@xref
 \protected@write\@auxout{}{\string\glsxtr@recordsee{#1}{\@gls@xref}}%
}
\newcommand{\printunsrtglossaryunit}{%
  \print@noop@unsrtglossaryunit
}
\newcommand*{\glsxtr@setup@record}{\let\@@do@wrglossary\glsxtr@@do@wrglossary}
\newcommand*{\glsxtr@indexonly@saveentrycounter}{%
 \ifKV@glslink@noindex
 \else
   \glsxtr@saveentrycounter
 \fi
}
\newcommand*{\glsxtr@addloclistfield}{%
 \key@ifundefined{glossentry}{loclist}%
 {%
   \define@key{glossentry}{loclist}{\def\@glo@loclist{##1}}%
   \appto\@gls@keymap{,{loclist}{loclist}}%
   \appto\@newglossaryentryprehook{\def\@glo@loclist{}}%
   \appto\@newglossaryentryposthook{%
```

```
      \gls@assign@field{}{\@glo@label}{loclist}{\@glo@loclist}%
    }%
    \glssetnoexpandfield{loclist}%
  }%
  {}%
  \key@ifundefined{glossentry}{location}%
  {%
    \define@key{glossentry}{location}{\def\@glo@location{##1}}%
    \appto\@gls@keymap{,{location}{location}}%
    \appto\@newglossaryentryprehook{\def\@glo@location{}}%
    \appto\@newglossaryentryposthook{%
      \gls@assign@field{}{\@glo@label}{location}{\@glo@location}%
    }%
    \glssetnoexpandfield{location}%
  }%
  {}%
  \key@ifundefined{glossentry}{group}%
  {%
    \define@key{glossentry}{group}{\def\@glo@group{##1}}%
    \appto\@gls@keymap{,{group}{group}}%
    \appto\@newglossaryentryprehook{\def\@glo@group{}}%
    \appto\@newglossaryentryposthook{%
      \gls@assign@field{}{\@glo@label}{group}{\@glo@group}%
    }%
    \glssetnoexpandfield{group}%
  }%
  {}%
}
\newcommand*{\@glsxtr@record@setting}{off}
\newcommand*{\@glsxtr@record@setting@alsoindex}{alsoindex}
\newcommand*{\@glsxtr@record@setting@only}{only}
\newcommand*{\@glsxtr@record@setting@nameref}{nameref}
\newcommand*{\@glsxtr@if@record@only}[2]{%
 \ifx\@glsxtr@record@setting\@glsxtr@record@setting@only
  #1%
 \else
  \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
    #1%
  \else
    #2%
  \fi
 \fi
}
\newcommand*{\@glsxtr@record@setting@off}{off}
\newcommand\@glsxtr@warn@hybrid@noprintgloss{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesExtraWarningNoLine{No glossaries have been defined}%
  }%
  {%
```

```
    \GlossariesExtraWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^JYou have requested the hybrid setting
      record=\@glsxtr@record@setting\space which requires a
      combination of bib2gls (to fetch entries) and makeindex/xindy
      (to sort and collate the entries). If you only want to use
      bib2gls then change the option to record=only or record=nameref}%
  }%
}
\newcommand*{\@glsxtr@record@only@setup}{%
 \def\glsxtr@setup@record{%
   \@glsxtr@autoseeindexfalse
   \let\@do@seeglossary\@glsxtr@recordsee
   \let\@glsxtr@record\@@glsxtr@record
   \let\@@do@wrglossary\@glsxtr@do@record@wrglossary
   \let\@gls@saveentrycounter\relax
   \let\glsxtrundefaction\@glsxtr@warn@undefaction
   \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
   \glsxtr@addloclistfield
   \renewcommand*{\@glsxtr@autoindexcrossrefs}{}%
   \let\@@glsxtr@recordcounter\@glsxtr@op@recordcounter
   \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
   \def\glsxtrsetaliasnoindex{}%
   \ifdef\@gls@setupsort@none{\@gls@setupsort@none}{}%
   \def\glsxtrNoGlossaryWarning{\@glsxtr@record@noglossarywarning}%
   \RequirePackage{glossaries-extra-bib2gls}[=v1.48]%
 }%
}
\define@choicekey{glossaries-extra.sty}{record}
 [\@glsxtr@record@setting\glsxtr@record@nr]%
 {off,only,alsoindex,nameref,hybrid}%
 [only]%
 {%
   \ifcase\glsxtr@record@nr\relax
     \def\glsxtr@setup@record{%
       \renewcommand*{\@do@seeglossary}{\@glsxtr@doseeglossary}%
       \renewcommand*{\@glsxtr@record}[3]{}%
       \let\@@do@wrglossary\glsxtr@@do@wrglossary
       \let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
       \let\glsxtrundefaction\@glsxtr@err@undefaction
       \let\glsxtr@warnonexistsordo\@gobble
       \let\@@glsxtr@recordcounter\@glsxtr@noop@recordcounter
       \def\printunsrtglossaryunit{\print@noop@unsrtglossaryunit}%
       \undef\glsxtrsetaliasnoindex
     }%
   \or
     \@glsxtr@record@only@setup
   \or
     \def\glsxtr@setup@record{%
       \renewcommand*{\@glsxtr@record@setting@alsoindex}{alsoindex}%
```

```
      \renewcommand*{\@do@seeglossary}{\@glsxtr@dosee@alsoindex@glossary}%
      \let\@glsxtr@record\@@glsxtr@record
      \let\@@do@wrglossary\glsxtr@do@alsoindex@wrglossary
      \let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
      \let\glsxtrundefaction\@glsxtr@warn@undefaction
      \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
      \glsxtr@addloclistfield
      \let\@@glsxtr@recordcounter\@glsxtr@op@recordcounter
      \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
      \undef\glsxtrsetaliasnoindex
    }%
  \or
    \@glsxtr@record@only@setup
    \ifundef\hyperlink
    {\GlossariesExtraWarning{You have requested record=nameref but
     the document doesn't support hyperlinks}}%
    {}%
  \or
    \def\glsxtr@setup@record{%
      \renewcommand*{\@glsxtr@record@setting@alsoindex}{hybrid}%
      \renewcommand*{\@do@seeglossary}{\@glsxtr@dosee@alsoindex@glossary}%
      \let\@glsxtr@record\@@glsxtr@record
      \let\@@do@wrglossary\glsxtr@do@alsoindex@wrglossary
      \let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
      \let\glsxtrundefaction\@glsxtr@warn@undefaction
      \let\glsxtr@warnonexistsordo\@glsxtr@warn@onexistsordo
      \glsxtr@addloclistfield
      \let\@@glsxtr@recordcounter\@glsxtr@op@recordcounter
      \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
      \undef\glsxtrsetaliasnoindex
    }%
  \fi
 }
\newcommand*{\@glsxtr@docdefval}{0}
\newcommand*{\if@glsxtrdocdef}{\ifnum\@glsxtr@docdefval>0 }
\newcommand*{\@glsxtrdocdeftrue}{\def\@glsxtr@docdefval{1}}
\newcommand*{\@glsxtrdocdeffalse}{\def\@glsxtr@docdefval{0}}
\define@choicekey{glossaries-extra.sty}{docdef}
 [\@glsxtr@docdefsetting\@glsxtr@docdefval]%
 {false,true,restricted,atom}[true]%
{%
  \ifnum\@glsxtr@docdefval>1\relax
    \renewcommand*{\@glsdoifexistsorwarn}{\glsdoifexists}%
  \else
    \renewcommand*{\@glsdoifexistsorwarn}{\glsdoifexistsorwarn}%
  \fi
}
\newcommand*{\if@glsxtrdocdefrestricted}{\ifnum\@glsxtr@docdefval>1 }
\newcommand*{\@glsdoifexistsorwarn}{\glsdoifexistsorwarn}
\define@boolkey{glossaries-extra.sty}[@glsxtr]{indexcrossrefs}[true]{%
```

```
\if@glsxtrindexcrossrefs
\else
 \renewcommand*{\@glsxtr@autoindexcrossrefs}{}%
\fi
}
\@@glsxtrindexcrossrefsfalse
\newcommand*{\@glsxtr@autoindexcrossrefs}{\@glsxtrindexcrossrefstrue}
\define@boolkey{glossaries-extra.sty}[@glsxtr@]{autoseeindex}[true]{%
}
\@glsxtr@autoseeindextrue
\define@boolkey{glossaries-extra.sty}[@glsxtr@]{equations}[true]{%
}
\@glsxtr@equationsfalse
\let\glsxtr@float\@float
\let\glsxtr@dblfloat\@dblfloat
\define@boolkey{glossaries-extra.sty}[@glsxtr@]{floats}[true]{%
  \if@glsxtr@floats
   \renewcommand*{\@float}[1]{\renewcommand{\glscounter}{##1}\glsxtr@float{##1}}%
   \renewcommand*{\@dblfloat}[1]{\renewcommand{\glscounter}{##1}\glsxtr@dblfloat{##1}}%
  \else
   \let\@float\glsxtr@float
   \let\@dblfloat\glsxtr@dblfloat
  \fi
}
\@glsxtr@floatsfalse
\newcommand*{\GlossariesExtraWarning}[1]{\PackageWarning{glossaries-extra}{#1}}
\newcommand*{\GlossariesExtraWarningNoLine}[1]{%
 \PackageWarningNoLine{glossaries-extra}{#1}}
\@glsxtr@declareoption{nowarn}{%
  \let\GlossariesExtraWarning\@gobble
  \let\GlossariesExtraWarningNoLine\@gobble
  \glsxtr@dooption{nowarn}%
}
\newcommand*{\@glsxtr@defpostpunc}{}
\@glsxtr@declareoption{postdot}{%
  \glsxtr@dooption{nopostdot=false}%
  \renewcommand*{\@glsxtr@defpostpunc}{%
    \renewcommand*{\glspostdescription}{%
      \ifglsnopostdot\else.\spacefactor\sfcode`\. \fi}%
  }%
}

\define@choicekey{glossaries-extra.sty}{nopostdot}{true,false}[true]{%
  \glsxtr@dooption{nopostdot=#1}%
  \renewcommand*{\@glsxtr@defpostpunc}{%
    \renewcommand*{\glspostdescription}{%
      \ifglsnopostdot\else.\spacefactor\sfcode`\. \fi}%
  }%
}
\define@key{glossaries-extra.sty}{postpunc}{%
```

```
    \glsxtr@dooption{nopostdot=false}%
    \ifstrequal{#1}{dot}%
    {%
      \renewcommand*{\@glsxtr@defpostpunc}{%
        \renewcommand*{\glspostdescription}{.\spacefactor\sfcode`\. }%
      }%
    }%
    {%
      \ifstrequal{#1}{comma}%
      {%
        \renewcommand*{\@glsxtr@defpostpunc}{%
          \renewcommand*{\glspostdescription}{,}%
        }%
      }%
      {%
        \ifstrequal{#1}{none}%
        {%
          \glsxtr@dooption{nopostdot=true}%
          \renewcommand*{\@glsxtr@defpostpunc}{%
            \renewcommand*{\glspostdescription}{}%
          }%
        }%
        {%
          \renewcommand*{\@glsxtr@defpostpunc}{%
            \renewcommand*{\glspostdescription}{#1}%
          }%
        }%
      }%
    }%
  }%
}
\newcommand*{\glsxtrabbrvtype}{\glsdefaulttype}
\newcommand*{\@glsxtr@abbreviationsdef}{}

\newcommand*{\@glsxtr@doabbreviationsdef}{%
  \@ifpackageloaded{babel}%
  {\providecommand{\abbreviationsname}{\acronymname}}%
  {\providecommand{\abbreviationsname}{Abbreviations}}%
  \newglossary[glg-abr]{abbreviations}{gls-abr}{glo-abr}{\abbreviationsname}%
  \renewcommand*{\glsxtrabbrvtype}{abbreviations}%
  \newcommand*{\printabbreviations}[1][]{%
    \printglossary[type=\glsxtrabbrvtype,##1]%
  }%
  \disable@keys{glossaries-extra.sty}{abbreviations}%
  \ifglsacronym
  \else
    \renewcommand*{\acronymtype}{\glsxtrabbrvtype}%
  \fi
}%
\@glsxtr@declareoption{abbreviations}{%
  \let\@glsxtr@abbreviationsdef\@glsxtr@doabbreviationsdef
```

```
}
\newcommand*{\GlsXtrDefineAbbreviationShortcuts}{%
  \newcommand*{\ab}{\cgls}%
  \newcommand*{\abp}{\cglspl}%
  \newcommand*{\as}{\glsxtrshort}%
  \newcommand*{\asp}{\glsxtrshortpl}%
  \newcommand*{\al}{\glsxtrlong}%
  \newcommand*{\alp}{\glsxtrlongpl}%
  \newcommand*{\af}{\glsxtrfull}%
  \newcommand*{\afp}{\glsxtrfullpl}%
  \newcommand*{\Ab}{\cGls}%
  \newcommand*{\Abp}{\cGlspl}%
  \newcommand*{\As}{\Glsxtrshort}%
  \newcommand*{\Asp}{\Glsxtrshortpl}%
  \newcommand*{\Al}{\Glsxtrlong}%
  \newcommand*{\Alp}{\Glsxtrlongpl}%
  \newcommand*{\Af}{\Glsxtrfull}%
  \newcommand*{\Afp}{\Glsxtrfullpl}%
  \newcommand*{\AB}{\cGLS}%
  \newcommand*{\ABP}{\cGLSpl}%
  \newcommand*{\AS}{\GLSxtrshort}%
  \newcommand*{\ASP}{\GLSxtrshortpl}%
  \newcommand*{\AL}{\GLSxtrlong}%
  \newcommand*{\ALP}{\GLSxtrlongpl}%
  \newcommand*{\AF}{\GLSxtrfull}%
  \newcommand*{\AFP}{\GLSxtrfullpl}%
  \providecommand*{\newabbr}{\newabbreviation}%
  \let\GlsXtrDefineAbbreviationShortcuts\relax
}
\newcommand*{\GlsXtrDefineAcShortcuts}{%
  \newcommand*{\ac}{\cgls}%
  \newcommand*{\acp}{\cglspl}%
  \newcommand*{\acs}{\glsxtrshort}%
  \newcommand*{\acsp}{\glsxtrshortpl}%
  \newcommand*{\acl}{\glsxtrlong}%
  \newcommand*{\aclp}{\glsxtrlongpl}%
  \newcommand*{\acf}{\glsxtrfull}%
  \newcommand*{\acfp}{\glsxtrfullpl}%
  \newcommand*{\Ac}{\cGls}%
  \newcommand*{\Acp}{\cGlspl}%
  \newcommand*{\Acs}{\Glsxtrshort}%
  \newcommand*{\Acsp}{\Glsxtrshortpl}%
  \newcommand*{\Acl}{\Glsxtrlong}%
  \newcommand*{\Aclp}{\Glsxtrlongpl}%
  \newcommand*{\Acf}{\Glsxtrfull}%
  \newcommand*{\Acfp}{\Glsxtrfullpl}%
  \newcommand*{\AC}{\cGLS}%
  \newcommand*{\ACP}{\cGLSpl}%
  \newcommand*{\ACS}{\GLSxtrshort}%
  \newcommand*{\ACSP}{\GLSxtrshortpl}%
```

```
    \newcommand*{\ACL}{\GLSxtrlong}%
    \newcommand*{\ACLP}{\GLSxtrlongpl}%
    \newcommand*{\ACF}{\GLSxtrfull}%
    \newcommand*{\ACFP}{\GLSxtrfullpl}%
    \providecommand*{\newabbr}{\newabbreviation}%
    \let\GlsXtrDefineAcShortcuts\relax
}
\newcommand*{\GlsXtrDefineOtherShortcuts}{%
    \newcommand*{\newentry}{\newglossaryentry}%
    \ifdef\printsymbols
    {%
        \newcommand*{\newsym}{\glsxtrnewsymbol}%
    }{}%
    \ifdef\printnumbers
    {%
        \newcommand*{\newnum}{\glsxtrnewnumber}%
    }{}%
    \let\GlsXtrDefineOtherShortcuts\relax
}
\newcommand*{\@glsxtr@setupshortcuts}{}
\newcommand*{\@glsxtr@shortcutsval}{\ifglsacrshortcuts acro\else none\fi}%
\define@choicekey{glossaries-extra.sty}{shortcuts}%
 [\@glsxtr@shortcutsval\@glsxtr@shortcutsnr]%
 {acronyms,acro,abbreviations,abbr,other,all,true,ac,none,false}[true]{%
    \ifcase\@glsxtr@shortcutsnr\relax % acronyms
        \renewcommand*{\@glsxtr@setupshortcuts}{%
            \glsacrshortcutstrue
            \DefineAcronymSynonyms
        }%
    \or % acro
        \renewcommand*{\@glsxtr@setupshortcuts}{%
            \glsacrshortcutstrue
            \DefineAcronymSynonyms
        }%
    \or % abbreviations
        \renewcommand*{\@glsxtr@setupshortcuts}{%
            \GlsXtrDefineAbbreviationShortcuts
        }%
    \or % abbr
        \renewcommand*{\@glsxtr@setupshortcuts}{%
            \GlsXtrDefineAbbreviationShortcuts
        }%
    \or % other
        \renewcommand*{\@glsxtr@setupshortcuts}{%
            \GlsXtrDefineOtherShortcuts
        }%
    \or % all
        \renewcommand*{\@glsxtr@setupshortcuts}{%
            \glsacrshortcutstrue
            \GlsXtrDefineAcShortcuts
```

```
        \GlsXtrDefineAbbreviationShortcuts
        \GlsXtrDefineOtherShortcuts
      }%
    \or % true
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \glsacrshortcutstrue
        \GlsXtrDefineAcShortcuts
        \GlsXtrDefineAbbreviationShortcuts
        \GlsXtrDefineOtherShortcuts
      }%
    \or % ac
      \renewcommand*{\@glsxtr@setupshortcuts}{%
        \glsacrshortcutstrue
        \GlsXtrDefineAcShortcuts
      }%
    \else % none, false
      \renewcommand*{\@glsxtr@setupshortcuts}{}%
    \fi
 }
\newcommand*{\@glsxtr@doaccsupp}{}
\@glsxtr@declareoption{accsupp}{%
 \renewcommand*{\@glsxtr@doaccsupp}{\RequirePackage{glossaries-accsupp}}}
\newcommand*{\@glsxtr@doloadprefix}{}
\@glsxtr@declareoption{prefix}{%
 \renewcommand*{\@glsxtr@doloadprefix}{\RequirePackage{glossaries-prefix}}}
\newcommand{\glsxtrNoGlossaryWarning}[1]{%
  \GlossariesExtraWarning{Glossary '#1' is missing}%
  \@glsxtr@defaultnoglossarywarning{#1}%
}
\define@choicekey{glossaries-extra.sty}{nomissingglstext}
 [\@glsxtr@nomissingglstextval\@glsxtr@nomissingglstextnr]%
 {true,false}[true]{%
   \ifcase\@glsxtr@nomissingglstextnr\relax % true
     \renewcommand{\glsxtrNoGlossaryWarning}[1]{\null}%
   \else % false
     \renewcommand{\glsxtrNoGlossaryWarning}[1]{%
       \@glsxtr@defaultnoglossarywarning{#1}%
     }%
   \fi
 }
\newcommand*{\@glsxtr@redefstyles}{}
\define@key{glossaries-extra.sty}{stylemods}[default]{%
  \ifstrequal{#1}{default}%
  {%
    \renewcommand*{\@glsxtr@redefstyles}{%
      \RequirePackage{glossaries-extra-stylemods}}%
  }%
  {%
    \ifstrequal{#1}{all}%
    {%
```

```
      \renewcommand*{\@glsxtr@redefstyles}{%
        \PassOptionsToPackage{all}{glossaries-extra-stylemods}%
        \RequirePackage{glossaries-extra-stylemods}%
      }%
    }%
    {%
      \renewcommand*{\@glsxtr@redefstyles}{}%
      \@for\@glsxtr@tmp:=#1\do{%
        \IfFileExists{glossary-\@glsxtr@tmp.sty}%
        {%
          \eappto\@glsxtr@redefstyles{%
            \noexpand\RequirePackage{glossary-\@glsxtr@tmp}}%
        }%
        {%
          \PackageError{glossaries-extra}%
           {Glossaries style package 'glossary-\@glsxtr@tmp.sty'
            doesn't exist (did you mean to use the 'style' key?)}%
           {The list of values (#1) in the 'stylemods' key should
            match the glossary-xxx.sty files provided with
            glossaries.sty}%
        }%
      }%
      \appto\@glsxtr@redefstyles{\RequirePackage{glossaries-extra-stylemods}[=v1.48]}%
    }
  }%
}
\newcommand*{\@glsxtr@do@style}{}
\define@key{glossaries-extra.sty}{style}{%
 \renewcommand*{\@glsxtr@do@style}{%
   \setkeys{glossaries.sty}{style={#1}}%
   \setglossarystyle{#1}%
 }%
}
\newcommand*{\glsxtr@inc@wrglossaryctr}[1]{}
\newcommand*{\GlsXtrInternalLocationHyperlink}[3]{%
  \glsxtrhyperlink{#1#2#3}{#3}%
}
\newcommand*{\@glsxtr@wrglossary@locationhyperlink}[3]{%
  \pageref{wrglossary.#3}%
}
\@glsxtr@declareoption{indexcounter}{%
  \glsxtr@dooption{counter=wrglossary}%
  \ifundef\c@wrglossary
  {%
    \newcounter{wrglossary}%
    \renewcommand{\thewrglossary}{\arabic{wrglossary}}%
  }%
  {}%
  \renewcommand*{\glsxtr@inc@wrglossaryctr}[1]{%
    \ifdefstring\@gls@counter{wrglossary}%
```

```
      {%
        \refstepcounter{wrglossary}%
        \label{wrglossary.\thewrglossary}%
      }%
      {}%
    }%
    \renewcommand*{\GlsXtrInternalLocationHyperlink}[3]{%
      \ifdefstring\glsentrycounter{wrglossary}%
      {%
        \@glsxtr@wrglossary@locationhyperlink{##1}{##2}{##3}%
      }%
      {\glsxtrhyperlink{##1##2##3}{##3}}%
    }%
}
\newcommand*{\@glsxtrwrglossmark}{}
\newcommand*{\@@glsxtrwrglossmark}{}
\AtBeginDocument{\renewcommand*{\@@glsxtrwrglossmark}{\@glsxtrwrglossmark}}
\newcommand*{\glsxtrwrglossmark}{\ensuremath{\cdot}}
\newcommand\@glsxtr@doshowtarget[2]{#2}
\define@choicekey{glossaries-extra.sty}{debug}
 [\@glsxtr@debugval\@glsxtr@debugnr]%
 {true,false,showtargets,showwrgloss,all,showaccsupp}[true]{%
   \ifcase\@glsxtr@debugnr\relax % true
     \glsxtr@dooption{debug=true}%
     \renewcommand*{\@glsxtrwrglossmark}{}%
   \or % false
     \glsxtr@dooption{debug=false}%
     \renewcommand*{\@glsxtrwrglossmark}{}%
      \let\@glsxtr@doshowtarget\@secondoftwo
   \or % showtargets
     \glsxtr@dooption{debug=showtargets}%
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
   \or % showwrgloss
     \glsxtr@dooption{debug=true}%
     \renewcommand*{\@glsxtrwrglossmark}{\glsxtrwrglossmark}%
   \or % all
     \glsxtr@dooption{debug=showtargets,debug=showaccsupp}%
     \renewcommand*{\@glsxtrwrglossmark}{\glsxtrwrglossmark}%
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
   \or % showaccsupp
     \glsxtr@dooption{debug=showaccsupp}%
   \fi
 }
\newcommand*{\glsxtrshowtargetouter}{\glsshowtargetouter}
\newcommand*{\glsxtrshowtargetinner}[1]{\glsshowtargetinner{#1}}
\newcommand{\@glsxtrshowtargetleft}[2]{\@glsshowtarget{#1}#2\@glsxtrshowtargetmark}%
\newcommand{\@glsxtrshowtargetright}[2]{\@glsxtrshowtargetmark#2\@glsshowtarget{#1}}%
\newcommand{\@glsxtrshowtargetmark}{}%
\define@choicekey{glossaries-extra.sty}{showtargets}
 [\@glsxtr@showtargetsval\@glsxtr@showtargetsnr]%
```

834

```
  {left,right,innerleft,innerright,annoteleft,annoteright}%
  {%
    \glsxtr@dooption{debug=showtargets}%
    \ifcase\@glsxtr@showtargetsnr\relax
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
      \def\glsxtrshowtargetouter{\glsshowtargetouter}%
      \def\glsxtrshowtargetinner{\glsshowtargetinner}%
      \let\@glsxtrshowtargetmark\empty
    \or
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetright}%
      \def\glsxtrshowtargetouter{\glsshowtargetouter}%
      \def\glsxtrshowtargetinner{\glsshowtargetinner}%
      \let\@glsxtrshowtargetmark\empty
    \or
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
      \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
      \def\glsxtrshowtargetinner{\glsshowtargetinnersymleft}%
      \let\@glsxtrshowtargetmark\empty
    \or
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetright}%
      \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
      \def\glsxtrshowtargetinner{\glsshowtargetinnersymright}%
      \let\@glsxtrshowtargetmark\empty
    \or
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
      \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
      \def\glsxtrshowtargetinner{\glsshowtargetinnersymleft}%
      \def\@glsxtrshowtargetmark{\@glsshowtargetmarkfmt\glsxtrshowtargetsymbolright}%
    \or
      \def\@glsxtr@doshowtarget{\@glsxtrshowtargetright}%
      \def\glsxtrshowtargetouter{\glsxtrshowtargetinner}%
      \def\glsxtrshowtargetinner{\glsshowtargetinnersymright}%
      \def\@glsxtrshowtargetmark{\@glsshowtargetmarkfmt\glsxtrshowtargetsymbolleft}%
    \fi
 }
\DeclareOptionX*{%
 \expandafter\glsxtr@dooption\expandafter{\CurrentOption}}
\ProcessOptionsX
\RequirePackage{glossaries}
\@glsxtr@doaccsupp
\@glsxtr@doloadprefix
\@glsxtr@defpostpunc
\def\glsdoshowtarget{\@glsxtr@doshowtarget}
\newcommand{\glsxtrshowtargetsymbolright}{{\tiny$\triangleleft$}}%
\newcommand{\glsxtrshowtargetsymbolleft}{{\tiny$\triangleright$}}%
\providecommand*{\glsshowtargetinner}[1]{{\glsshowtargetfont [#1]}}
\providecommand*{\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*{\glsshowtargetinnersymleft}[1]{%
 \glsshowtargetinner{#1}\allowbreak\glsxtrshowtargetsymbolleft}
\newcommand*{\glsshowtargetinnersymright}[1]{%
```

```
    \glsxtrshowtargetsymbolright\allowbreak\glsshowtargetinner{#1}}
\providecommand*{\glsshowtargetouter}[1]{%
 \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\providecommand*{\@glsshowtarget}[1]{}
\def\glsshowtarget#1{%
 \glsxtrtitleorpdforheading
 {%
    \ifmmode
      \nfss@text{\glsxtrshowtargetinner{#1}}%
    \else
      \ifinner
        \glsxtrshowtargetinner{#1}%
      \else
        \glsxtrshowtargetouter{#1}%
      \fi
    \fi
 }%
 {[#1]}%
 {{\protect\glsshowtargetinner{#1}}}%
}
\newcommand*{\@glsshowtargetmarkfmt}[1]{%
 \glsxtrtitleorpdforheading
 {%
    \ifmmode \nfss@text{#1}\else #1\fi
 }%
 {}%
 {\ifmmode \nfss@text{#1}\else #1\fi}%
}
\let\@glsxtr@org@doseeglossary\@do@seeglossary
\newcommand*{\@glsxtr@doseeglossary}[2]{%
  \glsdoifexists{#1}%
  {%
    \@@glsxtrwrglossmark
    \@glsxtr@org@doseeglossary{#1}{#2}%
  }%
}
\newcommand*{\@glsxtr@dosee@alsoindex@glossary}[2]{%
  \@glsxtr@recordsee{#1}{#2}%
  \@glsxtr@doseeglossary{#1}{#2}%
}
\let\@glsxtr@org@gloautosee\@glo@autosee
\if@glsxtr@autoseeindex
\else
  \ifdef\@glsxtr@org@gloautosee
  {}%
  {\PackageError{glossaries-extra}{`autoseeindex=false' package
   option requires at least v4.30 of glossaries.sty}%
   {You need to update the glossaries.sty package}%
  }
\fi
```

```
\ifdef\@glo@autosee
{%
  \renewcommand*{\@glo@autosee}{%
    \if@glsxtr@autoseeindex\@glsxtr@org@gloautosee\fi}%
}%
{}
\renewcommand*{\gls@checkseeallowed}{%
 \if@glsxtr@autoseeindex\@gls@see@noindex\fi
}
\@glsxtr@abbreviationsdef
\let\@glsxtr@abbreviationsdef\relax
\@glsxtr@setupshortcuts
\@glsxtr@redef@forglsentries
\renewcommand{\glsxtr@dooption}[1]{\setupglossaries{#1}}%
\disable@keys{glossaries-extra.sty}{accsupp}
\newcommand*{\glossariesextrasetup}[1]{%
  \let\glsxtr@setup@record\relax
  \let\@glsxtr@setupshortcuts\relax
  \let\@glsxtr@redef@forglsentries\relax
  \let\@glsxtr@doloadprefix\relax
  \setkeys{glossaries-extra.sty}{#1}%
  \@glsxtr@abbreviationsdef
  \let\@glsxtr@abbreviationsdef\relax
  \@glsxtr@setupshortcuts
  \glsxtr@setup@record
  \@glsxtr@redef@forglsentries
  \@glsxtr@doloadprefix
}
\let\glsxtr@org@@do@wrglossary\@@do@wrglossary
\newcommand*{\glsxtr@@do@wrglossary}[1]{%
 \@@glsxtrwrglossmark
 \glsxtr@inc@wrglossaryctr{#1}%
 \glsxtr@org@@do@wrglossary{#1}%
}
\let\glsxtr@saveentrycounter\@gls@saveentrycounter
\let\@gls@saveentrycounter\glsxtr@indexonly@saveentrycounter
\renewcommand*\@gls@getcounterprefix[2]{%
  \protected@edef\@gls@thisloc{#1}\protected@edef\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
```

```
    \ifx\@glo@counterprefix\@empty
      \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
      \else
        \GlossariesExtraWarning{Hyper target '#2' can't be formed by
          prefixing^^Jlocation '#1'. You need to modify the
          definition of \string\theH\@gls@counter^^Jotherwise you
          will get the warning: "'name{\@gls@counter.#1}' has been^^J
          referenced but does not exist"%
          \ifx\@glsxtr@record@setting\@glsxtr@record@setting@only
          . You may want to consider using record=nameref instead%
          \fi}%
      \fi
    \fi
  \fi
}
\newcommand*{\@glsxtrdialecthook}{}
\glsxtr@setup@record
\AtBeginDocument{%
  \disable@keys{glossaries-extra.sty}{abbreviations,docdef,record}%
  \def\@glsxtrundeftag{\glsxtrundeftag}%
}
\newcommand*{\GlsXtrIfUnusedOrUndefined}[3]{%
  \ifglsentryexists{#1}%
  {\ifbool{glo@\glsdetoklabel{#1}@flag}{#3}{#2}}%
  {#2}%
}
\ifdef\s@ifglossaryexists
{}
{
  \renewcommand{\ifglossaryexists}{%
    \@ifstar\s@ifglossaryexists\@ifglossaryexists
  }
  \newcommand{\@ifglossaryexists}[3]{%
    \ifcsundef{@glotype@#1@out}{#3}{#2}%
  }
  \newcommand{\s@ifglossaryexists}[3]{%
    \ifcsundef{glolist@#1}{#3}{#2}%
  }
}
\newcommand{\glsxtrifemptyglossary}[3]{%
  \ifcsdef{glolist@#1}%
  {%
    \ifcsstring{glolist@#1}{,}{#2}{#3}%
  }%
  {%
    \glsxtrundefaction{Glossary type '#1' doesn't exist}{}%
    #2%
  }%
}
\newcommand*{\glsxtrifkeydefined}[3]{%
```

```
    \key@ifundefined{glossentry}{#1}{#3}{#2}%
}
\newcommand*{\glsxtrprovidestoragekey}{%
  \@ifstar\@sglsxtr@provide@storagekey\@glsxtr@provide@storagekey
}
\newcommand*{\@glsxtr@provide@storagekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{,{#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \ifblank{#3}
    {}%
    {%
        \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
    }%
  }%
  {%
    \ifblank{#3}
    {}%
    {%
      \providecommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
    }%
  }%
}
\newcommand*{\s@glsxtr@provide@storagekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
     {\csname gls@assign@#1@field\endcsname}[2]{%
        \@@gls@expand@field{##1}{#1}{##2}%
     }%
  }%
  {}%
  \@glsxtr@provide@addstoragekey{#1}%
}
\newcommand{\GlsXtrFmtField}{useri}
\newcommand{\GlsXtrFmtDefaultOptions}{noindex}
\newrobustcmd*{\glsxtrfmt}{\@ifstar\s@glsxtrfmt\@glsxtrfmt}
\newcommand*{\@glsxtrfmt}[3][]{\@@glsxtrfmt{#1}{#2}{#3}{}}
\newcommand*{\s@glsxtrfmt}[3][]{%
 \new@ifnextchar[{\s@@glsxtrfmt{#1}{#2}{#3}}%
  {\@@glsxtrfmt{#1}{#2}{#3}{}}%
}
\def\s@@glsxtrfmt#1#2#3[#4]{\@@glsxtrfmt{#1}{#2}{#3}{#4}}
\newcommand*{\@@glsxtrfmt}[4]{%
```

```
   \begingroup
     \def\glslabel{#2}%
     \glsdoifexistsordo{#2}%
     {%
       \ifglshasfield{\GlsXtrFmtField}{#2}%
       {%
         \let\do@gls@link@checkfirsthyper\relax
         \expandafter\@gls@link\expandafter[\GlsXtrFmtDefaultOptions,#1]{#2}%
           {\glsxtrfmtdisplay{\glscurrentfieldvalue}{#3}{#4}}%
       }%
       {\glsxtrfmtdisplay{@firstofone}{#3}{#4}}%
     }%
     {%
       \begingroup
         \@gls@setdefault@glslink@opts
         \setkeys{glslink}{\GlsXtrFmtDefaultOptions,#1}%
         \ifKV@glslink@noindex\else\glsadd{#2}\fi
       \endgroup
       \glsxtrfmtdisplay{@firstofone}{#3}{#4}%
     }%
 \endgroup
}
\newcommand{\glsxtrfmtdisplay}[3]{\csuse{#1}{#2}#3}
\ifdef\texorpdfstring
{
  \newcommand*{\glsxtrentryfmt}[2]{%
    \texorpdfstring{\@glsxtrentryfmt{#1}{#2}}{\glsxtrpdfentryfmt{#1}{#2}}%
  }
}
{
  \newcommand*{\glsxtrentryfmt}{\@glsxtrentryfmt}
}
\newcommand*{\glsxtrpdfentryfmt}[2]{#2}
\newrobustcmd*{\@glsxtrentryfmt}[2]{%
 {%
    \protected@edef\glslabel{#1}%
    \glsdoifexistsordo{#1}%
    {%
      \ifglshasfield{\GlsXtrFmtField}{#1}%
      {%
        \csuse{\glscurrentfieldvalue}{#2}%
      }%
      {#2}%
    }%
    {#2}%
 }%
}
\newcommand*{\glsxtrfieldlistadd}[3]{%
  \listcsadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
```

```
\newcommand*{\glsxtrfieldlistgadd}[3]{%
  \listcsgadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
\newcommand*{\glsxtrfieldlisteadd}[3]{%
  \listcseadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
\newcommand*{\glsxtrfieldlistxadd}[3]{%
  \listcsxadd{glo@\glsdetoklabel{#1}@#2}{#3}%
}
\newcommand*{\glsxtrfielddolistloop}[2]{%
  \dolistcsloop{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsxtrfieldforlistloop}[3]{%
  \forlistcsloop{#3}{glo@\glsdetoklabel{#1}@#2}%
}
\newrobustcmd*{\glsxtrfieldformatlist}[2]{%
 \begingroup
  \def\@dtl@formatlist@itemsep{}%
  \def\@dtl@formatlist@lastitem{}%
  \def\@dtl@formatlist@prelastitem{}%
  \def\@dtl@formatlist@prelastitemsep{}%
  \forlistcsloop{\@dtl@formatlist@handler}{glo@\glsdetoklabel{#1}@#2}%
  \@dtl@formatlist@prelastitem\@dtl@formatlist@lastitem
 \endgroup
}
\newcommand*{\glsxtrfieldifinlist}[5]{%
  \ifinlistcs{#3}{glo@\glsdetoklabel{#1}@#2}{#4}{#5}%
}
\newcommand*{\glsxtrfieldxifinlist}[5]{%
  \xifinlistcs{#3}{glo@\glsdetoklabel{#1}@#2}{#4}{#5}%
}
\newcommand*{\glsxtrforcsvfield}{%
  \@ifstar\s@glsxtrforcsvfield\@glsxtrforcsvfield
}
\newcommand*{\@glsxtrforcsvfield}[3]{%
 \@glsxtrifhasfield{#2}{#1}%
 {%
   \let\glsxtrendfor\@endfortrue
   \@for\@glsxtr@label:=\glscurrentfieldvalue\do
   {\expandafter#3\expandafter{\@glsxtr@label}}}%
 {}%
}
\newcommand*{\s@glsxtrforcsvfield}[3]{%
 \s@glsxtrifhasfield{#2}{#1}%
 {%
   \let\glsxtrendfor\@endfortrue
   \@for\@glsxtr@label:=\glscurrentfieldvalue\do
   {\expandafter#3\expandafter{\@glsxtr@label}}}%
 {}%
}
```

```
\newrobustcmd*{\glsxtrfieldformatcsvlist}[2]{%
 \@glsxtrifhasfield{#2}{#1}%
 {\@dtlformatlist\glscurrentfieldvalue}%
 {}%
}
\newcommand*{\GlsXtrIfFieldValueInCsvList}{%
  \@ifstar\s@GlsXtrIfFieldValueInCsvList\@GlsXtrIfFieldValueInCsvList
}
\newcommand*{\@GlsXtrIfFieldValueInCsvList}[5]{%
 \@glsxtrifhasfield{#2}{#1}%
 {%
    \expandafter\DTLifinlist\expandafter{\glscurrentfieldvalue}%
    {#3}{#4}{#5}%
 }%
 {#5}%
}
\newcommand*{\s@GlsXtrIfFieldValueInCsvList}[5]{%
 \s@glsxtrifhasfield{#2}{#1}%
 {%
    \expandafter\DTLifinlist\expandafter{\glscurrentfieldvalue}%
    {#3}{#4}{#5}%
 }%
 {#5}%
}
\newcommand*{\GlsXtrIfValueInFieldCsvList}{%
  \@ifstar\s@GlsXtrIfValueInFieldCsvList\@GlsXtrIfValueInFieldCsvList
}
\newcommand*{\@GlsXtrIfValueInFieldCsvList}[5]{%
 \@glsxtrifhasfield{#2}{#1}%
 {%
    \DTLifinlist{#3}{\glscurrentfieldvalue}{#4}{#5}%
 }%
 {#5}%
}
\newcommand*{\s@GlsXtrIfValueInFieldCsvList}[5]{%
 \s@glsxtrifhasfield{#2}{#1}%
 {%
    \DTLifinlist{#3}{\glscurrentfieldvalue}{#4}{#5}%
 }%
 {#5}%
}
\newcommand*{\xGlsXtrIfValueInFieldCsvList}{%
  \@ifstar\s@xGlsXtrIfValueInFieldCsvList\@xGlsXtrIfValueInFieldCsvList
}
\newcommand*{\@xGlsXtrIfValueInFieldCsvList}[5]{%
 \@glsxtrifhasfield{#2}{#1}%
 {%
    \protected@edef\@gls@tmp{#3}%
    \expandafter\DTLifinlist\expandafter{\@gls@tmp}{\glscurrentfieldvalue}{#4}{#5}%
 }%
```

```
 {#5}%
}
\newcommand*{\s@xGlsXtrIfValueInFieldCsvList}[5]{%
 \s@glsxtrifhasfield{#2}{#1}%
 {%
    \protected@edef\@gls@tmp{#3}%
    \expandafter\DTLifinlist\expandafter{\@gls@tmp}{\glscurrentfieldvalue}{#4}{#5}%
 }%
 {#5}%
}
\newrobustcmd{\glsxtrifhasfield}{%
  \@ifstar{\s@glsxtrifhasfield}{\@glsxtrifhasfield}%
}
\newcommand{\@glsxtrifhasfield}[4]{%
  {\s@glsxtrifhasfield{#1}{#2}{#3}{#4}}%
}
\newcommand{\s@glsxtrifhasfield}[4]{%
  \letcs{\glscurrentfieldvalue}{glo@\glsdetoklabel{#2}@#1}%
  \ifundef\glscurrentfieldvalue
  {#4}%
  {%
   \ifdefempty\glscurrentfieldvalue{#4}{#3}%
  }%
}
\newcommand{\GlsXtrIfFieldNonZero}{%
  \@ifstar\s@GlsXtrIfFieldNonZero\@GlsXtrIfFieldNonZero
}
\newcommand{\@GlsXtrIfFieldNonZero}[4]{%
  \@GlsXtrIfFieldCmpNum{#1}{#2}{=}{0}{#4}{#3}%
}
\newcommand{\s@GlsXtrIfFieldNonZero}[4]{%
  \s@GlsXtrIfFieldCmpNum{#1}{#2}{=}{0}{#4}{#3}%
}
\newcommand{\GlsXtrIfFieldEqNum}{%
  \@ifstar\s@GlsXtrIfFieldEqNum\@GlsXtrIfFieldEqNum
}
\newcommand{\@GlsXtrIfFieldEqNum}[5]{%
  \@GlsXtrIfFieldCmpNum{#1}{#2}{=}{#3}{#4}{#5}%
}
\newcommand{\s@GlsXtrIfFieldEqNum}[5]{%
  \s@GlsXtrIfFieldCmpNum{#1}{#2}{=}{#3}{#4}{#5}%
}
\newcommand{\GlsXtrIfFieldCmpNum}{%
  \@ifstar\s@GlsXtrIfFieldCmpNum\@GlsXtrIfFieldCmpNum
}
\newcommand{\@GlsXtrIfFieldCmpNum}[6]{%
  {%
    \letcs{\glscurrentfieldvalue}{glo@\glsdetoklabel{#2}@#1}%
    \ifundef\glscurrentfieldvalue
    {\def\glscurrentfieldvalue{0}}%
```

```
    {%
     \ifdefempty\glscurrentfieldvalue
     {\def\glscurrentfieldvalue{0}}%
     {}%
    }%
    \ifnum\glscurrentfieldvalue#3#4\relax #5\else #6\fi
  }%
}
\newcommand{\s@GlsXtrIfFieldCmpNum}[6]{%
  \letcs{\glscurrentfieldvalue}{glo@\glsdetoklabel{#2}@#1}%
  \ifundef\glscurrentfieldvalue
  {\def\glscurrentfieldvalue{0}}%
  {%
   \ifdefempty\glscurrentfieldvalue
   {\def\glscurrentfieldvalue{0}}%
   {}%
  }%
  \ifnum\glscurrentfieldvalue#3#4\relax #5\else #6\fi
}
\newcommand{\GlsXtrIfFieldUndef}[2]{%
 \ifcsundef{glo@\glsdetoklabel{#2}@#1}%
}
\newcommand*{\glsxtrusefield}[2]{%
  \@gls@entry@field{#1}{#2}%
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsxtrusefield}[2]{%
    \texorpdfstring
     {\@Gls@entry@field{#1}{#2}}
     {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*{\Glsxtrusefield}[2]{%
    \@Gls@entry@field{#1}{#2}%
  }
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSxtrusefield}[2]{%
    \texorpdfstring
     {\glsdoifexists{#1}{\mfirstucMakeUppercase{\@gls@entry@field{#1}{#2}}}}%
     {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*{\GLSxtrusefield}[2]{%
    \glsdoifexists{#1}{\mfirstucMakeUppercase{\@gls@entry@field{#1}{#2}}}%
  }
```

```
}
\newcommand*{\glsxtrentryparentname}[1]{%
  \ifcsdef{glo@\glsdetoklabel{#1}@parent}%
  {\csuse{glo@\csuse{glo@\glsdetoklabel{#1}@parent}@name}}%
  {}%
}
\newcommand*{\glsxtrdeffield}[2]{\csdef{glo@\glsdetoklabel{#1}@#2}}
\newcommand*{\glsxtredeffield}[2]{\protected@csedef{glo@\glsdetoklabel{#1}@#2}}
\newcommand*{\glsxtrapptocsvfield}[3]{%
 \ifcsdef{glo@\glsdetoklabel{#1}@#2}%
 {\csappto{glo@\glsdetoklabel{#1}@#2}{,#3}}%
 {\csdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
\newcommand*{\glsxtrsetfieldifexists}[3]{\glsdoifexists{#1}{#3}}
\newrobustcmd*{\GlsXtrSetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\csdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\GlstrLetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\cslet{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\csGlsXtrLetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\csletcs{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\GlsXtrLetFieldToField}[4]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\csletcs{glo@\glsdetoklabel{#1}@#2}{glo@\glsdetoklabel{#3}@#4}}%
}
\newrobustcmd*{\gGlsXtrSetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\csgdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\xGlsXtrSetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\protected@csxdef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\eGlsXtrSetField}[3]{%
  \glsxtrsetfieldifexists{#1}{#2}%
  {\protected@csedef{glo@\glsdetoklabel{#1}@#2}{#3}}%
}
\newcommand*{\GlsXtrIfFieldEqStr}{%
  \@ifstar\s@GlsXtrIfFieldEqStr\@GlsXtrIfFieldEqStr
}
\newrobustcmd*{\@GlsXtrIfFieldEqStr}[5]{%
  \@glsxtrifhasfield{#1}{#2}%
  {%
    \ifdefstring{\glscurrentfieldvalue}{#3}{#4}{#5}%
  }%
```

```
    {#5}%
}
\newrobustcmd*{\s@GlsXtrIfFieldEqStr}[5]{%
  \s@glsxtrifhasfield{#1}{#2}%
  {%
    \ifdefstring{\glscurrentfieldvalue}{#3}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\GlsXtrIfFieldEqXpStr}{%
  \@ifstar\s@GlsXtrIfFieldEqXpStr\@GlsXtrIfFieldEqXpStr
}
\newrobustcmd*{\@GlsXtrIfFieldEqXpStr}[5]{%
  \@glsxtrifhasfield{#1}{#2}%
  {%
    \protected@edef\@gls@tmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
  }%
  {#5}%
}
\newrobustcmd*{\s@GlsXtrIfFieldEqXpStr}[5]{%
  \s@glsxtrifhasfield{#1}{#2}%
  {%
    \protected@edef\@gls@tmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\GlsXtrIfXpFieldEqXpStr}{%
  \@ifstar\s@GlsXtrIfXpFieldEqXpStr\@GlsXtrIfXpFieldEqXpStr
}
\newrobustcmd*{\@GlsXtrIfXpFieldEqXpStr}[5]{%
  \@glsxtrifhasfield{#1}{#2}%
  {%
    \protected@edef\@gls@tmp{\glscurrentfieldvalue}%
    \let\glscurrentfieldvalue\@gls@tmp
    \protected@edef\@gls@tmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
  }%
  {#5}%
}
\newrobustcmd*{\s@GlsXtrIfXpFieldEqXpStr}[5]{%
  \s@glsxtrifhasfield{#1}{#2}%
  {%
    \protected@edef\@gls@tmp{\glscurrentfieldvalue}%
    \let\glscurrentfieldvalue\@gls@tmp
    \protected@edef\@gls@tmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@gls@tmp}{#4}{#5}%
  }%
  {#5}%
```

```
}
\ifdef\foreignlanguage
{
  \ifdef\GetTrackedDialectFromLanguageTag
  {
    \newcommand{\GlsXtrForeignText}[2]{%
      \let\@glsxtr@org@currentfieldvalue\glscurrentfieldvalue
      \glsxtrifhasfield{\GlsXtrForeignTextField}{#1}%
      {%
        \expandafter\GetTrackedDialectFromLanguageTag\expandafter
          {\glscurrentfieldvalue}{\@glsxtr@dialect}%
        \let\@glsxtr@locale\glscurrentfieldvalue
        \let\glscurrentfieldvalue\@glsxtr@org@currentfieldvalue
        \ifdefempty\@glsxtr@dialect
        {%
          \ifundef\TrackedDialectClosestSubMatch
          {%
            \GlossariesExtraWarning{Can't obtain dialect label
              (tracklang v1.3.6+ required)}%
          }%
          {\let\@glsxtr@dialect\TrackedDialectClosestSubMatch}%
        }%
        {}%
        \ifdefempty\@glsxtr@dialect
        {%
        }%
        {%
          \ifcsundef{captions\@glsxtr@dialect}{}%
          {%
            \IfTrackedDialectHasMapping{\@glsxtr@dialect}%
            {%
              \edef\@glsxtr@dialect{%
                \GetTrackedDialectToMapping{\@glsxtr@dialect}}%
              \ifcsundef{captions\@glsxtr@dialect}{}%
              {%
                \ifcsundef{captions\@tracklang@lang}{}%
                {%
                  \let\@glsxtr@dialect\@tracklang@lang
                }%
              }%
            }%
            {%
              \ifcsundef{captions\@tracklang@lang}{}%
              {%
                \let\@glsxtr@dialect\@tracklang@lang
              }%
            }%
          }%
        }%
        \ifdefempty\@glsxtr@dialect
```

```
        {%
          \GlsXtrUnknownDialectWarning{\@glsxtr@locale}{\@tracklang@lang}%
          #2%
        }%
        {\foreignlanguage{\@glsxtr@dialect}{#2}}%
      }%
      {#2}% key not set
    }
  }
  {
    \newcommand{\GlsXtrForeignText}[2]{%
      \GlossariesExtraWarning{Can't encapsulate foreign text:
        tracklang v1.3.6+ required}%
      #2%
    }
  }
}
{
  \newcommand{\GlsXtrForeignText}[2]{#2}
}
\newcommand*{\GlsXtrForeignTextField}{userii}
\newcommand*{\GlsXtrUnknownDialectWarning}[2]{%
  \GlossariesExtraWarning{Can't determine valid dialect label
   for locale '#1' (root language: #2)}%
}
\ifdef\GlsEntryCounterLabelPrefix
{%
  \newcommand*{\glsxtrpageref}[1]{%
    \ifglsentrycounter
      \pageref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \ifglssubentrycounter
        \pageref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
      \else
        \gls{#1}%
      \fi
    \fi
  }
}%
{%
  \newcommand*{\glsxtrpageref}[1]{%
    \ifglsentrycounter
      \pageref{glsentry-\glsdetoklabel{#1}}%
    \else
      \ifglssubentrycounter
        \pageref{glsentry-\glsdetoklabel{#1}}%
      \else
        \gls{#1}%
      \fi
    \fi
```

```
    }
}%
\newcommand{\apptoglossarypreamble}[2][\glsdefaulttype]{%
  \ifcsdef{glolist@#1}%
  {%
   \ifcsundef{@glossarypreamble@#1}%
   {\csdef{@glossarypreamble@#1}{}}%
   {}%
   \csappto{@glossarypreamble@#1}{#2}%
  }%
  {%
    \GlossariesExtraWarning{Glossary '#1' is not defined}%
  }%
}
\newcommand{\preglossarypreamble}[2][\glsdefaulttype]{%
  \ifcsdef{glolist@#1}%
  {%
   \ifcsundef{@glossarypreamble@#1}%
   {\csdef{@glossarypreamble@#1}{}}%
   {}%
   \cspreto{@glossarypreamble@#1}{#2}%
  }%
  {%
    \GlossariesExtraWarning{Glossary '#1' is not defined}%
  }%
}
\ifdef\@gls@entry@field
{
 \renewcommand*{\@gls@entry@field}[2]{\csuse{glo@\glsdetoklabel{#1}@#2}}
}
{}
\renewcommand*{\ifglsused}[3]{%
  \glsdoifexists{#1}{\ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}}%
}
\renewcommand*{\longnewglossaryentry}{%
 \@ifstar\@glsxtr@s@longnewglossaryentry\@glsxtr@longnewglossaryentry
}
\newcommand{\@glsxtr@s@longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
      \bgroup
        \let\@org@newglossaryentryprehook\@newglossaryentryprehook
        \long\def\@newglossaryentryprehook{%
           \long\def\@glo@desc{#3}%
           \@org@newglossaryentryprehook
        }%
        \renewcommand*{\gls@assign@desc}[1]{%
           \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
           \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@descplural}%
        }
```

849

```
        \gls@defglossaryentry{#1}{#2}%
      \egroup
  }%
}
\newcommand{\@glsxtr@longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
      \bgroup
        \let\@org@newglossaryentryprehook\@newglossaryentryprehook
        \long\def\@newglossaryentryprehook{%
          \long\def\@glo@desc{#3\glsxtrpostlongdescription}%
          \@org@newglossaryentryprehook
        }%
        \renewcommand*{\gls@assign@desc}[1]{%
          \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
          \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@descplural}%
        }
        \gls@defglossaryentry{#1}{#2}%
      \egroup
  }%
}
\newcommand*{\glsxtrpostlongdescription}{\leavevmode\unskip\nopostdesc}
\renewcommand{\newignoredglossary}{%
 \@ifstar\glsxtr@s@newignoredglossary\glsxtr@org@newignoredglossary
}
\newcommand*{\glsxtr@org@newignoredglossary}[1]{%
  \ifcsdef{glolist@#1}
  {%
    \glsxtrundefaction{Glossary type '#1' already exists}{}%
  }%
  {%
    \ifdefempty\@ignored@glossaries
    {%
      \protected@edef\@ignored@glossaries{#1}%
    }%
    {%
      \protected@eappto\@ignored@glossaries{,#1}%
    }%
    \csgdef{glolist@#1}{,}%
    \ifcsundef{gls@#1@entryfmt}%
    {%
      \defglsentryfmt[#1]{\glsentryfmt}%
    }%
    {}%
    \ifdefempty\@gls@nohyperlist
    {%
      \renewcommand*{\@gls@nohyperlist}{#1}%
    }%
    {%
      \protected@eappto\@gls@nohyperlist{,#1}%
```

```
    }%
  }%
}
\newcommand*{\glsxtr@s@newignoredglossary}[1]{%
  \ifcsdef{glolist@#1}
  {%
    \glsxtrundefaction{Glossary type '#1' already exists}{}%
  }%
  {%
    \ifdefempty\@ignored@glossaries
    {%
      \protected@edef\@ignored@glossaries{#1}%
    }%
    {%
      \protected@eappto\@ignored@glossaries{,#1}%
    }%
    \csgdef{glolist@#1}{,}%
    \ifcsundef{gls@#1@entryfmt}%
    {%
      \defglsentryfmt[#1]{\glsentryfmt}%
    }%
    {}%
  }%
}
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \ifcsdef{@glotype@#1@title}%
      {\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}%
      {\def\glossarytoctitle{\glossarytitle}}%
    }%
  }%
}
{
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{@glotype@#1@title}%
    {\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}%
    {\def\glossarytoctitle{\glossarytitle}}%
  }
}
\newcommand{\provideignoredglossary}{%
 \@ifstar\glsxtr@s@provideignoredglossary\glsxtr@provideignoredglossary
}
\newcommand*{\glsxtr@provideignoredglossary}[1]{%
  \ifcsdef{glolist@#1}
```

851

```
    {}%
    {%
      \ifdefempty\@ignored@glossaries
      {%
        \protected@edef\@ignored@glossaries{#1}%
      }%
      {%
        \protected@eappto\@ignored@glossaries{,#1}%
      }%
      \csgdef{glolist@#1}{,}%
      \ifcsundef{gls@#1@entryfmt}%
      {%
        \defglsentryfmt[#1]{\glsentryfmt}%
      }%
      {}%
      \ifdefempty\@gls@nohyperlist
      {%
        \renewcommand*{\@gls@nohyperlist}{#1}%
      }%
      {%
        \protected@eappto\@gls@nohyperlist{,#1}%
      }%
    }%
}
\newcommand*{\glsxtr@s@provideignoredglossary}[1]{%
  \ifcsdef{glolist@#1}
  {}%
  {%
    \ifdefempty\@ignored@glossaries
    {%
      \protected@edef\@ignored@glossaries{#1}%
    }%
    {%
      \protected@eappto\@ignored@glossaries{,#1}%
    }%
    \csgdef{glolist@#1}{,}%
    \ifcsundef{gls@#1@entryfmt}%
    {%
      \defglsentryfmt[#1]{\glsentryfmt}%
    }%
    {}%
  }%
}
\newcommand*{\glsxtrcopytoglossary}[2]{%
  \glsdoifexists{#1}%
  {%
    \ifcsdef{glolist@#2}
    {%
      \protected@cseappto{glolist@#2}{#1,}%
    }%
```

```
    {%
      \glsxtrundefaction{Glossary type '#2' doesn't exist}{}%
    }%
  }%
}
\renewcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}%
  {%
    \protected@edef\glslabel{\glsdetoklabel{#1}}%
    \glsxtrundefaction{Glossary entry '\glslabel'
    has not been defined}{You need to define a glossary entry before
    you can reference it.}%
  }%
}
\renewcommand{\glsdoifnoexists}[2]{%
   \ifglsentryexists{#1}{%
   \glsxtrundefaction{Glossary entry '\glsdetoklabel{#1}'
   has already been defined}{}}{#2}%
}
\ifdef\glsdoifexistsordo
{%
  \renewcommand{\glsdoifexistsordo}[3]{%
    \ifglsentryexists{#1}{#2}%
    {%
      \glsxtrundefaction{Glossary entry '\glsdetoklabel{#1}'
      has not been defined}{You need to define a glossary entry
      before you can use it.}%
      #3
    }%
  }%
}
{%
  \glsxtr@warnonexistsordo\glsdoifexistsordo
  \newcommand{\glsdoifexistsordo}[3]{%
    \ifglsentryexists{#1}{#2}%
    {%
      \glsxtrundefaction{Glossary entry '\glsdetoklabel{#1}'
      has not been defined}{You need to define a glossary entry
      before you can use it.}%
      #3
    }%
  }%
}
\ifdef\doifglossarynoexistsordo
{%
  \renewcommand{\doifglossarynoexistsordo}[3]{%
    \ifglossaryexists*{#1}%
    {%
      \glsxtrundefaction{Glossary type '#1' already exists}{}%
      #3
```

```
    }%
    {#2}%
  }%
}
{%
  \glsxtr@warnonexistsordo\doifglossarynoexistsordo
  \newcommand{\doifglossarynoexistsordo}[3]{%
    \ifglossaryexists*{#1}%
    {%
      \glsxtrundefaction{Glossary type '#1' already exists}{}%
      #3%
    }%
    {#2}%
  }%
}

\appto\@newglossaryentryposthook{%
  \ifdefvoid\@glo@see
  {\csxdef{glo@\@glo@label @see}{}}%
  {%
    \csxdef{glo@\@glo@label @see}{\@glo@see}%
    \if@glsxtr@autoseeindex
      \@glsxtr@autoindexcrossrefs
    \fi
  }%
}
\appto\@gls@keymap{,{see}{see}}
\newcommand*{\glsxtrusesee}[1]{%
  \glsdoifexists{#1}%
  {%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@see}%
    \ifdefempty\@glo@see
    {}%
    {%
      \expandafter\glsxtr@usesee\@glo@see\@end@glsxtr@usesee
    }%
  }%
}
\newcommand*{\glsxtr@usesee}[1][\seename]{%
  \@glsxtr@usesee[#1]%
}
\def\@glsxtr@usesee[#1]#2\@end@glsxtr@usesee{%
  \glsxtruseseeformat{#1}{#2}%
}
\newcommand*{\glsxtruseseeformat}[2]{%
  \glsseeformat[#1]{#2}{}%
}
\renewcommand*{\glsseeitemformat}[1]{%
 \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
}
```

```
\newcommand*{\glsxtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {\glsxtrhiername{\glscurrentfieldvalue}\glsxtrhiernamesep}%
    {}%
    \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
  }%
}
\newcommand*{\Glsxtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {%
      \Glsxtrhiername{\glscurrentfieldvalue}\glsxtrhiernamesep
      \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
    }%
    {\ifglshasshort{#1}{\Glsfmttext{#1}}{\Glsfmtname{#1}}}%
  }%
}
\newcommand*{\GlsXtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {\GlsXtrhiername{\glscurrentfieldvalue}\glsxtrhiernamesep}%
    {}%
    \ifglshasshort{#1}{\Glsfmttext{#1}}{\Glsfmtname{#1}}%
  }%
}
\newcommand*{\GLSxtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {%
      \GLSxtrhiername{\glscurrentfieldvalue}\glsxtrhiernamesep
      \ifglshasshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
    }%
    {\ifglshasshort{#1}{\GLSfmttext{#1}}{\GLSfmtname{#1}}}%
  }%
}
\newcommand*{\GLSXTRhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsxtrifhasfield{parent}{#1}%
    {\GLSXTRhiername{\glscurrentfieldvalue}\glsxtrhiernamesep}%
    {}
    \ifglshasshort{#1}{\GLSfmttext{#1}}{\GLSfmtname{#1}}%
  }%
}
\newcommand*{\glsxtrhiernamesep}{\,{\small$\triangleright$}\,}
```

```
\newcommand*{\glsxtruseseealso}[1]{%
  \glsdoifexists{#1}%
  {%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@seealso}%
    \ifdefempty\@glo@see
    {}%
    {%
      \expandafter\glsxtruseseealsoformat\expandafter{\@glo@see}%
    }%
  }%
}
\newcommand*{\glsxtrusealias}[1]{%
  \glsdoifexists{#1}%
  {%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@alias}%
    \ifdefempty\@glo@see
    {}%
    {%
      \glsxtruseseeformat{\seename}{\@glo@see}%
    }%
  }%
}
\newcommand*{\glsxtruseseealsoformat}[1]{%
  \glsseeformat[\seealsoname]{#1}{}%
}
\newrobustcmd{\glsxtrseelist}[1]{%
  \protected@edef\@glo@tmp{\noexpand\glsseelist{#1}}\@glo@tmp
}
\renewrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \let\@glsseeitem\@glsxtr@seefirstitem
  \let\@glsseelastsep\glsseelastsep
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\@glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\@glsseelastsep
    \let\@gls@donext\glsseesep
    \let\@glsseeitem\@glsxtr@seeitem
    \let\@glsseelastsep\glsseelastoxfordsep
  }%
}
\newcommand*{\@glsxtr@seeitem}[1]{%
 \glsxtrifmulti{#1}{\mglsseeitem{#1}}{\glsseeitem{#1}}%
}
\newcommand*{\@glsxtr@seefirstitem}[1]{%
```

856

```
  \glsxtrifmulti{#1}{\mglsseefirstitem{#1}}{\glsseefirstitem{#1}}%
}
\newcommand*{\mglsseeitem}[1]{%
 \mglsname[all={noindex},setup={hyper=allmain}]{#1}%
}
\newcommand*{\mglsseefirstitem}{\mglsseeitem}
\newcommand*{\glsseefirstitem}{\glsseeitem}
\newcommand*{\glsseelastoxfordsep}{\glsseelastsep}
\ifdef\alsoname
{\providecommand{\seealsoname}{\alsoname}}
{\providecommand{\seealsoname}{see also}}
\ifdef\@xdycrossrefhook
{
  \appto\@xdycrossrefhook{%
    \write\glswrite{(define-crossref-class \string"seealso\string"
      :unverified )}%
    \write\glswrite{(markup-crossref-list
      :class \string"seealso\string"^^J\space\space\space
      :open \string"\string\glsxtruseseealsoformat\glsopenbrace\string"
      :close \string"\glsclosebrace\string")}%
  }
  \appto\@xdylocationclassorder{\space\string"seealso\string"}
  \newrobustcmd*{\glsxtrindexseealso}[2]{%
    \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
      \@glsxtr@recordsee{#1}{#2}%
    \fi
    \glsdoifexists{#1}%
    {%
      \@@glsxtrwrglossmark
      \def\@gls@xref{#2}%
      \@onelevel@sanitize\@gls@xref
      \@gls@checkmkidxchars\@gls@xref
      \gls@glossary{\csname glo@#1@type\endcsname}{%
        (indexentry
          :tkey (\csname glo@#1@index\endcsname)
          :xref (\string"\@gls@xref\string")
          :attr \string"seealso\string"
        )
      }%
    }%
  }
}
{
  \newrobustcmd*{\glsxtrindexseealso}{\glssee[\seealsoname]}
}
\ifdef\gls@set@xr@key
{
  \define@key{glossentry}{alias}{%
    \gls@set@xr@key{alias}{\@glo@alias}{#1}%
  }
```

```
\define@key{glossentry}{seealso}{%
  \gls@set@xr@key{seealso}{\@glo@seealso}{#1}%
}
\appto\@gls@keymap{,{alias}{alias},{seealso}{seealso}}
\appto\@newglossaryentryprehook{\def\@glo@alias{}\def\@glo@seealso{}}%
\appto\@newglossaryentryposthook{%
  \ifdefvoid\@glo@seealso
   {\csxdef{glo@\@glo@label @seealso}{}}%
   {%
      \csxdef{glo@\@glo@label @seealso}{\@glo@seealso}%
      \if@glsxtr@autoseeindex
        \@glsxtr@autoindexcrossrefs
      \fi
   }%
  \ifdefvoid\@glo@alias
   {\csxdef{glo@\@glo@label @alias}{}}%
   {%
      \csxdef{glo@\@glo@label @alias}{\@glo@alias}%
   }%
}
\newcommand*{\glsxtralias}[1]{\@gls@entry@field{#1}{alias}}
\newcommand*{\glsxtrseealsolabels}[1]{\@gls@entry@field{#1}{seealso}}
\appto\@glo@autoseehook{%
  \ifdefvoid\@glo@alias
  {%
    \ifdefvoid\@glo@seealso
    {}%
    {%
      \protected@edef\@do@glssee{\noexpand\glsxtrindexseealso
        {\@glo@label}{\@glo@seealso}}%
      \@do@glssee
    }%
  }%
  {%
    \ifdefvoid\@glo@see
    {%
      \protected@edef\@do@glssee{\noexpand\glssee{\@glo@label}{\@glo@alias}}%
      \@do@glssee
    }%
    {}%
  }%
}%
}
{
  \glsaddstoragekey*{alias}{}{\glsxtralias}
  \glsaddstoragekey*{seealso}{}{\glsxtrseealsolabels}
  \appto\@newglossaryentryposthook{%
    \ifcsvoid{glo@\@glo@label @alias}%
    {%
      \ifcsvoid{glo@\@glo@label @seealso}%
```

```
        {}%
        {%
          \protected@edef\@do@glssee{\noexpand\glsxtrindexseealso
            {\@glo@label}{\csuse{glo@\@glo@label @seealso}}}%
          \@do@glssee
        }%
      }%
      {%
        \ifdefvoid\@glo@see
        {%
          \protected@edef\@do@glssee{\noexpand\glssee
            {\@glo@label}{\csuse{glo@\@glo@label @alias}}}%
          \@do@glssee
        }%
        {}%
      }%
    }
  }
  \AtEndDocument{\if@glsxtrindexcrossrefs\glsxtraddallcrossrefs\fi}
  \newcommand*{\glsxtraddallcrossrefs}{%
    \forallglossaries{\@glo@type}%
    {%
      \forglsentries[\@glo@type]{\@glo@label}%
      {%
        \ifglsused{\@glo@label}%
        {\expandafter\@glsxtr@addunusedxrefs\expandafter{\@glo@label}}{}%
      }%
    }%
  }
  \newcommand*{\@glsxtr@addunusedxrefs}[1]{%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@see}%
    \ifdefvoid\@glo@see
    {}%
    {%
      \expandafter\glsxtr@addunused\@glo@see\@end@glsxtr@addunused
    }%
    \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@seealso}%
    \ifdefvoid\@glo@see
    {}%
    {%
      \expandafter\glsxtr@addunused\@glo@see\@end@glsxtr@addunused
    }%
  }
  \newcommand*{\glsxtr@addunused}[1][]{%
    \@glsxtr@addunused
  }
  \def\@glsxtr@addunused#1\@end@glsxtr@addunused{%
   \@for\@glsxtr@label:=#1\do
   {%
      \glsxtrifmulti\@glsxtr@label
```

```
    {%
      \letcs\@glsxtr@labellist{@gls@combined@\@glsxtr@label @list}%
      \@for\@glsxtr@multilabel:=\@glsxtr@labellist\do
      {\@glsxtr@addunused\@glsxtr@multilabel\@end@glsxtr@addunused}%
    }%
    {%
      \ifglsused{\@glsxtr@label}{}%
      {%
        \glsadd[format=glsxtrunusedformat]{\@glsxtr@label}%
        \glsunset{\@glsxtr@label}%
        \expandafter\@glsxtr@addunusedxrefs\expandafter{\@glsxtr@label}%
      }%
    }%
 }%
}
\newcommand*{\glsxtrunusedformat}[1]{\unskip}
\ifdef\gls@begindocdefs
{%
  \renewcommand*{\gls@begindocdefs}{%
    \ifnum\@glsxtr@docdefval=1\relax
      \@gls@enablesavenonumberlist
      \edef\@gls@restoreat{%
        \noexpand\catcode`\noexpand\@=\number\catcode`\@\relax}%
      \makeatletter
      \InputIfFileExists{\jobname.glsdefs}{}{}%
      \@gls@restoreat
      \undef\@gls@restoreat
      \gls@defdocnewglossaryentry
    \else
      \ifnum\@glsxtr@docdefval=3\relax
        \@gls@enablesavenonumberlist
        \let\gls@checkseeallowed\relax
        \let\newglossaryentry\new@atom@glossaryentry
        \global\newwrite\@gls@deffile
        \immediate\openout\@gls@deffile=\jobname.glsdefs
        \forallglsentries{\@glsentry}{\@gls@writedef{\@glsentry}}%
      \fi
    \fi
  }
}
{%
  \ifnum\@glsxtr@docdefval=3\relax
    \PackageError{glossaries-extra}{Package option
    `docdef=\@glsxtr@docdefsetting' requires at least version 4.37
    of the base glossaries.sty package}{}
  \fi
}
\newrobustcmd{\new@atom@glossaryentry}[2]{%
  \gls@defglossaryentry{#1}{#2}%
  \@gls@writedef{#1}%
```

860

```
}
\let\glsxtr@orgmakenoidxglossaries\makenoidxglossaries
\renewcommand{\makenoidxglossaries}{%
 \@domakeglossaries
 {%
  \ifdefequal\@glsxtr@record@setting\@glsxtr@record@setting@off
  {%
    \glsxtr@orgmakenoidxglossaries
    \renewcommand{\@do@seeglossary}[2]{%
      \@@glsxtrwrglossmark
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
          {\csname glo@\@gls@label @type\endcsname}%
          {\@gls@label}%
          {%
            \string\glsseeformat##2{}%
          }%
      }%
    }%
    \if@glsxtrdocdefrestricted
      \renewcommand*{\@gls@reference}[3]{%
        \ifcsundef{@glsref@##1}{\csgdef{@glsref@##1}{}}{}%
        \ifinlistcs{##2}{@glsref@##1}%
        {}%
        {\listcsgadd{@glsref@##1}{##2}}%
        \ifcsundef{glo@\glsdetoklabel{##2}@loclist}%
        {\csgdef{glo@\glsdetoklabel{##2}@loclist}{}}%
        {}%
        \listcsgadd{glo@\glsdetoklabel{##2}@loclist}{##3}%
      }%
    \else
      \@glsxtrdocdeffalse
    \fi
    \disable@keys{glossaries-extra.sty}{docdef}%
  }%
  {%
    \PackageError{glossaries-extra}{\string\makenoidxglossaries\space
     not permitted\MessageBreak
     with record=\@glsxtr@record@setting\space package option}%
    {You may only use \string\makenoidxglossaries\ space with the
     record=off option}%
  }%
 }%
}
\renewcommand*{\gls@defdocnewglossaryentry}{%
  \ifcase\@glsxtr@docdefval
    \renewcommand*{\newglossaryentry}[2]{%
      \PackageError{glossaries-extra}{Glossary entries must
       be \MessageBreak defined in the preamble with \MessageBreak
```

```
      package option 'docdef=false'\MessageBreak(consider using
      'docdef=restricted')}{Move your glossary definitions to
      the preamble. You can also put them in a \MessageBreak separate file
      and load them with \string\loadglsentries.}%
    }%
  \or
    \let\gls@checkseeallowed\relax
    \let\newglossaryentry\new@glossaryentry
  \else
    \let\gls@checkseeallowed\relax
  \fi
}%
\newcommand*{\GlsXtrEnableOnTheFly}{%
  \@ifstar\@sGlsXtrEnableOnTheFly\@GlsXtrEnableOnTheFly
}
\newcommand*{\@sGlsXtrEnableOnTheFly}{%
  \renewcommand*{\glsdetoklabel}[1]{%
    \expandafter\@glsxtr@ifcsstart\string##1 \@glsxtr@end@
    {%
      \expandafter\detokenize\expandafter{##1}%
    }%
    {\detokenize{##1}}%
  }%
  \@GlsXtrEnableOnTheFly
}
\def\@glsxtr@ifcsstart#1#2\@glsxtr@end@#3#4{%
  \expandafter\if\glsbackslash#1%
    #3%
  \else
    #4%
  \fi
}
\newcommand*{\glsxtrstarflywarn}{%
  \GlossariesExtraWarning{Experimental starred version of
  \string\GlsXtrEnableOnTheFly\space in use (please ensure you have
  read the warnings in the glossaries-extra user manual)}%
}
\newcommand*{\@GlsXtrEnableOnTheFly}{%
  \newcommand*{\glsxtrcat}{general}
  \newcommand*{\glsxtr}[1][]{%
   \def\glsxtr@keylist{##1}%
   \@glsxtr
  }
  \newcommand*{\@glsxtr}[2][]{%
   \ifglsentryexists{##2}%
   {%
     \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
   }%
   {%
     \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
```

```
      description={\nopostdesc},##1}%
 }%
 \expandafter\gls\expandafter[\glsxtr@keylist]{##2}%
}
\newcommand*{\Glsxtr}[1][]{%
 \def\glsxtr@keylist{##1}%
 \@Glsxtr
}
\newcommand*{\@Glsxtr}[2][]{%
 \ifglsentryexists{##2}%
 {%
    \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
 }%
 {%
    \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
      description={\nopostdesc},##1}%
 }%
 \expandafter\Gls\expandafter[\glsxtr@keylist]{##2}%
}
\newcommand*{\glsxtrpl}[1][]{%
 \def\glsxtr@keylist{##1}%
 \@glsxtrpl
}
\newcommand*{\@glsxtrpl}[2][]{%
 \ifglsentryexists{##2}%
 {%
    \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
 }%
 {%
    \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
      description={\nopostdesc},##1}%
 }%
 \expandafter\glspl\expandafter[\glsxtr@keylist]{##2}%
}
\newcommand*{\Glsxtrpl}[1][]{%
 \def\glsxtr@keylist{##1}%
 \@Glsxtrpl
}
\newcommand*{\@Glsxtrpl}[2][]{%
 \ifglsentryexists{##2}
 {%
    \ifblank{##1}{}{\GlsXtrWarning{##1}{##2}}%
 }%
 {%
    \gls@defglossaryentry{##2}{name={##2},category=\glsxtrcat,
      description={\nopostdesc},##1}%
 }%
 \expandafter\Glspl\expandafter[\glsxtr@keylist]{##2}%
}
\newcommand*{\GlsXtrWarning}[2]{%
```

```
    \def\@glsxtr@optlist{##1}%
    \@onelevel@sanitize\@glsxtr@optlist
    \GlossariesExtraWarning{The options '\@glsxtr@optlist' have
    been ignored for entry '##2' as it has already been defined}%
  }
  \renewcommand\@printglossary[2]{%
    \def\@glsxtr@printglossopts{##1}%
    \@glsxtr@orgprintglossary{##1}{##2}%
    \def\@glsxtr{\@glsxtr@disabledflycommand\glsxtr}%
    \def\@glsxtrpl{\@glsxtr@disabledflycommand\glsxtrpl}%
    \def\@Glsxtr{\@glsxtr@disabledflycommand\Glsxtr}%
    \def\@Glsxtrpl{\@glsxtr@disabledflycommand\Glsxtrpl}%
  }
  \newcommand*{\@glsxtr@disabledflycommand}[1]{%
    \PackageError{glossaries-extra}%
    {\string##1\space can't be used after any of the \MessageBreak
     glossaries have been displayed}%
    {The on-the-fly commands enabled by
     \string\GlsXtrEnableOnTheFly\space may only be used \MessageBreak
     before the glossaries. If you want to use any entries \MessageBreak
     after any of the glossaries, you must use the standard \MessageBreak
     method of first defining the entry and then using the \MessageBreak
     entry with commands like \string\gls}%
     \@@glsxtr@disabledflycommand
  }%
  \newcommand*{\@@glsxtr@disabledflycommand}[2][]{##2}
  \let\GlsXtrEnableOnTheFly\relax
}
\@onlypreamble\GlsXtrEnableOnTheFly
\newcommand*{\@glsxtr@current@style}{\@glossary@default@style}
\renewcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries-extra}{Glossary style '#1' undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
    \protected@edef\@glsxtr@current@style{#1}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\ifdef\@glossary@default@style
{}
{%
  \let\@glossary@default@style\relax
}
\ifdef\glslistdottedwidth
{%
```

```
      \ifdim\glslistdottedwidth=.5\hsize
        \setlength{\glslistdottedwidth}{-\dimexpr\maxdimen-1sp\relax}
        \AtBeginDocument{%
          \ifdim\glslistdottedwidth=-\dimexpr\maxdimen-1sp\relax
           \setlength{\glslistdottedwidth}{.5\columnwidth}%
          \fi
       }%
     \fi
   }
   {}%
   \ifdef\glsdescwidth
   {%
     \ifdim\glsdescwidth=.6\hsize
       \setlength{\glsdescwidth}{-\dimexpr\maxdimen-1sp\relax}
       \AtBeginDocument{%
         \ifdim\glsdescwidth=-\dimexpr\maxdimen-1sp\relax
          \setlength{\glsdescwidth}{.6\columnwidth}%
         \fi
      }%
    \fi
   }
   {}%
   \ifdef\glspagelistwidth
   {%
     \ifdim\glspagelistwidth=.1\hsize
       \setlength{\glspagelistwidth}{-\dimexpr\maxdimen-1sp\relax}
       \AtBeginDocument{%
         \ifdim\glspagelistwidth=-\dimexpr\maxdimen-1sp\relax
          \setlength{\glspagelistwidth}{.1\columnwidth}%
         \fi
      }%
    \fi
   }
   {}%
   \def\org@glossaryentrynumbers#1{#1\gls@save@numberlist{#1}}%
   \ifx\org@glossaryentrynumbers\glossaryentrynumbers
     \glsnonumberlistfalse
     \renewcommand*{\glossaryentrynumbers}[1]{%
       \ifglsentryexists{\glscurrententrylabel}%
       {%
         \@glsxtrpreloctag
         \GlsXtrFormatLocationList{#1}%
         \@glsxtrpostloctag
         \gls@save@numberlist{#1}%
       }{}%
     }%
   \else
     \glsnonumberlisttrue
     \renewcommand*{\glossaryentrynumbers}[1]{%
       \ifglsentryexists{\glscurrententrylabel}%
```

```
    {%
      \gls@save@numberlist{#1}%
    }{}%
  }%
\fi
\newcommand*{\GlsXtrFormatLocationList}[1]{#1}
\newcommand*{\GlsXtrEnablePreLocationTag}[2]{%
  \let\@glsxtrpreloctag\@@glsxtrpreloctag
  \let\@glsxtrpostloctag\@@glsxtrpostloctag
  \renewcommand*{\@glsxtr@pagetag}{#1}%
  \renewcommand*{\@glsxtr@pagestag}{#2}%
  \renewcommand*{\@glsxtr@savepreloctag}[2]{%
    \csgdef{@glsxtr@preloctag@##1}{##2}%
  }%
  \renewcommand*{\@glsxtr@doloctag}{%
    \ifcsundef{@glsxtr@preloctag@\glscurrententrylabel}%
    {%
      \GlossariesWarning{Missing pre-location tag for `\glscurrententrylabel'.
        Rerun required}%
    }%
    {%
      \csuse{@glsxtr@preloctag@\glscurrententrylabel}%
    }%
  }%
}
\@onlypreamble\GlsXtrEnablePreLocationTag
\newcommand*{\@@glsxtrpreloctag}{%
    \let\@glsxtr@org@delimN\delimN
    \let\@glsxtr@org@delimR\delimR
    \let\@glsxtr@org@glsignore\glsignore
    \gdef\@glsxtr@thisloctag{\@glsxtr@pagetag}%
    \renewcommand*{\delimN}{%
      \gdef\@glsxtr@thisloctag{\@glsxtr@pagestag}%
      \@glsxtr@org@delimN}%
    \renewcommand*{\delimR}{%
      \gdef\@glsxtr@thisloctag{\@glsxtr@pagestag}%
      \@glsxtr@org@delimR}%
    \renewcommand*{\glsignore}[1]{%
      \gdef\@glsxtr@thisloctag{\relax}%
      \@glsxtr@org@glsignore{##1}}%
    \@glsxtr@doloctag
}
\newcommand*{\@glsxtrpreloctag}{}
\newcommand*{\@glsxtr@pagetag}{}%
\newcommand*{\@glsxtr@pagestag}{}%
\newcommand*{\@@glsxtrpostloctag}{%
    \let\delimN\@glsxtr@org@delimN
    \let\delimR\@glsxtr@org@delimR
    \let\glsignore\@glsxtr@org@glsignore
    \protected@write\@auxout{}%
```

```
    {\string\@glsxtr@savepreloctag{\glscurrententrylabel}{\@glsxtr@thisloctag}}%
}
\newcommand*{\@glsxtrpostloctag}{}
\newcommand*{\@glsxtr@savepreloctag}[2]{}
\protected@write\@auxout{}{%
  \string\providecommand\string\@glsxtr@savepreloctag[2]{}}
\newcommand*{\@glsxtr@doloctag}{}
\renewcommand*{\KV@printgloss@nonumberlist}[1]{%
 \XKV@plfalse
 \XKV@sttrue
 \XKV@checkchoice[\XKV@resa]{#1}{true,false}%
 {%
   \csname glsnonumberlist\XKV@resa\endcsname
   \ifglsnonumberlist
     \def\glossaryentrynumbers##1{\gls@save@numberlist{##1}}%
   \else
     \def\glossaryentrynumbers##1{%
       \@glsxtrpreloctag
       \GlsXtrFormatLocationList{##1}%
       \@glsxtrpostloctag
       \gls@save@numberlist{##1}}%
   \fi
 }%
}
\renewcommand*{\glsentryfmt}{%
 \ifglshasshort{\glslabel}{\glssetabbrvfmt{\glscategory{\glslabel}}}{}%
 \glsifregular{\glslabel}%
 {\glsxtrregularfont{\glsgenentryfmt}}%
 {%
   \ifglshasshort{\glslabel}%
   {\glsxtrabbreviationfont{\glsxtrgenabbrvfmt}}%
   {\glsxtrregularfont{\glsgenentryfmt}}%
 }%
}
\newcommand*{\glsxtrregularfont}[1]{#1}
\newcommand*{\glsxtrabbreviationfont}[1]{#1}
\renewcommand{\@gls@field@link}[4][]{%
 \@glsxtr@record{#2}{#3}{glslink}%
 \glsdoifexists{#3}%
 {%
   \let\glsxtrorg@ifKV@glslink@hyper\ifKV@glslink@hyper
   \@gls@save@glslocal
   \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
   \def\glscustomtext{#4}%
   \@glsxtr@field@linkdefs
   #1%
   \@gls@link[#2]{#3}{#4}%
   \let\ifKV@glslink@hyper\glsxtrorg@ifKV@glslink@hyper
   \@gls@restore@glslocal
 }%
```

```
  \glspostlinkhook
}
\let\@glsxtr@org@gls@\@gls@
\def\@gls@#1#2{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \@glsxtr@org@gls@{#1}{#2}%
}%
\let\@glsxtr@org@glspl@\@glspl@
\def\@glspl@#1#2{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \@glsxtr@org@glspl@{#1}{#2}%
}%
\let\@glsxtr@org@Gls@\@Gls@
\def\@Gls@#1#2{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \@glsxtr@org@Gls@{#1}{#2}%
}%
\let\@glsxtr@org@Glspl@\@Glspl@
\def\@Glspl@#1#2{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \@glsxtr@org@Glspl@{#1}{#2}%
}%
\let\@glsxtr@org@GLS@\@GLS@
\def\@GLS@#1#2{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \@glsxtr@org@GLS@{#1}{#2}%
}%
\let\@glsxtr@org@GLSpl@\@GLSpl@
\def\@GLSpl@#1#2{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \@glsxtr@org@GLSpl@{#1}{#2}%
}%
\renewcommand*{\@glsdisp}[3][]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \def\glscustomtext{#3}%
    \def\glsinsert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\renewcommand*{\@gls@@link}[3][]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexistsordo{#2}%
  {%
```

```
    \let\do@gls@link@checkfirsthyper\relax
    \def\glscustomtext{#3}%
    \@glsxtr@field@linkdefs
    \@gls@link[#1]{#2}{#3}%
  }%
  {%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newcommand*{\glsxtrinitwrgloss}{%
 \glsifattribute{\glslabel}{wrgloss}{after}%
 {%
    \glsxtrinitwrglossbeforefalse
 }%
 {%
    \glsxtrinitwrglossbeforetrue
 }%
}
\newif\ifglsxtrinitwrglossbefore
\glsxtrinitwrglossbeforetrue
\define@choicekey{glslink}{wrgloss}%
[\@glsxtr@wrglossval\@glsxtr@wrglossnr]%
{before,after}%
{%
  \ifcase\@glsxtr@wrglossnr\relax
    \glsxtrinitwrglossbeforetrue
  \or
    \glsxtrinitwrglossbeforefalse
  \fi
}
\define@key{glslink}{thevalue}{\def\@glsxtr@thevalue{#1}}
\define@key{glslink}{theHvalue}{\def\@glsxtr@theHvalue{#1}}
\define@boolkey{glslink}[glsxtr@]{hyperoutside}[true]{}
\glsxtr@hyperoutsidetrue
\define@key{glslink}{textformat}{%
  \ifcsdef{#1}
  {%
    \letcs{\@glsxtr@local@textformat}{#1}%
  }%
  {%
    \PackageError{glossaries-extra}{Unknown control sequence name '#1'}{}%
  }%
}
\define@key{glslink}{prefix}{\def\glolinkprefix{#1}}
\newcommand*{\glsxtrinithyperoutside}{%
 \glsifattribute{\glslabel}{hyperoutside}{false}%
 {%
    \glsxtr@hyperoutsidefalse
 }%
```

```
 {%
    \glsxtr@hyperoutsidetrue
 }%
}
\newcommand*{\glsxtr@inc@linkcount}{}
\newcommand*{\glslinkpresetkeys}{}
\newrobustcmd*{\GlsXtrExpandedFmt}[2]{%
  \protected@edef\@glsxtr@tmp{#2}%
  \expandafter#1\expandafter{\@glsxtr@tmp}%
}
\newcommand*{\@glsxtr@use@equation@counter}{%
  \@glsxtr@ifnum@mmode{\def\@gls@counter{equation}}{}%
}
\newcommand*{\glsxtr@do@autoadd}[1]{}
\newcommand*{\GlsXtrAutoAddOnFormat}[3][\glslabel]{%
  \renewcommand*{\glsxtr@do@autoadd}[1]{%
    \begingroup
      \protected@edef\@glsxtr@do@autoadd{%
          \noexpand\ifstrequal{##1}{glslink}%
          {%
            \noexpand\DTLifinlist{\@glsnumberformat}{#2}{\noexpand\glsadd[format={\@glsnumberformat},#
          }%
          {}%
      }%
      \@glsxtr@do@autoadd
    \endgroup
  }%
}
\providecommand*{\glslinkwrcontent}[1]{{#1}}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \let\@glsnumberformat\@glsxtr@defaultnumberformat
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \let\@glsxtr@org@glolinkprefix\glolinkprefix
  \let\@glsxtr@local@textformat\relax
  \def\@glsxtr@thevalue{}%
  \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
  \glsxtrinitwrgloss
  \glsxtrinithyperoutside
  \@gls@setdefault@glslink@opts
   \glsxtr@inc@linkcount
   \if@glsxtr@equations
     \@glsxtr@use@equation@counter
   \fi
```

```
\do@glsdisablehyperinlist
\do@gls@link@checkfirsthyper
\glslinkpresetkeys
\setkeys{glslink}{#1}%
\glsxtr@do@autoadd{glslink}%
\glslinkpostsetkeys
\ifdefempty{\@glsxtr@thevalue}%
{%
  \@gls@saveentrycounter
}%
{%
  \let\theglsentrycounter\@glsxtr@thevalue
  \def\theHglsentrycounter{\@glsxtr@theHvalue}%
}%
\@gls@setsort{\glslabel}%
\ifx\@glsxtr@local@textformat\relax
   \glshasattribute{\glslabel}{textformat}%
   {%
      \protected@edef\@glsxtr@attrval{\glsgetattribute{\glslabel}{textformat}}%
      \ifcsdef{\@glsxtr@attrval}%
      {%
         \letcs{\@glsxtr@textformat}{\@glsxtr@attrval}%
      }%
      {%
         \GlossariesExtraWarning{Unknown control sequence name
         '\@glsxtr@attrval' supplied in textformat attribute
         for entry '\glslabel'. Reverting to default \string\glstextformat}%
         \let\@glsxtr@textformat\glstextformat
      }%
   }%
   {%
      \let\@glsxtr@textformat\glstextformat
   }%
\else
   \let\@glsxtr@textformat\@glsxtr@local@textformat
\fi
\glslinkwrcontent
{%
  \ifglsxtrinitwrglossbefore
    \@do@wrglossary{#2}%
  \fi
  \ifKV@glslink@hyper
    \ifglsxtr@hyperoutside
      \@glslink{\glolinkprefix\glslabel}{\@glsxtr@textformat{#3}}%
    \else
      \@glsxtr@textformat{\@glslink{\glolinkprefix\glslabel}{#3}}%
    \fi
  \else
    \ifglsxtr@hyperoutside
      \glsdonohyperlink{\glolinkprefix\glslabel}{\@glsxtr@textformat{#3}}%
```

```
        \else
          \@glsxtr@textformat{\glsdonohyperlink{\glolinkprefix\glslabel}{#3}}%
        \fi
      \fi
      \ifglsxtrinitwrglossbefore
      \else
        \@do@wrglossary{#2}%
      \fi
    }%
    \let\glolinkprefix\@glsxtr@org@glolinkprefix
    \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
    \@gls@restore@glslocal
}
\define@key{glossadd}{thevalue}{\def\@glsxtr@thevalue{#1}}
\define@key{glossadd}{theHvalue}{\def\@glsxtr@theHvalue{#1}}
\newcommand*{\glsaddpresetkeys}{}

\newcommand*{\glsaddpostsetkeys}{}
\renewrobustcmd*{\glsadd}[2][]{%
  \glsxtrifinmark
  {}%
  {%
    \@gls@adjustmode
    \begingroup
      \@glsxtr@record{#1}{#2}{glossadd}%
      \glsdoifexists{#2}%
      {%
        \let\@glsnumberformat\@glsxtr@defaultnumberformat
        \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
        \def\@glsxtr@thevalue{}%
        \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
        \glsaddpresetkeys
        \setkeys{glossadd}{#1}%
        \glsaddpostsetkeys
        \ifdefempty{\@glsxtr@thevalue}%
        {%
          \@gls@saveentrycounter
        }%
        {%
          \let\theglsentrycounter\@glsxtr@thevalue
          \def\theHglsentrycounter{\@glsxtr@theHvalue}%
        }%
        \@gls@setsort{#2}%
        \KV@glslink@noindexfalse
        \@@do@wrglossary{#2}%
      }%
    \endgroup
  }%
}
\newrobustcmd{\glsaddeach}[2][]{%
```

```
      \@for\@gls@thislabel:=#2\do{\glsadd[#1]{\@gls@thislabel}}%
}
\newcommand*{\@glsxtr@field@linkdefs}{%
  \let\glsxtrifwasfirstuse\@secondoftwo
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@firstofthree
  \let\glsinsert\@empty
}
\newcommand*{\glsxtrassignfieldfont}[1]{%
  \ifglsentryexists{#1}%
  {%
    \ifglshasshort{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glsifregular{#1}%
      {\let\@gls@field@font\glsxtrregularfont}%
      {\let\@gls@field@font\@firstofone}%
    }%
    {%
      \glsifnotregular{#1}%
      {\let\@gls@field@font\@firstofone}%
      {\let\@gls@field@font\glsxtrregularfont}%
    }%
  }%
  {%
    \let\@gls@field@font\@gobble
  }%
}
\def\@glstext@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccesstext{#2}#3}}%
}
\def\@GLStext@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]{#1}{#2}%
    {\@gls@field@font{\GLSaccesstext{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@Glstext@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@secondofthree]{#1}{#2}%
    {\@gls@field@font{\Glsaccesstext{#2}#3}}%
}
\newcommand*{\glsxtrchecknohyperfirst}[1]{%
  \glsifattribute{#1}{nohyperfirst}{true}{\KV@glslink@hyperfalse}{}%
}
\def\@glsfirst@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsxtrifwasfirstuse\@firstoftwo
   \glsxtrchecknohyperfirst{#2}%
```

```
    ]{#1}{#2}%
    {\@gls@field@font{\glsaccessfirst{#2}#3}}%
}
\def\@Glsfirst@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsxtrifwasfirstuse\@firstoftwo
   \let\glscapscase\@secondofthree
   \glsxtrchecknohyperfirst{#2}%
  ]%
    {#1}{#2}{\@gls@field@font{\Glsaccessfirst{#2}#3}}%
}
\def\@GLSfirst@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsxtrifwasfirstuse\@firstoftwo
   \let\glscapscase\@thirdofthree
   \glsxtrchecknohyperfirst{#2}%
  ]%
    {#1}{#2}{\@gls@field@font{\GLSaccessfirst{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glsplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glsifplural\@firstoftwo]{#1}{#2}%
     {\@gls@field@font{\glsaccessplural{#2}#3}}%
}
\def\@Glsplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsifplural\@firstoftwo
   \let\glscapscase\@secondofthree
  ]%
     {#1}{#2}{\@gls@field@font{\Glsaccessplural{#2}#3}}%
}
\def\@GLSplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsifplural\@firstoftwo
   \let\glscapscase\@thirdofthree
  ]%
     {#1}{#2}{\@gls@field@font{\GLSaccessplural{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glsfirstplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsxtrifwasfirstuse\@firstoftwo
   \let\glsifplural\@firstoftwo
   \glsxtrchecknohyperfirst{#2}%
  ]%
    {#1}{#2}{\@gls@field@font{\glsaccessfirstplural{#2}#3}}%
```

```
}
\def\@Glsfirstplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsxtrifwasfirstuse\@firstoftwo
   \let\glsifplural\@firstoftwo
   \let\glscapscase\@secondofthree
   \glsxtrchecknohyperfirst{#2}%
  ]%
   {#1}{#2}{\@gls@field@font{\Glsaccessfirstplural{#2}#3}}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glsxtrifwasfirstuse\@firstoftwo
   \let\glsifplural\@firstoftwo
   \let\glscapscase\@thirdofthree
   \glsxtrchecknohyperfirst{#2}%
  ]%
   {#1}{#2}%
   {\@gls@field@font{\GLSaccessfirstplural{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glsname@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccessname{#2}#3}}%
}
\def\@Glsname@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]{#1}{#2}%
  {\@gls@field@font{\Glsaccessname{#2}#3}}%
}
\def\@GLSname@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@thirdoftwo]%
     {#1}{#2}%
     {\@gls@field@font{\GLSaccessname{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glsdesc@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccessdesc{#2}#3}}%
}
\def\@Glsdesc@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]{#1}{#2}%
  {\@gls@field@font{\Glsaccessdesc{#2}#3}}%
}
\def\@GLSdesc@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
```

```
    \@gls@field@link[\let\glscapscase\@thirdoftwo]%
      {#1}{#2}{\@gls@field@font{\GLSaccessdesc{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glsdescplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo
   \let\glsifplural\@firstoftwo
  ]{#1}{#2}{\@gls@field@font{\glsaccessdescplural{#2}#3}}%
}
\def\@Glsdescplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo
   \let\glsifplural\@firstoftwo
  ]{#1}{#2}{\@gls@field@font{\Glsaccessdescplural{#2}#3}}%
}
\def\@GLSdesc@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@thirdoftwo
   \let\glsifplural\@firstoftwo
  ]%
    {#1}{#2}%
    {\@gls@field@font{\GLSaccessdescplural{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glssymbol@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccesssymbol{#2}#3}}%
}
\def\@Glssymbol@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]%
    {#1}{#2}{\@gls@field@font{\Glsaccesssymbol{#2}#3}}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@thirdoftwo]%
    {#1}{#2}{\@gls@field@font{\GLSaccesssymbol{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glssymbolplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo
   \let\glsifplural\@firstoftwo
  ]{#1}{#2}{\@gls@field@font{\glsaccesssymbolplural{#2}#3}}%
}
\def\@Glssymbolplural@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
```

```
  \@gls@field@link
  [\let\glscapscase\@secondoftwo
   \let\glsifplural\@firstoftwo
  ]{#1}{#2}{\@gls@field@font{\Glsaccesssymbolplural{#2}#3}}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@thirdoftwo
   \let\glsifplural\@firstoftwo
  ]%
    {#1}{#2}%
    {\@gls@field@font{\GLSaccesssymbolplural{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@Glsuseri@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]{#1}{#2}%
  {\@gls@field@font{\Glsentryuseri{#2}#3}}%
}
\def\@GLSuseri@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@thirdoftwo]%
    {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}}%
}
\def\@Glsuserii@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]%
   {#1}{#2}{\@gls@field@font{\Glsentryuserii{#2}#3}}%
}
\def\@GLSuserii@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@thirdoftwo]%
    {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}}%
}
\def\@Glsuseriii@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]%
   {#1}{#2}{\@gls@field@font{\Glsentryuseriii{#2}#3}}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@thirdoftwo]%
    {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuseriii{#2}#3}}}%
}
\def\@Glsuseriv@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link
```

```
      [\let\glscapscase\@secondoftwo]%
       {#1}{#2}{\@gls@field@font{\Glsentryuseriv{#2}#3}}%
    }
    \def\@GLSuseriv@#1#2[#3]{%
      \glsxtrassignfieldfont{#2}%
      \@gls@field@link[\let\glscapscase\@thirdoftwo]%
        {#1}{#2}%
        {\@gls@field@font{\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}}%
    }
    \def\@Glsuserv@#1#2[#3]{%
      \glsxtrassignfieldfont{#2}%
      \@gls@field@link
      [\let\glscapscase\@secondoftwo]%
       {#1}{#2}{\@gls@field@font{\Glsentryuserv{#2}#3}}%
    }
    \def\@GLSuserv@#1#2[#3]{%
      \glsxtrassignfieldfont{#2}%
      \@gls@field@link[\let\glscapscase\@thirdoftwo]%
        {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}}%
    }
    \def\@Glsuservi@#1#2[#3]{%
      \glsxtrassignfieldfont{#2}%
      \@gls@field@link
      [\let\glscapscase\@secondoftwo]%
       {#1}{#2}{\@gls@field@font{\Glsentryuservi{#2}#3}}%
    }
    \def\@GLSuservi@#1#2[#3]{%
      \glsxtrassignfieldfont{#2}%
      \@gls@field@link[\let\glscapscase\@thirdoftwo]%
        {#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuservi{#2}#3}}}%
    }
    \newcommand*{\@@glsxtr@base@acrcmd@warn}[2]{%
      \GlossariesExtraWarning{Base acronym command \string#1\space
       should not be used with new abbreviation definitions. Use
       \string#2\space instead}%
    }
    \let\@glsxtr@base@acrcmd\@@glsxtr@base@acrcmd@warn
    \def\@acrshort#1#2[#3]{%
      \@glsxtr@base@acrcmd\acrshort\glsxtrshort
      \glsdoifexists{#2}%
      {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsxtrifwasfirstuse\@secondoftwo
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
          \acronymfont{\glsaccessshort{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
```

```
    }%
    \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \@glsxtr@base@acrcmd\Acrshort\Glsxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccessshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
  \@glsxtr@base@acrcmd\ACRshort\GLSxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsaccessshort{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@acrshortpl#1#2[#3]{%
  \@glsxtr@base@acrcmd\acrshortpl\glsxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsaccessshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
```

```
}
\def\@Acrshortpl#1#2[#3]{%
  \@glsxtr@base@acrcmd\Acrshortpl\Glsxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccessshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRshortpl#1#2[#3]{%
  \@glsxtr@base@acrcmd\ACRshortpl\GLSxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsaccessshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \@glsxtr@base@acrcmd\acrlong\glsxtrlong
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsaccesslong{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
```

```
    \@glsxtr@base@acrcmd\Acrlong\Glsxtrlong
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@secondofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\Glsaccesslong{#2}}#3%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
    \@glsxtr@base@acrcmd\ACRlong\GLSxtrlong
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@thirdofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \mfirstucMakeUppercase{\acronymfont{\glsaccesslong{#2}}#3}%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\def\@acrlongpl#1#2[#3]{%
    \@glsxtr@base@acrcmd\acrlongpl\glsxtrlongpl
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@firstofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\glsaccesslongpl{#2}}#3%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\def\@Acrlongpl#1#2[#3]{%
    \@glsxtr@base@acrcmd\Acrlongpl\Glsxtrlongpl
    \glsdoifexists{#2}%
```

```
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@secondofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\Glsaccesslongpl{#2}}#3%
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\def\@ACRlongpl#1#2[#3]{%
  \@glsxtr@base@acrcmd\ACRlongpl\GLSxtrlongpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsaccesslongpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@acrfull#1#2[#3]{%
  \@glsxtr@base@acrcmd\acrfull\glsxtrfull
  \acrfullfmt{#1}{#2}{#3}%
}
\def\@Acrfull#1#2[#3]{%
  \@glsxtr@base@acrcmd\Acrfull\Glsxtrfull
  \Acrfullfmt{#1}{#2}{#3}%
}
\def\@ACRfull#1#2[#3]{%
  \@glsxtr@base@acrcmd\ACRfull\GLSxtrfull
  \ACRfullfmt{#1}{#2}{#3}%
}
\def\@acrfullpl#1#2[#3]{%
  \@glsxtr@base@acrcmd\acrfullpl\glsxtrfullpl
  \acrfullplfmt{#1}{#2}{#3}%
}
\def\@Acrfullpl#1#2[#3]{%
  \@glsxtr@base@acrcmd\Acrfullpl\Glsxtrfullpl
  \Acrfullplfmt{#1}{#2}{#3}%
}
\def\@ACRfullpl#1#2[#3]{%
```

```
      \@glsxtr@base@acrcmd\ACRfullpl\GLSxtrfullpl
      \ACRfullplfmt{#1}{#2}{#3}%
}
\renewcommand*{\@glsaddkey}[7]{%
    \key@ifundefined{glossentry}{#1}%
    {%
        \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
        \appto\@gls@keymap{,{#1}{#1}}%
        \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
        \appto\@newglossaryentryposthook{%
            \letcs{\@glo@tmp}{@glo@#1}%
            \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
        }%
        \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
        \newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
        \ifcsdef{@gls@user@#1@}%
        {%
            \PackageError{glossaries}%
            {Can't define '\string#5' as helper command
             '\expandafter\string\csname @gls@user@#1@\endcsname' already
              exists}%
            {}%
        }%
        {%
            \expandafter\newcommand\expandafter*\expandafter
              {\csname @gls@user@#1\endcsname}[2][]{%
                \new@ifnextchar[%
                  {\csuse{@gls@user@#1@}{##1}{##2}}%
                  {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
            \csdef{@gls@user@#1@}##1##2[##3]{%
                \@gls@field@link{##1}{##2}{#3{##2}##3}%
            }%
            \newrobustcmd*{#5}{%
                \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
        }%
        \ifcsdef{@Gls@user@#1@}%
        {%
            \PackageError{glossaries}%
            {Can't define '\string#6' as helper command
             '\expandafter\string\csname @Gls@user@#1@\endcsname' already
              exists}%
            {}%
        }%
        {%
            \expandafter\newcommand\expandafter*\expandafter
              {\csname @Gls@user@#1\endcsname}[2][]{%
                \new@ifnextchar[%
                  {\csuse{@Gls@user@#1@}{##1}{##2}}%
                  {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
            \csdef{@Gls@user@#1@}##1##2[##3]{%
```

```
        \@gls@field@link[\let\glscapscase\@secondofthree]%
          {##1}{##2}{#4{##2}##3}%
      }%
      \newrobustcmd*{#6}{%
        \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
    }%
    \ifcsdef{@GLS@user@#1@}%
    {%
        \PackageError{glossaries}%
        {Can't define '\string#7' as helper command
         '\expandafter\string\csname @GLS@user@#1@\endcsname' already
         exists}%
        {}%
    }%
    {%
        \expandafter\newcommand\expandafter*\expandafter
          {\csname @GLS@user@#1@\endcsname}[2][]{%
            \new@ifnextchar[%
              {\csuse{@GLS@user@#1@}{##1}{##2}}%
              {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
        \csdef{@GLS@user@#1@}##1##2[##3]{%
          \@gls@field@link[\let\glscapscase\@thirdofthree]%
            {##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
        }%
        \newrobustcmd*{#7}{%
          \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
    }%
  }%
  {%
    \PackageError{glossaries-extra}{Key '#1' already exists}{}%
  }%
}
\providecommand*{\@gls@link@nocheckfirsthyper}{}
\let\@glsxtr@org@checkfirsthyper\@gls@link@checkfirsthyper
\renewcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
   {\let\glsxtrifwasfirstuse\@secondoftwo}
   {\let\glsxtrifwasfirstuse\@firstoftwo}%
  \protected@edef\glscategorylabel{\glscategory{\glslabel}}%
  \ifglsused{\glslabel}%
  {%
    \glsifcategoryattribute{\glscategorylabel}{nohypernext}{true}%
      {\KV@glslink@hyperfalse}{}%
  }%
  {%
    \glsifcategoryattribute{\glscategorylabel}{nohyperfirst}{true}%
      {\KV@glslink@hyperfalse}{}%
  }%
  \glslinkcheckfirsthyperhook
}
```

```
\ifdef\do@glsdisablehyperinlist
{%
  \let\@glsxtr@do@glsdisablehyperinlist\do@glsdisablehyperinlist
  \renewcommand*{\do@glsdisablehyperinlist}{%
    \@glsxtr@do@glsdisablehyperinlist
    \glsifattribute{\glslabel}{nohyper}{true}{\KV@glslink@hyperfalse}{}%
  }
}
{}
\define@boolkey{glslink}{noindex}[true]{}
\KV@glslink@noindexfalse
\providecommand*{\@gls@save@glslocal}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\providecommand*{\@gls@restore@glslocal}{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset\glslocalunset
  \else
    \let\@gls@do@glsunset\glsunset
  \fi
}
\providecommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
\ifdef\@gls@setdefault@glslink@opts
{
  \renewcommand*{\@gls@setdefault@glslink@opts}{%
    \KV@glslink@noindexfalse
    \@glsxtrsetaliasnoindex
  }
}
{
  \newcommand*{\@gls@setdefault@glslink@opts}{%
    \KV@glslink@noindexfalse
    \@glsxtrsetaliasnoindex
  }
  \preto\do@glsdisablehyperinlist{\@gls@setdefault@glslink@opts}
}
\providecommand*{\glsxtrsetaliasnoindex}{%
 \KV@glslink@noindextrue
}
\newcommand*{\@glsxtrsetaliasnoindex}{%
 \ifcsvoid{glo@\glsdetoklabel{\glslabel}@alias}%
 {}%
 {%
   \let\glsxtrindexaliased\@glsxtrindexaliased
   \glsxtrsetaliasnoindex
   \let\glsxtrindexaliased\@no@glsxtrindexaliased
 }%
}
\newcommand{\@glsxtrindexaliased}{%
 \ifKV@glslink@noindex
```

```
 \else
   \begingroup
   \let\@glsnumberformat\@glsxtr@defaultnumberformat
   \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{\glslabel}@counter\endcsname}%
   \glsxtr@saveentrycounter
   \@@do@wrglossary{\glsxtralias{\glslabel}}%
   \endgroup
 \fi
}
\newcommand{\@no@glsxtrindexaliased}{%
  \PackageError{glossaries-extra}{\string\glsxtrindexaliased\space
  not permitted outside definition of \string\glsxtrsetaliasnoindex}%
  {}%
}
\let\glsxtrindexaliased\@no@glsxtrindexaliased
\newcommand*{\GlsXtrSetDefaultGlsOpts}[1]{%
  \renewcommand*{\@gls@setdefault@glslink@opts}{%
    \setkeys{glslink}{#1}%
    \@glsxtrsetaliasnoindex
  }%
}
\newcommand*{\glsxtrifindexing}[2]{%
  \ifKV@glslink@noindex #2\else #1\fi
}
\renewcommand*{\glswriteentry}[2]{%
  \glsxtrifindexing
  {%
   \ifglsindexonlyfirst
     \GlsXtrIfUnusedOrUndefined{#1}
     {#2}%
     {\glsxtrdoautoindexname{#1}{dualindex}}%
   \else
     \glsifattribute{#1}{indexonlyfirst}{true}%
     {%
       \GlsXtrIfUnusedOrUndefined{#1}%
       {#2}%
       {\glsxtrdoautoindexname{#1}{dualindex}}%
     }%
     {#2}%
   \fi
  }%
  {}%
}
\appto\@@do@@wrglossary{\@glsxtr@do@@wrindex
  \glsxtrdowrglossaryhook{\@gls@label}%
}
\appto\gls@noidxglossary{\@glsxtr@do@@wrindex
  \glsxtrdowrglossaryhook{\@gls@label}%
}
\newcommand*{\@glsxtr@do@@wrindex}{%
```

886

```
      \glsxtrdoautoindexname{\@gls@label}{dualindex}%
}
\newcommand*{\glsxtrdowrglossaryhook}[1]{}
\newcommand*{\@gls@alt@hyp@opt}[1]{%
 \let\glslinkvar\@firstofthree
 \let\@gls@hyp@opt@cs#1\relax
 \@ifstar{\s@gls@hyp@opt}%
 {\@ifnextchar+%
   {\@firstoftwo{\p@gls@hyp@opt}}%
   {%
     \expandafter\@ifnextchar\@gls@alt@hyp@opt@char
     {\@firstoftwo{\@alt@gls@hyp@opt}}%
     {#1}%
   }%
 }%
}
\newcommand*{\@alt@gls@hyp@opt}[1][]{%
 \let\glslinkvar\@firstofthree
 \expandafter\@gls@hyp@opt@cs\expandafter[\@gls@alt@hyp@opt@keys,#1]}
\newcommand*{\@gls@alt@hyp@opt@char}{}
\newcommand*{\@gls@alt@hyp@opt@keys}{}
\newcommand*{\GlsXtrSetAltModifier}[2]{%
  \let\@gls@hyp@opt\@gls@alt@hyp@opt
  \ifstrequal{#1}{+}%
  {\PackageError{glossaries-extra}%
   {Can't use '#1' as modifier (it's already in use)}{}}%
  {%
    \ifstrequal{#1}{*}%
    {\PackageError{glossaries-extra}%
     {Can't use '#1' as modifier (it's already in use)}{}}%
    {}%
  }%
  \def\@gls@alt@hyp@opt@char{#1}%
  \def\@gls@alt@hyp@opt@keys{#2}%
  \ifdefequal\@glsxtr@record@setting\@glsxtr@record@setting@off
  {}%
  {%
    \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@altmodifier}[1]{}}%
    \protected@write\@auxout{}{\string\@glsxtr@altmodifier{#1}}%
  }%
}
\let\glsxtr@org@dohyperlink\glsdohyperlink
\ifdef\glsnavhyperlink
{
  \renewcommand*{\glsnavhyperlink}[3][\@glo@type]{%
    \protected@edef\gls@grplabel{#2}\protected@edef\@gls@grptitle{#3}%
    {%
      \let\glsxtrdohyperlink\glsxtr@org@dohyperlink
      \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
    }%
```

```
  }%
}
{}
\ifdef\@@gls@navhypertarget
{}
{%
 \renewcommand*{\glsnavhypertarget}{\protect\@@gls@navhypertarget}
 \newcommand*{\@@gls@navhypertarget}[3][\@glo@type]{%
   \@glsnavhypertarget{#1}{#2}{#3}%
 }
}%
\ifdef\@glsnavhypertarget
{%
  \renewcommand*{\@glsnavhypertarget}[3]{%
    \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
    \@glsxtr@do@org@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
    \ifcsdef{@gls@hypergrouplist@#1}%
    {%
      \letcs\@gls@list{@gls@hypergrouplist@#1}%
      \protected@edef\@gls@thishypernavlabel{#2}%
      \expandafter\DTLifinlist\expandafter{\@gls@thishypernavlabel}\@gls@list{}%
      {%
        \GlossariesWarningNoLine{Navigation panel
          for glossary type '#1'^^Jmissing group '#2'}%
        \gdef\gls@hypergrouprerun{%
          \GlossariesWarningNoLine{Navigation panel
          has changed. Rerun LaTeX}}%
      }%
    }%
    {%
      \GlossariesWarningNoLine{Navigation panel
        for glossary type '#1'^^Jmissing group '#2'}%
      \gdef\gls@hypergrouprerun{%
        \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
    }%
  }%
}
{}
\newcommand*{\glsxtrdohyperlink}[2]{%
 \glshasattribute{\glslabel}{targeturl}%
 {%
   \glshasattribute{\glslabel}{targetname}%
   {%
     \glshasattribute{\glslabel}{targetcategory}%
     {%
       \hyperref{\glsgetattribute{\glslabel}{targeturl}}%
         {\glsgetattribute{\glslabel}{targetcategory}}%
         {\glsgetattribute{\glslabel}{targetname}}%
         {{\glsxtrprotectlinks#2}}%
```

```
      }%
      {%
        \hyperref{\glsgetattribute{\glslabel}{targeturl}}%
          {}%
          {\glsgetattribute{\glslabel}{targetname}}%
          {{\glsxtrprotectlinks#2}}%
      }%
    }%
    {%
      \href{\glsgetattribute{\glslabel}{targeturl}}%
        {{\glsxtrprotectlinks#2}}%
    }%
  }%
  {%
    \glsfieldfetch{\glslabel}{alias}{\gloaliaslabel}%
    \ifdefvoid\gloaliaslabel
    {%
      \glsxtrhyperlink{#1}{{\glsxtrprotectlinks#2}}%
    }%
    {%
      \glsxtrifmulti\gloaliaslabel
      {%
        \letcs\gloaliaslabel{@gls@combined@\gloaliaslabel @main}%
      }%
      {}%
      \glsxtrhyperlink
      {\glolinkprefix\glsdetoklabel{\gloaliaslabel}}%
      {{\glsxtrprotectlinks#2}}%
    }%
  }%
}

\newcommand{\glsxtrhyperlink}[2]{%
  \glsdoshowtarget{#1}{\hyperlink{#1}{#2}}%
}%
\renewrobustcmd*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
 \glsdoifexists{#2}%
 {%
   \def\@glo@label{#2}%
   {\protected@edef\glslabel{#2}%
   \@glslink{\glolinkprefix\glslabel}{#1}}%
 }%
}
\renewcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \def\@glslink{\glsdonohyperlink}%
  \let\@glstarget\@secondoftwo
}
\renewcommand{\glsenablehyper}{%
 \KV@glslink@hypertrue
```

```
 \def\@glslink{\glsxtrdohyperlink}%
 \def\@glstarget{\glsdohypertarget}%
}
\def\glsdonohyperlink#1#2{{\glsxtrprotectlinks #2}}
\ifcsundef{hyperlink}%
{%
  \def\@glslink{\glsdonohyperlink}
}%
{%
  \def\@glslink{\glsxtrdohyperlink}
}
\newcommand*{\glsxtrprotectlinks}{%
  \KV@glslink@hyperfalse
  \KV@glslink@noindextrue
  \let\@gls@\@glsxtr@p@text@
  \let\@Gls@\@Glsxtr@p@text@
  \let\@GLS@\@GLSxtr@p@text@
  \let\@glspl@\@glsxtr@p@plural@
  \let\@Glspl@\@Glsxtr@p@plural@
  \let\@GLSpl@\@GLSxtr@p@plural@
  \let\@glsxtrshort\@glsxtr@p@short@
  \let\@Glsxtrshort\@Glsxtr@p@short@
  \let\@GLSxtrshort\@GLSxtr@p@short@
  \let\@glsxtrlong\@glsxtr@p@long@
  \let\@Glsxtrlong\@Glsxtr@p@long@
  \let\@GLSxtrlong\@GLSxtr@p@long@
  \let\@glsxtrshortpl\@glsxtr@p@shortpl@
  \let\@Glsxtrshortpl\@Glsxtr@p@shortpl@
  \let\@GLSxtrshortpl\@GLSxtr@p@shortpl@
  \let\@glsxtrlongpl\@glsxtr@p@longpl@
  \let\@Glsxtrlongpl\@Glsxtr@p@longpl@
  \let\@GLSxtrlongpl\@GLSxtr@p@longpl@
  \let\@acrshort\@glsxtr@p@acrshort@
  \let\@Acrshort\@Glsxtr@p@acrshort@
  \let\@ACRshort\@GLSxtr@p@acrshort@
  \let\@acrshortpl\@glsxtr@p@acrshortpl@
  \let\@Acrshortpl\@Glsxtr@p@acrshortpl@
  \let\@ACRshortpl\@GLSxtr@p@acrshortpl@
  \let\@acrlong\@glsxtr@p@acrlong@
  \let\@Acrlong\@Glsxtr@p@acrlong@
  \let\@ACRlong\@GLSxtr@p@acrlong@
  \let\@acrlongpl\@glsxtr@p@acrlongpl@
  \let\@Acrlongpl\@Glsxtr@p@acrlongpl@
  \let\@ACRlongpl\@GLSxtr@p@acrlongpl@
}
\def\@glsxtr@p@text@#1#2[#3]{{\@glstext@{#1}{#2}[#3]}}
\def\@Glsxtr@p@text@#1#2[#3]{{\@Glstext@{#1}{#2}[#3]}}
\def\@GLSxtr@p@text@#1#2[#3]{{\@GLStext@{#1}{#2}[#3]}}
\def\@glsxtr@p@plural@#1#2[#3]{{\@glsplural@{#1}{#2}[#3]}}
\def\@Glsxtr@p@plural@#1#2[#3]{{\@Glsplural@{#1}{#2}[#3]}}
```

```
\def\@GLSxtr@p@plural@#1#2[#3]{{\@GLSplural@{#1}{#2}[#3]}}
\def\@glsxtr@p@short@#1#2[#3]{%
 {%
  \glssetabbrvfmt{\glscategory{#2}}%
  \glsabbrvfont{\glsentryshort{#2}}#3%
 }%
}
\def\@Glsxtr@p@short@#1#2[#3]{%
 {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \glsabbrvfont{\Glsentryshort{#2}}#3%
 }%
}
\def\@GLSxtr@p@short@#1#2[#3]{%
  {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \mfirstucMakeUppercase{\glsabbrvfont{\glsentryshort{#2}}#3%
  }%
}
\def\@glsxtr@p@shortpl@#1#2[#3]{%
 {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \glsabbrvfont{\glsentryshortpl{#2}}#3%
 }%
}
\def\@Glsxtr@p@shortpl@#1#2[#3]{%
 {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \glsabbrvfont{\Glsentryshortpl{#2}}#3%
 }%
}
\def\@GLSxtr@p@shortpl@#1#2[#3]{%
  {%
   \glssetabbrvfmt{\glscategory{#2}}%
   \mfirstucMakeUppercase{\glsabbrvfont{\glsentryshortpl{#2}}#3}%
  }%
}
\def\@glsxtr@p@long@#1#2[#3]{{\glsentrylong{#2}#3}}
\def\@Glsxtr@p@long@#1#2[#3]{{\Glsentrylong{#2}#3}}
\def\@GLSxtr@p@long@#1#2[#3]{%
  {\mfirstucMakeUppercase{\glslongfont{\glsentrylong{#2}}#3}}}
\def\@glsxtr@p@longpl@#1#2[#3]{{\glsentrylongpl{#2}#3}}
\def\@Glsxtr@p@longpl@#1#2[#3]{{\glslongfont{\Glsentrylongpl{#2}}#3}}
\def\@GLSxtr@p@longpl@#1#2[#3]{%
  {\mfirstucMakeUppercase{\glslongfont{\glsentrylongpl{#2}}#3}}}
\def\@glsxtr@p@acrshort@#1#2[#3]{{\acronymfont{\glsentryshort{#2}}#3}}
\def\@Glsxtr@p@acrshort@#1#2[#3]{{\acronymfont{\Glsentryshort{#2}}#3}}
\def\@GLSxtr@p@acrshort@#1#2[#3]{%
  {\mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}}}
\def\@glsxtr@p@acrshortpl@#1#2[#3]{{\acronymfont{\glsentryshortpl{#2}}#3}}
```

```
\def\@Glsxtr@p@acrshortpl@#1#2[#3]{{\acronymfont{\Glsentryshortpl{#2}}#3}}
\def\@GLSxtr@p@acrshortpl@#1#2[#3]{%
  {\mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}}}
\def\@glsxtr@p@acrlong@#1#2[#3]{{\glsentrylong{#2}#3}}
\def\@Glsxtr@p@acrlong@#1#2[#3]{{\Glsentrylong{#2}#3}}
\def\@GLSxtr@p@acrlong@#1#2[#3]{%
 {\mfirstucMakeUppercase{\glsentrylong{#2}#3}}}
\def\@glsxtr@p@acrlongpl@#1#2[#3]{{\glsentrylongpl{#2}#3}}
\def\@Glsxtr@p@acrlongpl@#1#2[#3]{{\Glsentrylongpl{#2}#3}}
\def\@GLSxtr@p@acrlongpl@#1#2[#3]{%
 {\mfirstucMakeUppercase{\glsentrylongpl{#2}#3}}}
\newcommand*{\@glsxtrp@opt}{hyper=false,noindex}
\newcommand*{\glsxtrsetpopts}[1]{%
  \renewcommand*{\@glsxtrp@opt}{#1}%
}
\newcommand*{\glossxtrsetpopts}{%
  \glsxtrsetpopts{noindex}%
}
\newrobustcmd*{\@@glsxtrp}[2]{%
  {%
    \let\glspostlinkhook\relax
    \csname#1\expandafter\endcsname\expandafter[\@glsxtrp@opt]{#2}[]%
  }%
}
\newrobustcmd*{\@glsxtrp}[2]{%
  \ifcsdef{gls#1}%
  {%
    \@@glsxtrp{gls#1}{#2}%
  }%
  {%
    \ifcsdef{glsxtr#1}%
    {%
      \@@glsxtrp{glsxtr#1}{#2}%
    }%
    {%
      \PackageError{glossaries-extra}{`#1' not recognised by
        \string\glsxtrp}{}%
    }%
  }%
}
\newrobustcmd*{\@Glsxtrp}[2]{%
  \ifcsdef{Gls#1}%
  {%
    \@@glsxtrp{Gls#1}{#2}%
  }%
  {%
    \ifcsdef{Glsxtr#1}%
    {%
      \@@glsxtrp{Glsxtr#1}{#2}%
    }%
```

892

```
    {%
      \PackageError{glossaries-extra}{'#1' not recognised by
        \string\Glsxtrp}{}%
    }%
  }%
}
\newrobustcmd*{\@GLSxtrp}[2]{%
  \ifcsdef{GLS#1}%
  {%
    \@@glsxtrp{GLS#1}{#2}%
  }%
  {%
    \ifcsdef{GLSxtr#1}%
    {%
      \@@glsxtrp{GLSxtr#1}{#2}%
    }%
    {%
      \PackageError{glossaries-extra}{'#1' not recognised by
        \string\GLSxtrp}{}%
    }%
  }%
}
\newrobustcmd*{\glsxtr@headentry@p}[2]{%
 \glsifattribute{#1}{headuc}{true}%
 {%
    \mfirstucMakeUppercase{\@gls@entry@field{#1}{#2}}%
 }%
 {%
    \@gls@entry@field{#1}{#2}%
 }%
}
\ifdef\texorpdfstring
{
  \newcommand{\glsxtrp}[2]{%
    \protect\NoCaseChange
    {%
      \protect\texorpdfstring
      {%
        \protect\glsxtrifinmark
        {%
          \ifcsdef{glsxtrhead#1}%
          {%
            {\protect\csuse{glsxtrhead#1}{#2}}%
          }%
          {%
            \glsxtr@headentry@p{#2}{#1}%
          }%
        }%
        {%
          \@glsxtrp{#1}{#2}%
```

893

```
          }%
        }%
        {%
          \protect\@gls@entry@field{#2}{#1}%
        }%
      }%
    }
  }
  {
    \newcommand{\glsxtrp}[2]{%
      \protect\NoCaseChange
      {%
        \protect\glsxtrifinmark
        {%
          \ifcsdef{glsxtrhead#1}%
          {%
            {\protect\csuse{glsxtrhead#1}}%
          }%
          {%
            \glsxtr@headentry@p{#2}{#1}%
          }%
        }%
        {%
          \@glsxtrp{#1}{#2}%
        }%
      }%
    }
  }
  \newcommand*{\glsps}{\glsxtrp{short}}
  \newcommand*{\glspt}{\glsxtrp{text}}
  \ifdef\texorpdfstring
  {
    \newcommand{\Glsxtrp}[2]{%
      \protect\NoCaseChange
      {%
        \protect\texorpdfstring
        {%
          \protect\glsxtrifinmark
          {%
            \ifcsdef{Glsxtrhead#1}%
            {%
              {\protect\csuse{Glsxtrhead#1}{#2}}%
            }%
            {%
              \protect\@Gls@entry@field{#2}{#1}%
            }%
          }%
          {%
            \@Glsxtrp{#1}{#2}%
          }%
```

894

```
        }%
        {%
          \protect\@gls@entry@field{#2}{#1}%
        }%
      }%
    }
  }
  {
    \newcommand{\Glsxtrp}[2]{%
      \protect\NoCaseChange
      {%
        \protect\glsxtrifinmark
        {%
          \ifcsdef{Glsxtrhead#1}%
          {%
            {\protect\csuse{Glsxtrhead#1}}%
          }%
          {%
            \protect\@Gls@entry@field{#2}{#1}%
          }%
        }%
        {%
          \@Glsxtrp{#1}{#2}%
        }%
      }%
    }
  }
  \ifdef\texorpdfstring
  {
    \newcommand{\GLSxtrp}[2]{%
      \protect\NoCaseChange
      {%
        \protect\texorpdfstring
        {%
          \protect\glsxtrifinmark
          {%
            \ifcsdef{GLSxtr#1}%
            {%
              {\protect\GLSxtrshort[noindex,hyper=false]{#1}[]}%
            }%
            {%
              \protect\mfirstucMakeUppercase
              {%
                \protect\@gls@entry@field{#2}{#1}%
              }%
            }%
          }%
          {%
            \@GLSxtrp{#1}{#2}%
          }%
```

```
      }%
      {%
        \protect\@gls@entry@field{#2}{#1}%
      }%
    }%
  }
}
{
  \newcommand{\GLSxtrp}[2]{%
    \protect\NoCaseChange
    {%
      \protect\glsxtrifinmark
      {%
        \ifcsdef{GLSxtr#1}%
        {%
          {\protect\GLSxtrshort[noindex,hyper=false]{#1}[]}%
        }%
        {%
          \protect\mfirstucMakeUppercase
          {%
            \protect\@gls@entry@field{#2}{#1}%
          }%
        }%
      }%
      {%
        \@GLSxtrp{#1}{#2}%
      }%
    }%
  }
}
\newcommand*{\@glsxtr@unset}[1]{%
  \@@glsunset{#1}%
  \glsxtrpostunset{#1}%
}%
\let\@glsunset\@glsxtr@unset
\newcommand*{\glsxtrpostunset}[1]{}
\newcommand*{\GlsXtrStartUnsetBuffering}{%
  \@ifstar\s@GlsXtrStartUnsetBuffering\@GlsXtrStartUnsetBuffering
}
\newcommand*{\@GlsXtrStartUnsetBuffering}{%
  \let\@glsxtr@org@unset@buffer\@glsxtr@unset@buffer
  \def\@glsxtr@unset@buffer{}%
  \let\@glsunset\@glsxtrbuffer@unset
}
\newcommand*{\s@GlsXtrStartUnsetBuffering}{%
  \let\@glsxtr@org@unset@buffer\@glsxtr@unset@buffer
  \def\@glsxtr@unset@buffer{}%
  \let\@glsunset\@glsxtrbuffer@nodup@unset
}
\newcommand*{\@glsxtrbuffer@unset}[1]{%
```

```
      \listxadd\@glsxtr@unset@buffer{#1}%
}
\newcommand*{\@glsxtrbuffer@nodup@unset}[1]{%
  \expandafter\ifinlist\expandafter{#1}{\@glsxtr@unset@buffer}{}%
  {\listxadd\@glsxtr@unset@buffer{#1}}%
}
\newcommand*{\GlsXtrStopUnsetBuffering}{%
  \@ifstar\s@GlsXtrStopUnsetBuffering\@GlsXtrStopUnsetBuffering
}
\newcommand*{\@GlsXtrStopUnsetBuffering}{%
  \let\@glsunset\@glsxtr@unset
  \forlistloop\@glsunset\@glsxtr@unset@buffer
  \let\@glsxtr@unset@buffer\@glsxtr@org@unset@buffer
}
\newcommand*{\s@GlsXtrStopUnsetBuffering}{%
  \forlistloop\@glslocalunset\@glsxtr@unset@buffer
  \let\@glsunset\@glsxtr@unset
}
\newcommand*{\GlsXtrDiscardUnsetBuffering}{%
  \let\@glsunset\@glsxtr@unset
  \let\@glsxtr@unset@buffer\@glsxtr@org@unset@buffer
}
\newcommand*{\GlsXtrForUnsetBufferedList}[1]{%
  \forlistloop#1\@glsxtr@unset@buffer
}
\renewcommand*{\@glslocalunset}[1]{%
  \@@glslocalunset{#1}%
  \glsxtrpostlocalunset{#1}%
}%
\newcommand*{\glsxtrpostlocalunset}[1]{}
\renewcommand*{\@glsreset}[1]{%
  \@@glsreset{#1}%
  \glsxtrpostreset{#1}%
}%
\newcommand*{\glsxtrpostreset}[1]{}
\renewcommand*{\@glslocalreset}[1]{%
  \@@glslocalreset{#1}%
  \glsxtrpostlocalreset{#1}%
}%
\newcommand*{\glsxtrpostlocalreset}[1]{}
\newcommand*{\glslocalreseteach}[1]{%
  \gls@ifnotmeasuring
  {%
    \@for\@gls@thislabel:=#1\do{%
      \glsdoifexists{\@gls@thislabel}%
      {%
        \@glslocalreset{\@gls@thislabel}%
      }%
    }%
  }%
```

```
}
\newcommand*{\glslocalunseteach}[1]{%
  \gls@ifnotmeasuring
  {%
    \@for\@gls@thislabel:=#1\do{%
      \glsdoifexists{\@gls@thislabel}%
      {%
        \@glslocalunset{\@gls@thislabel}%
      }%
    }%
  }%
}
\newcommand*{\GlsXtrEnableEntryCounting}[2]{%
  \glsenableentrycount
  \renewcommand*{\gls}{\cgls}%
  \renewcommand*{\Gls}{\cGls}%
  \renewcommand*{\glspl}{\cglspl}%
  \renewcommand*{\Glspl}{\cGlspl}%
  \renewcommand*{\GLS}{\cGLS}%
  \renewcommand*{\GLSpl}{\cGLSpl}%
  \@glsxtr@setentrycountunsetattr{#1}{#2}%
  \let\GlsXtrEnableEntryCounting\@glsxtr@setentrycountunsetattr
  \renewcommand*{\GlsXtrEnableEntryUnitCounting}[3]{%
   \PackageError{glossaries-extra}{\string\GlsXtrEnableEntryUnitCounting\space
    can't be used with \string\GlsXtrEnableEntryCounting}%
   {Use one or other but not both commands}}%
}
\newcommand*{\@glsxtr@setentrycountunsetattr}[2]{%
 \@for\@glsxtr@cat:=#1\do
 {%
   \ifdefempty{\@glsxtr@cat}{}%
   {%
     \glssetcategoryattribute{\@glsxtr@cat}{entrycount}{#2}%
   }%
 }%
}
\renewcommand*{\glsenableentrycount}{%
  \appto\@newglossaryentry@defcounters{\@@newglossaryentry@defcounters}%
  \renewcommand*{\gls@defdocnewglossaryentry}{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
      may only be used in the preamble when entry counting has
      been activated}{If you use \string\glsenableentrycount\space
      you must place all entry definitions in the preamble not in
      the document environment}%
    }%
  }%
  \newcommand*{\glsentrycurrcount}[1]{%
   \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
   {0}{\@gls@entry@field{##1}{currcount}}%
```

898

```
    }%
  \newcommand*{\glsentryprevcount}[1]{%
   \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
   {0}{\@gls@entry@field{##1}{prevcount}}%
  }%
  \let\@glsxtr@entrycount@org@unset\glsxtrpostunset
  \renewcommand*{\glsxtrpostunset}[1]{%
    \@glsxtr@entrycount@org@unset{##1}%
    \@gls@increment@currcount{##1}%
  }%
  \let\@glsxtr@entrycount@org@localunset\glsxtrpostlocalunset
  \renewcommand*{\glsxtrpostlocalunset}[1]{%
    \@glsxtr@entrycount@org@localunset{##1}%
    \@gls@local@increment@currcount{##1}%
  }%
  \let\@glsxtr@entrycount@org@reset\glsxtrpostreset
  \renewcommand*{\glsxtrpostreset}[1]{%
    \@glsxtr@entrycount@org@reset{##1}%
    \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
  }%
  \let\@glsxtr@entrycount@org@localreset\glsxtrpostlocalreset
  \renewcommand*{\glsxtrpostlocalreset}[1]{%
    \@glsxtr@entrycount@org@localreset{##1}%
    \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
  }%
  \let\@cgls@\@@cgls@
  \let\@cglspl@\@@cglspl@
  \let\@cGls@\@@cGls@
  \let\@cGlspl@\@@cGlspl@
  \let\@cGLS@\@@cGLS@
  \let\@cGLSpl@\@@cGLSpl@
  \AtEndDocument{\@gls@write@entrycounts}%
  \renewcommand*{\@gls@entry@count}[2]{%
    \csgdef{glo@\glsdetoklabel{##1}@prevcount}{##2}%
  }%
  \let\glsenableentrycount\relax
  \renewcommand*{\glsenableentryunitcount}{%
    \PackageError{glossaries-extra}{\string\glsenableentryunitcount\space
     can't be used with \string\glsenableentrycount}%
    {Use one or other but not both commands}%
  }%
}
\renewcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
    {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
  \count@=0\relax
  \forallglsentries{\@glsentry}{%
    \glshasattribute{\@glsentry}{entrycount}%
    {%
      \ifglsused{\@glsentry}%
```

```
      {%
        \immediate\write\@auxout
          {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
      }%
      {}%
      \advance\count@ by \@ne
    }%
    {}%
  }%
  \ifnum\count@=0
    \GlossariesExtraWarningNoLine{Entry counting has been enabled
     \MessageBreak with \string\glsenableentrycount\space but the
     \MessageBreak attribute 'entrycount' hasn't
     \MessageBreak been assigned to any of the defined
     \MessageBreak entries}%
  \fi
}
\newcommand*{\glsxtrifcounttrigger}[3]{%
 \glshasattribute{#1}{entrycount}%
 {%
   \ifnum\glsentryprevcount{#1}>\glsgetattribute{#1}{entrycount}\relax
     #3%
   \else
     #2%
   \fi
 }%
 {#3}%
}
\def\@@cgls@#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
  {%
    \cglsformat{#2}{#3}%
    \glsunset{#2}%
  }%
  {%
    \@gls@{#1}{#2}[#3]%
  }%
}%
\def\@@cglspl@#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
  {%
    \cglsplformat{#2}{#3}%
    \glsunset{#2}%
  }%
  {%
    \@glspl@{#1}{#2}[#3]%
  }%
}%
\def\@@cGls@#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
```

900

```
    {%
      \cGlsformat{#2}{#3}%
      \glsunset{#2}%
    }%
    {%
      \@Gls@{#1}{#2}[#3]%
    }%
}%
\def\@@cGlspl@#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
    {%
      \cGlsplformat{#2}{#3}%
      \glsunset{#2}%
    }%
    {%
      \@Glspl@{#1}{#2}[#3]%
    }%
}%
\def\@@cGLS@#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
    {%
      \cGLSformat{#2}{#3}%
      \glsunset{#2}%
    }%
    {%
      \@GLS@{#1}{#2}[#3]%
    }%
}%
\def\@@cGLSpl@#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
    {%
      \cGLSplformat{#2}{#3}%
      \glsunset{#2}%
    }%
    {%
      \@GLSpl@{#1}{#2}[#3]%
    }%
}%
\def\@cgls@#1#2[#3]{\@gls@{#1}{#2}[#3]}
\def\@cGls@#1#2[#3]{\@Gls@{#1}{#2}[#3]}
\def\@cglspl@#1#2[#3]{\@glspl@{#1}{#2}[#3]}
\def\@cGlspl@#1#2[#3]{\@Glspl@{#1}{#2}[#3]}
\newrobustcmd*{\cGLS}{\@gls@hyp@opt\@cGLS}
\newcommand*{\@cGLS}[2][]{%
  \new@ifnextchar[{\@cGLS@{#1}{#2}}{\@cGLS@{#1}{#2}[]}%
}
\def\@cGLS@#1#2[#3]{\@GLS@{#1}{#2}[#3]}
\newcommand*{\cGLSformat}[2]{%
 \expandafter\mfirstucMakeUppercase\expandafter{\cglsformat{#1}{#2}}%
}
```

901

```
\newrobustcmd*{\cGLSpl}{\@gls@hyp@opt\@cGLSpl}
\newcommand*{\@cGLSpl}[2][]{%
  \new@ifnextchar[{\@cGLSpl@{#1}{#2}}{\@cGLSpl@{#1}{#2}[]}%
}
\def\@cGLSpl@#1#2[#3]{\@GLSpl@{#1}{#2}[#3]}
\newcommand*{\cGLSplformat}[2]{%
 \expandafter\mfirstucMakeUppercase\expandafter{\cglsplformat{#1}{#2}}%
}
\renewcommand*{\cglsformat}[2]{%
  \glsifregular{#1
  {\glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}}#2%
}
\renewcommand*{\cGlsformat}[2]{%
  \glsifregular{#1
  {\Glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}}#2%
}
\renewcommand*{\cglsplformat}[2]{%
  \glsifregular{#1
  {\glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}}#2%
}
\renewcommand*{\cGlsplformat}[2]{%
  \glsifregular{#1
  {\Glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}}#2%
}
\newcommand*{\@@newglossaryentry@defunitcounters}{%
  \protected@edef\@glo@countunit{\csuse{@glsxtr@categoryattr@@\@glo@category @unitcount}}%
  \ifdefvoid\@glo@countunit
  {}%
  {%
    \@glsxtr@ifunitcounter{\@glo@countunit}%
    {}%
    {\expandafter\@glsxtr@addunitcounter\expandafter{\@glo@countunit}}%
  }%
}
\newcommand*{\@glsxtr@unitcountlist}{}
\newcommand*{\@glsxtr@addunitcounter}[1]{%
 \listadd{\@glsxtr@unitcountlist}{#1}%
 \ifcsundef{glsxtr@theunit@#1}
 {%
   \ifcsdef{theH#1}%
   {\csdef{glsxtr@theunit@#1}{\csuse{theH#1}}}%
   {\csdef{glsxtr@theunit@#1}{\csuse{the#1}}}%
 }%
 {}%
}
\newcommand*{\@glsxtr@ifunitcounter}[3]{%
```

```
      \xifinlist{#1}{\@glsxtr@unitcountlist}{#2}{#3}%
}
\newcommand*\@glsxtr@currentunitcount[1]{%
 glo@\glsdetoklabel{#1}@currunit@\glsgetattribute{#1}{unitcount}.%
 \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
}
\newcommand*\@glsxtr@previousunitcount[1]{%
 glo@\glsdetoklabel{#1}@prevunit@\glsgetattribute{#1}{unitcount}.%
 \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
}
\newcommand*{\@gls@increment@currunitcount}[1]{%
  \glshasattribute{#1}{unitcount}%
  {%
    \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{#1}}%
    \ifcsundef{\@glsxtr@csname}%
    {%
      \csgdef{\@glsxtr@csname}{1}%
      \listcsxadd
       {glo@\glsdetoklabel{#1}@unitlist}%
       {\glsgetattribute{#1}{unitcount}.%
        \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
       }%
    }%
    {%
      \csxdef{\@glsxtr@csname}%
      {\number\numexpr\csname\@glsxtr@csname\endcsname+1}%
    }%
  }%
  {}%
}
\newcommand*{\@gls@local@increment@currunitcount}[1]{%
  \glshasattribute{#1}{unitcount}%
  {%
    \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{#1}}%
    \ifcsundef{\@glsxtr@csname}%
    {%
      \csdef{\@glsxtr@csname}{1}%
      \listcseadd
       {glo@\glsdetoklabel{#1}@unitlist}%
       {\glsgetattribute{#1}{unitcount}.%
        \csuse{glsxtr@theunit@\glsgetattribute{#1}{unitcount}}%
       }%
    }%
    {%
      \csedef{\@glsxtr@csname}%
      {\number\numexpr\csname\@glsxtr@csname\endcsname+1}%
    }%
  }%
  {}%
}
```

```
\newcommand*{\@glsxtr@currunitcount}[2]{%
 \ifcsundef
 {glo@\glsdetoklabel{#1}@currunit@#2}%
 {0}%
 {\csuse{glo@\glsdetoklabel{#1}@currunit@#2}}%
}%
\newcommand*{\@glsxtr@prevunitcount}[2]{%
 \ifcsundef
 {glo@\glsdetoklabel{#1}@prevunit@#2}%
 {0}%
 {\csuse{glo@\glsdetoklabel{#1}@prevunit@#2}}%
}%
\newcommand*{\glsenableentryunitcount}{%
  \appto\@newglossaryentry@defcounters{\@@newglossaryentry@defunitcounters}%
  \renewcommand*{\gls@defdocnewglossaryentry}{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
      may only be used in the preamble when entry counting has
      been activated}{If you use \string\glsenableentryunitcount\space
      you must place all entry definitions in the preamble not in
      the document environment}%
    }%
  }%
  \newcommand*{\glsentrycurrcount}[1]{%
    \@glsxtr@currunitcount{##1}{\glsgetattribute{##1}{unitcount}.%
     \csuse{glsxtr@theunit@\glsgetattribute{##1}{unitcount}}}%
  }%
  \newcommand*{\glsentryprevcount}[1]{%
    \@glsxtr@prevunitcount{##1}{\glsgetattribute{##1}{unitcount}.%
     \csuse{glsxtr@theunit@\glsgetattribute{##1}{unitcount}}}%
  }%
  \newcommand*{\glsentryprevtotalcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@prevunittotal}%
    {0}%
    {%
      \number\csuse{glo@\glsdetoklabel{##1}@prevunittotal}
    }%
  }%
  \newcommand*{\glsentryprevmaxcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@prevunitmax}%
    {0}%
    {%
      \number\csuse{glo@\glsdetoklabel{##1}@prevunitmax}
    }%
  }%
  \let\@glsxtr@entryunitcount@org@unset\glsxtrpostunset
  \renewcommand*{\glsxtrpostunset}[1]{%
    \@glsxtr@entryunitcount@org@unset{##1}%
    \@gls@increment@currunitcount{##1}%
  }%
```

904

```
\let\@glsxtr@entryunitcount@org@localunset\glsxtrpostlocalunset
\renewcommand*{\glsxtrpostlocalunset}[1]{%
  \@glsxtr@entryunitcount@org@localunset{##1}%
  \@gls@local@increment@currunitcount{##1}%
}%
\let\@glsxtr@entryunitcount@org@reset\glsxtrpostreset
\renewcommand*{\glsxtrpostreset}[1]{%
  \glshasattribute{##1}{unitcount}%
  {%
    \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{##1}}%
    \ifcsundef{\@glsxtr@csname}%
    {}%
    {\csgdef{\@glsxtr@csname}{0}}%
  }%
  {}%
}%
\let\@glsxtr@entryunitcount@org@localreset\glsxtrpostlocalreset
\renewcommand*{\glsxtrpostlocalreset}[1]{%
  \@glsxtr@entryunitcount@org@localreset{##1}%
  \glshasattribute{##1}{unitcount}%
  {%
    \protected@edef\@glsxtr@csname{\@glsxtr@currentunitcount{##1}}%
    \ifcsundef{\@glsxtr@csname}%
    {}%
    {\csdef{\@glsxtr@csname}{0}}%
  }%
  {}%
}%
\let\@cgls@\@@cgls@
\let\@cglspl@\@@cglspl@
\let\@cGls@\@@cGls@
\let\@cGlspl@\@@cGlspl@
\let\@cGLS@\@@cGLS@
\let\@cGLSpl@\@@cGLSpl@
\AtEndDocument{\@gls@write@entryunitcounts}%
\renewcommand*{\@gls@entry@unitcount}[3]{%
  \csgdef{glo@\glsdetoklabel{##1}@prevunit@##3}{##2}%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevunittotal}%
  {\csgdef{glo@\glsdetoklabel{##1}@prevunittotal}{##2}}%
  {%
    \csxdef{glo@\glsdetoklabel{##1}@prevunittotal}{
      \number\numexpr\csuse{glo@\glsdetoklabel{##1}@prevunittotal}+##2}%
  }%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevunitmax}%
  {\csgdef{glo@\glsdetoklabel{##1}@prevunitmax}{##2}}%
  {%
    \ifnum\csuse{glo@\glsdetoklabel{##1}@prevunitmax}<##2
     \csgdef{glo@\glsdetoklabel{##1}@prevunitmax}{##2}%
    \fi
  }%
```

```
    }%
    \let\glsenableentryunitcount\relax
    \renewcommand*{\glsenableentrycount}{%
      \PackageError{glossaries-extra}{\string\glsenableentrycount\space
       can't be used with \string\glsenableentryunitcount}%
      {Use one or other but not both commands}%
    }%
}
\@onlypreamble\glsenableentryunitcount
\newcommand*{\@gls@entry@unitcount}[3]{}
\newcommand*{\@gls@write@entryunitcounts@do}[1]{%
  \immediate\write\@auxout
    {\string\@gls@entry@unitcount
      {\@glsentry}%
      {\@glsxtr@currunitcount{\@glsentry}{#1}%
      }%
      {#1}}%
}
\newcommand*{\@gls@write@entryunitcounts}{%
  \immediate\write\@auxout
    {\string\providecommand*{\string\@gls@entry@unitcount}[3]{}}%
  \count@=0\relax
  \forallglsentries{\@glsentry}{%
    \glshasattribute{\@glsentry}{unitcount}%
    {%
      \ifglsused{\@glsentry}%
      {%
        \forlistcsloop
          {\@gls@write@entryunitcounts@do}%
          {glo@\glsdetoklabel{\@glsentry}@unitlist}%
      }%
      {}%
      \advance\count@ by \@ne
    }%
    {}%
  }%
  \ifnum\count@=0
    \GlossariesExtraWarningNoLine{Entry counting has been enabled
     \MessageBreak with \string\glsenableentryunitcount\space but the
     \MessageBreak attribute `unitcount' hasn't
     \MessageBreak been assigned to any of the defined
     \MessageBreak entries}%
  \fi
}
\newcommand*{\GlsXtrEnableEntryUnitCounting}[3]{%
  \glsenableentryunitcount
  \renewcommand*{\gls}{\cgls}%
  \renewcommand*{\Gls}{\cGls}%
  \renewcommand*{\glspl}{\cglspl}%
  \renewcommand*{\Glspl}{\cGlspl}%
```

```
  \renewcommand*{\GLS}{\cGLS}%
  \renewcommand*{\GLSpl}{\cGLSpl}%
  \@glsxtr@setentryunitcountunsetattr{#1}{#2}{#3}%
  \let\GlsXtrEnableEntryUnitCounting\@glsxtr@setentryunitcountunsetattr
  \renewcommand*{\GlsXtrEnableEntryCounting}[2]{%
   \PackageError{glossaries-extra}{\string\GlsXtrEnableEntryCounting\space
    can't be used with \string\GlsXtrEnableEntryUnitCounting}%
   {Use one or other but not both commands}}%
}
\newcommand*{\@glsxtr@setentryunitcountunsetattr}[3]{%
 \@for\@glsxtr@cat:=#1\do
 {%
   \ifdefempty{\@glsxtr@cat}{}%
   {%
     \glssetcategoryattribute{\@glsxtr@cat}{entrycount}{#2}%
     \glssetcategoryattribute{\@glsxtr@cat}{unitcount}{#3}%
   }%
 }%
}
\renewcommand*{\SetGenericNewAcronym}{%
  \ifdefequal\@addtoacronymlists\@glsxtr@org@addtoacronynlists
  {}%
  {%
    \GlossariesWarning{\string\SetGenericNewAcronym\space used
     without restoring base acronym functions with
     \string\RestoreAcronyms}%
  }%
  \let\@Gls@entryname\@Gls@acrentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \protected@edef\@do@newglossaryentry{%
      \noexpand\newglossaryentry{\the\glslabeltok}%
      {%
        type=\acronymtype,%
        name={\expandonce{\acronymentry{##2}}},%
        sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
        text={\the\glsshorttok},%
        short={\the\glsshorttok},%
        shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
```

```
          long={\the\glslongtok},%
          longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
          category=acronym,
          \GenericAcronymFields,%
          \the\glskeylisttok
        }%
      }%
      \@do@newglossaryentry
    }%
    \renewcommand*{\acrfullfmt}[3]{%
      \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
    \renewcommand*{\Acrfullfmt}[3]{%
      \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
    \renewcommand*{\ACRfullfmt}[3]{%
      \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}}%
    \renewcommand*{\acrfullplfmt}[3]{%
      \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
    \renewcommand*{\Acrfullplfmt}[3]{%
      \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
    \renewcommand*{\ACRfullplfmt}[3]{%
      \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}}%
    \renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}%
    \renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
    \renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
    \renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\let\@glsxtr@org@setacronymstyle\setacronymstyle
\let\@glsxtr@org@newacronymstyle\newacronymstyle
\let\@glsxtr@acronymlists\@glsacronymlists
\let\@glsxtr@org@addtoacronynlists\@addtoacronymlists
\let\@glsxtr@org@setacronymlists\SetAcronymLists
\newcommand{\@glsxtr@abbrlists}{}
\newcommand*{\forallabbreviationlists}[2]{%
  \@for#1:=\@glsxtr@abbrlists\do{\ifdefempty{#1}{}{#2}}%
}
\newcommand*{\@glsxtr@addabbreviationlist}[1]{%
  \protected@edef\@glo@type{#1}%
  \ifdefempty\@glsxtr@abbrlists
  {\let\@glsxtr@abbrlists\@glo@type}%
  {%
    \ifdefequal\@glsxtr@abbrlists\@glo@type
    {}%
    {%
      \expandafter\DTLifinlist\expandafter{\@glo@type}{\@glsxtr@abbrlists}{}%
      {\protected@eappto\@glsxtr@abbrlists{,\@glo@type}}%
    }%
  }%
}
```

```
\renewcommand*{\forallacronyms}[2]{%
  \@glsxtr@base@acrcmd\forallacronyms\forallabbreviationlists
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*{\MakeAcronymsAbbreviations}{%
   \@for\@gls@type:=\@glsacronymlists\do{%
     \csgdef{gls@\@gls@type @entryfmt}{\glsentryfmt}%
   }%
   \let\@glsxtr@acronymlists\@glsacronymlists
   \let\@glsacronymlists\@empty
   \let\@addtoacronymlists\@gobble
   \let\SetAcronymLists\@gobble
   \let\@glsxtr@base@acrcmd\@@glsxtr@base@acrcmd@warn
   \renewcommand*{\newacronym}[4][]{%
     \glsxtr@newabbreviation{type=\acronymtype,category=acronym,##1}{##2}{##3}{##4}%
   }%
   \renewcommand*{\firstacronymfont}[1]{\glsfirstabbrvfont{##1}}%
   \renewcommand*{\acronymfont}[1]{\glsabbrvfont{##1}}%
   \renewcommand*{\setacronymstyle}[1]{%
     \PackageError{glossaries-extra}{\string\setacronymstyle{##1}
     unavailable.
     Use \string\setabbreviationstyle[acronym]\space instead.
     The original acronym interface can be restored with
     \string\RestoreAcronyms}{}%
   }%
   \renewcommand*{\newacronymstyle}[1]{%
     \GlossariesExtraWarning{New acronym style `##1' won't be
     available unless you restore the original acronym interface with
     \string\RestoreAcronyms}%
     \@glsxtr@org@newacronymstyle{##1}%
   }%
}
\MakeAcronymsAbbreviations
\newcommand*{\RestoreAcronyms}{%
  \let\@glsacronymlists\@glsxtr@acronymlists
  \let\@addtoacronymlists\@glsxtr@org@addtoacronynlists
  \let\SetAcronymLists\@glsxtr@org@setacronymlists
  \let\@glsxtr@base@acrcmd\@gobbletwo
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gls@type}%
  }%
  \SetGenericNewAcronym
  \renewcommand{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand{\acronymfont}[1]{##1}%
  \let\setacronymstyle\@glsxtr@org@setacronymstyle
  \let\newacronymstyle\@glsxtr@org@newacronymstyle
  \renewcommand*\@gls@link@checkfirsthyper{%
    \ifglsused{\glslabel}%
    {\let\glsxtrifwasfirstuse\@secondoftwo}
    {\let\glsxtrifwasfirstuse\@firstoftwo}%
```

```
    \@glsxtr@org@checkfirsthyper
  }
  \glssetcategoryattribute{acronym}{regular}{false}%
  \setacronymstyle{long-short}%
}
\renewcommand*{\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{#1}})}%
  \ifdim\dimen@<\glsacspacemax~\else\space\fi
}
\newcommand*{\glsacspacemax}{3em}
\newcommand*{\@glsxtr@reg@glosslist}{}
\let\@glsxtr@org@makeglossaries\makeglossaries
\providecommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\providecommand{\@domakeglossaries}[1]{#1}
\renewcommand*{\makeglossaries}[1][]{%
 \@domakeglossaries
 {%
   \@glsxtr@if@record@only
   {%
    \PackageError{glossaries-extra}{\string\makeglossaries\space
     not permitted\MessageBreak with record=\@glsxtr@record@setting\space
     package option}%
    {You may only use \string\makeglossaries\space with
     record=off or record=hybrid options}%
   }%
   {%
     \ifblank{#1}%
     {%
        \@glsxtr@org@makeglossaries
        \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
          \let\warn@noprintglossary\@glsxtr@warn@hybrid@noprintgloss
        \fi
     }%
     {%
       \ifx\@glsxtr@record@setting\@glsxtr@record@setting@alsoindex
         \PackageError{glossaries-extra}{\string\makeglossaries[#1]\space
         not permitted\MessageBreak with record=\@glsxtr@record@setting\space package option}%
         {You may only use the hybrid \string\makeglossaries[...]\space with
```

```
    record=off option}%
\else
  \ifdef\@gls@@automake@immediate{\@gls@@automake@immediate}{}%
  \protected@edef\@glsxtr@reg@glosslist{#1}%
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \protected@write\@auxout{}{\string\providecommand
    \string\@glsorder[1]{}}
  \protected@write\@auxout{}{\string\providecommand
    \string\@istfilename[1]{}}
  \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
  \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
  \protected@write\@auxout{}{\string\glsxtr@makeglossaries{#1}}
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
  \@for\@glo@type:=#1\do{%
   \ifdefempty{\@glo@type}{}{\@makeglossary{\@glo@type}}%
  }%
  \renewcommand*\newglossary[4][]{%
  \PackageError{glossaries}{New glossaries
  must be created before \string\makeglossaries}{You need
  to move \string\makeglossaries\space after all your
  \string\newglossary\space commands}}%
   \let\@makeglossary\@gobble
   \renewcommand\makeglossaries[1][]{}%
   \@disable@onlypremakeg
   \let\gls@checkseeallowed\relax
   \renewcommand*{\@do@seeglossary}[2]{%
     \glsdoifexists{##1}%
     {%
       \protected@edef\@gls@label{\glsdetoklabel{##1}}%
       \protected@edef\@gls@type{\csname glo@\@gls@label @type\endcsname}%
       \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
       {\@glsxtr@org@doseeglossary{##1}{##2}}%
       {%
         \@@glsxtrwrglossmark
         \protected@write\@auxout{}{%
           \string\@gls@reference
             {\gls@type}{\@gls@label}{\string\glsseeformat##2{}}%
         }%
       }%
     }%
   }%
   \let\@glsxtr@@do@@wrglossary\@@do@@wrglossary
   \def\@@do@@wrglossary{%
     \protected@edef\@gls@type{\csname glo@\@gls@label @type\endcsname}%
     \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
     {\@glsxtr@@do@@wrglossary}%
     {\gls@noidxglossary}%
   }%
   \let\warn@nomakeglossaries\relax
   \let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
```

```
\renewcommand{\@gls@noref@warn}[1]{%
  \protected@edef\@gls@type{##1}%
  \expandafter\DTLifinlist\expandafter{\@gls@type}{\@glsxtr@reg@glosslist}%
  {%
    \GlossariesExtraWarning{Can't use
       \string\printnoidxglossary[type={\@gls@type}]
       when '\@gls@type' is listed in the optional argument of
       \string\makeglossaries}%
  }%
  {%
    \GlossariesWarning{Empty glossary for
    \string\printnoidxglossary[type={##1}].
    Rerun may be required (or you may have forgotten to use
    commands like \string\gls)}%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
  {\@glsxtr@idx@displaynumberlist{##1}}%
  {\@glsxtr@noidx@displaynumberlist{##1}}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
  {\@glsxtr@idx@entrynumberlist{##1}}%
  {\@glsxtr@noidx@entrynumberlist{##1}}%
}%
\renewcommand*{\glsnumberlistloop}[2]{%
  \expandafter\DTLifinlist\expandafter{##1}{\@glsxtr@reg@glosslist}%
  {%
    \PackageError{glossaries-extra}{\string\glsnumberlistloop\space
      not available for glossary '##1'}{}%
  }%
  {\@glsxtr@noidx@numberlistloop{##1}{##2}}%
}%
\renewcommand*{\glsprestandardsort}[3]{%
  \expandafter\DTLifinlist\expandafter{##2}{\@glsxtr@reg@glosslist}%
  {%
    \glsdosanitizesort
  }%
  {%
    \ifglssanitizesort
     \@gls@noidx@sanitizesort
    \else
     \@gls@noidx@nosanitizesort
    \fi
  }%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries-extra}{Glossary entries must be defined
   in the preamble\MessageBreak when you use the optional argument
```

```
                of \string\makeglossaries}{Either move your definitions to the
                preamble or don't use the optional argument of
                \string\makeglossaries}%
            }%
            \let\@glo@assign@sortkey\@glsxtr@mixed@assign@sortkey
            \renewcommand*{\@printgloss@setsort}{%
              \expandafter\@glsxtr@gettype\expandafter,\@glsxtr@printglossopts,%
                type=\glsdefaulttype,\@end@glsxtr@gettype
              \def\@glo@sorttype{\@glo@default@sorttype}%
            }%
            \ifglsautomake
              \renewcommand*{\@gls@doautomake}{%
                \@for\@gls@type:=\@glsxtr@reg@glosslist\do{%
                  \ifdefempty{\@gls@type}{}{\@gls@automake{\@gls@type}}%
                }%
              }%
            \fi
            \ifdef\@glo@check@sortallowed{\@glo@check@sortallowed\makeglossaries}{}%
          \fi
      }%
    }%
 }%
}
\ifdef\@printgloss@checkexists
{\newcommand{\glsxtr@printgloss@checkexists}{\@printgloss@checkexists}}
{\newcommand{\glsxtr@printgloss@checkexists}[2]{#2}}
\newcommand{\@glsxtr@orgprintglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{%
      \ifcsdef{@glotype@\@glo@type @title}%
      {\csuse{@glotype@\@glo@type @title}}%
      {\glossaryname}}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '\@glo@type'. \MessageBreak
        Using deprecated fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoctitle{\glssettoctitle{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \bgroup
    \@printgloss@setsort
    \setkeys{printgloss}{#1}%
    \ifx\glossarytitle\org@glossarytitle
```

913

```
      \else
        \cslet{@glotype@\@glo@type @title}{\glossarytitle}%
      \fi
      \let\currentglossary\@glo@type
      \let\org@glossaryentrynumbers\glossaryentrynumbers
      \let\glsnonextpages\@glsnonextpages
      \let\glsnextpages\@glsnextpages
      \glsxtractivatenopost
      \gls@dotoctitle
      \@glossarystyle
      \let\gls@org@glossaryentryfield\glossentry
      \let\gls@org@glossarysubentryfield\subglossentry
      \renewcommand{\glossentry}[1]{%
        \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
        \gls@org@glossaryentryfield{##1}%
      }%
      \renewcommand{\subglossentry}[2]{%
        \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
        \gls@org@glossarysubentryfield{##1}{##2}%
      }%
      \@gls@preglossaryhook
      \glsxtr@printgloss@checkexists{\@glo@type}{#2}%
    \egroup
    \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
    \global\let\warn@noprintglossary\relax
}
\newcommand*{\glsxtractivatenopost}{%
  \let\nopostdesc\@nopostdesc
  \let\glsxtrnopostpunc\@glsxtr@nopostpunc
}
\newrobustcmd*{\glsxtrnopostpunc}{}
\newcommand{\@glsxtr@nopostpunc}{%
 \let\@@glsxtr@org@postdescription\glspostdescription
 \ifglsnopostdot
   \renewcommand{\glspostdescription}{%
     \glsnopostdottrue
     \let\glspostdescription\@@glsxtr@org@postdescription
     \let\glsxtrrestorepostpunc\@glsxtr@restore@postpunc
     \glsxtrpostdescription
     \@glsxtr@nopostpunc@postdesc}%
 \else
   \renewcommand{\glspostdescription}{%
     \let\glspostdescription\@@glsxtr@org@postdescription
     \let\glsxtrrestorepostpunc\@glsxtr@restore@postpunc
     \glsxtrpostdescription
     \@glsxtr@nopostpunc@postdesc}%
 \fi
 \glsnopostdotfalse
}
\newcommand*{\@glsxtr@nopostpunc@postdesc}{}
```

```
\newcommand*{\@glsxtr@restore@postpunc}{%
 \def\@glsxtr@nopostpunc@postdesc{%
   \@glsxtr@org@postdescription
   \let\@glsxtr@nopostpunc@postdesc\@empty
   \let\glsxtrrestorepostpunc\@empty
 }%
}
\newcommand*{\glsxtrrestorepostpunc}{}
\renewcommand{\@printglossary}[2]{%
  \def\@glsxtr@printglossopts{#1}%
  \@glsxtr@orgprintglossary{#1}{#2}%
}
\define@choicekey{printgloss}{target}
[\@glsxtr@printglossval\@glsxtr@printglossnr]%
{true,false}[true]%
{%
  \ifcase\@glsxtr@printglossnr
    \def\@glstarget{\glsdohypertarget}%
  \else
    \let\@glstarget\@secondoftwo
  \fi
}
\newcommand{\@glsxtrhypernameprefix}{}
\define@key{printgloss}{targetnameprefix}{%
  \renewcommand{\@glsxtrhypernameprefix}{#1}%
}
\define@key{printgloss}{prefix}{%
  \renewcommand{\glolinkprefix}{#1}%
}
\define@key{printgloss}{label}{%
  \glsxtrsetglossarylabel{#1}%
}
\newcommand{\glsxtrsetglossarylabel}[1]{%
  \renewcommand*{\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{#1}%
  }%
}
\newcount\@glsxtr@leveloffset
\define@key{printgloss}{leveloffset}{%
  \@glsxtr@assign@leveloffset#1\relax
}
\newcommand*{\@glsxtr@assign@leveloffset}{%
 \@ifnextchar+{\p@glsxtr@assign@leveloffset}{\np@glsxtr@assign@leveloffset}%
}
\newcommand*{\p@glsxtr@assign@leveloffset}[1]{%
 \@ifnextchar+{\pp@glsxtr@assign@leveloffset}{\np@glsxtr@assign@leveloffset}%
}
\def\np@glsxtr@assign@leveloffset#1\relax{\@glsxtr@leveloffset=#1\relax}
\def\pp@glsxtr@assign@leveloffset#1\relax{\advance\@glsxtr@leveloffset by #1\relax}
```

```
\define@boolkey{printgloss}[glsxtr@printgloss@]{groups}[true]{}
\glsxtr@printgloss@groupstrue
\let\@glsxtr@org@glsdohypertarget\glsdohypertarget
\renewcommand{\glsdohypertarget}[2]{%
  \@glsxtr@org@glsdohypertarget{\@glsxtrhypernameprefix#1}{#2}%
}
\ifx\@glstarget\@glsxtr@org@glsdohypertarget
 \def\@glstarget{\glsdohypertarget}%
\fi
\newcommand{\@glsxtr@do@org@target}[2]{%
 {%
  \let\glsdohypertarget\@glsxtr@org@glsdohypertarget
  \@glstarget{#1}{#2}%
 }%
}
\newcommand*{\glsxtr@makeglossaries}[1]{}
\def\@glsxtr@gettype#1,type=#2,#3\@end@glsxtr@gettype{%
  \def\@glo@type{#2}%
}
\newcommand\@glsxtr@mixed@assign@sortkey[1]{%
  \protected@edef\@glo@type{\@glo@type}%
  \expandafter\DTLifinlist\expandafter{\@glo@type}{\@glsxtr@reg@glosslist}%
  {%
    \@glo@no@assign@sortkey{#1}%
  }%
  {%
    \@@glo@assign@sortkey{#1}%
  }%
}%
\let\@glsxtr@idx@displaynumberlist\glsdisplaynumberlist
\newcommand*{\@glsxtr@noidx@displaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{}%
    \def\@gls@noidxloclist@prev{}%
    \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
    \@gls@noidxloclist@finalsep
    \@gls@noidxloclist@prev
  }%
  {%
    \glsxtrundeftag
```

```
      \glsdoifexists{#1}%
      {%
        \GlossariesWarning{Missing location list for '#1'. Either
          a rerun is required or you haven't referenced the entry.}%
      }%
    }%
}%

\newcommand*{\@glsxtr@noidx@numberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc#2\relax
  \let\glsseeformat#3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    \glsxtrundeftag
    \glsdoifexists{#1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry.}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\newcommand*{\@glsxtr@noidx@entrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    \glsxtrundeftag
    \glsdoifexists{#1}%
    {%
      \GlossariesWarning{Missing location list for '#1'. Either
        a rerun is required or you haven't referenced the entry.}%
    }%
  }%
}%
\newcommand*{\@glsxtr@idx@entrynumberlist}[1]{\glsentrynumberlist{#1}}
\renewcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{#1}%
  \ifdefvoid\@glsxtr@titlelabel
  {}%
  {%
```

```
    \protected@edef\@glsxtr@titlelabel{\csuse{glsxtr@grouptitle@#1}}%
  }%
  \ifdefvoid{\@glsxtr@titlelabel}%
  {%
    \DTLifint{#1}%
    {%
      \ifnum#1<256\relax
        \edef#2{\char#1\relax}%
      \else
        \edef#2{#1}%
      \fi
    }%
    {%
      \ifcsundef{#1groupname}%
      {\def#2{#1}}%
      {\letcs#2{#1groupname}}%
    }%
  }%
  {%
    \let#2\@glsxtr@titlelabel
  }%
}
\let\glsxtr@org@getgrouptitle\@gls@getgrouptitle
\newrobustcmd{\glsxtrgetgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
  \@onelevel@sanitize\@glsxtr@titlelabel
  \ifcsdef{\@glsxtr@titlelabel}
  {\letcs{#2}{\@glsxtr@titlelabel}}%
  {\glsxtr@org@getgrouptitle{#1}{#2}}%
}
\let\@gls@getgrouptitle\glsxtrgetgrouptitle
\newcommand{\glsxtrsetgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
  \@onelevel@sanitize\@glsxtr@titlelabel
  \protected@csxdef{\@glsxtr@titlelabel}{#2}%
}
\newcommand{\glsxtrlocalsetgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
  \@onelevel@sanitize\@glsxtr@titlelabel
  \protected@csedef{\@glsxtr@titlelabel}{#2}%
}
\renewcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergrouplist@\@glo@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergrouplist@\@glo@type\endcsname
```

918

```
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \glsxtrgetgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}
\renewcommand*{\@print@noidx@glossary}{%
  \ifcsdef{@glsref@\@glo@type}%
  {%
    \ifcsdef{@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler '\@glo@sorttype'}{}%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{@glsref@\@glo@type}%
    \end{theglossary}%
    \glossarypostamble
  }%
  {%
    \glsxtrifemptyglossary{\@glo@type}%
    {}%
    {\glossarysection[\glossarytoctitle]{\glossarytitle}}%
    \@gls@noref@warn{\@glo@type}%
  }%
}
\renewcommand*{\glsnoidxdisplayloc}[4]{%
  \setentrycounter[#1]{#2}%
  \@glsxtr@display@loc#3\empty\end@glsxtr@display@loc{#4}%
}
\def\@glsxtr@display@loc#1#2\end@glsxtr@display@loc#3{%
  \ifx#1(\relax
    \glsxtrdisplaystartloc{#2}{#3}%
  \else
    \ifx#1)\relax
      \glsxtrdisplayendloc{#2}{#3}%
    \else
      \glsxtrdisplaysingleloc{#1#2}{#3}%
    \fi
  \fi
}
```

```
\newcommand*{\glsxtrdisplaysingleloc}[2]{%
  \csuse{#1}{#2}%
}
\newcommand*{\glsxtrdisplaystartloc}[2]{%
  \protected@edef\glsxtrlocrangefmt{#1}%
  \ifx\glsxtrlocrangefmt\empty
    \def\glsxtrlocrangefmt{glsnumberformat}%
  \fi
  \expandafter\glsxtrdisplaysingleloc
    \expandafter{\glsxtrlocrangefmt}{#2}%
}
\newcommand*{\glsxtrdisplayendloc}[2]{%
  \protected@edef\@glsxtr@tmp{#1}%
  \ifdefempty{\@glsxtr@tmp}{\def\@glsxtr@tmp{glsnumberformat}}{}%
  \ifx\glsxtrlocrangefmt\@glsxtr@tmp
  \else
    \GlossariesExtraWarning{Mismatched end location range
      (start=\glsxtrlocrangefmt, end=\@glsxtr@tmp)}%
  \fi
  \expandafter\glsxtrdisplayendlochook\expandafter{\@glsxtr@tmp}{#2}%
  \expandafter\glsxtrdisplaysingleloc
    \expandafter{\glsxtrlocrangefmt}{#2}%
  \def\glsxtrlocrangefmt{}%
}
\newcommand*{\glsxtrdisplayendlochook}[2]{}
\newcommand*{\glsxtrlocrangefmt}{}
\renewcommand*{\setentrycounter}[2][]{%
  \def\glsxtrcounterprefix{#1}%
  \ifx\glsxtrcounterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\def\@gls@removespaces#1 #2\@nil{%
 \toks@=\expandafter{\the\toks@#1}%
 \ifx\\#2\\%
   \edef\@glo@tmp{\the\toks@}%
   \ifx\@glo@tmp\empty
   \else
    \expandafter\glsxtrlocationhyperlink\expandafter
     \glsentrycounter\expandafter\@glo@counterprefix\expandafter{\the\toks@}%
   \fi
 \else
   \@gls@ReturnAfterFi{%
     \@gls@removespaces#2\@nil
   }%
 \fi
}
```

```
\newcommand*{\glsxtrlocationhyperlink}[3]{%
  \ifdefvoid\glsxtrsupplocationurl
  {%
    \GlsXtrInternalLocationHyperlink{#1}{#2}{#3}%
  }%
  {%
    \hyperref{\glsxtrsupplocationurl}{}{#1#2#3}{#3}%
  }%
}
\newcommand*{\glsxtrsupphypernumber}[1]{%
 {%
    \glshasattribute{\glscurrententrylabel}{externallocation}%
    {%
      \def\glsxtrsupplocationurl{%
        \glsgetattribute{\glscurrententrylabel}{externallocation}}%
    }%
    {%
      \def\glsxtrsupplocationurl{}%
    }%
    \glshypernumber{#1}%
 }%
}
\renewcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  {}%
  {\glsxtrNoGlossaryWarning{\@glo@type}}%
  \ifglsxindy
    \ifcsundef{@xdy@\@glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
        }%
      }%
    }%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
              @language\endcsname}}%
        }%
      }%
```

```
    }%
    \@do@auxoutstuff
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
         \string\providecommand\string\@gls@codepage[2]{}}%
        \noexpand\immediate\noexpand\write\@auxout{%
         \string\@gls@codepage{\@glo@type}{\gls@codepage}}%
      }%
    }%
    \@do@auxoutstuff
  \fi
  \renewcommand*{\@warn@nomakeglossaries}{%
    \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
  }%
}
\newcommand{\GlsXtrNoGlsWarningHead}[2]{%
 This document is incomplete. The external file associated with
 the glossary '#1' (which should be called \texttt{#2})
 hasn't been created.%
}
\newcommand{\GlsXtrNoGlsWarningEmptyStart}{%
  This has probably happened because there are no entries defined
  in this glossary.%
}
\newcommand{\GlsXtrNoGlsWarningEmptyMain}{%
 If you don't want this glossary,
 add \texttt{nomain} to your package option list when you load
 \texttt{glossaries-extra.sty}. For example:%
}
\newcommand{\GlsXtrNoGlsWarningEmptyNotMain}[1]{%
 Did you forget to use \texttt{type=#1} when you defined your
 entries? If you tried to load entries into this glossary with
 \texttt{\string\loadglsentries} did you remember to use
 \texttt{[#1]} as the optional argument? If you did, check that
 the definitions in the file you loaded all had the type set
 to \texttt{\string\glsdefaulttype}.%
}
\newcommand{\GlsXtrNoGlsWarningCheckFile}[1]{%
  Check the contents of the file \texttt{#1}. If
  it's empty, that means you haven't indexed any of your entries in this
  glossary (using commands like \texttt{\string\gls} or
  \texttt{\string\glsadd}) so this list can't be generated.
  If the file isn't empty, the document build process hasn't been
  completed.%
}
\newcommand{\GlsXtrNoGlsWarningAutoMake}[1]{%
  You may need to rerun \LaTeX. If you already have, it may be that
  \TeX's shell escape doesn't allow you to run
```

```
  \ifglsxindy xindy\else makeindex\fi. Check the
  transcript file \texttt{\jobname.log}. If the shell escape is
  disabled, try one of the following:

  \begin{itemize}
    \item Run the external (Lua) application:

      \texttt{makeglossaries-lite \string"\jobname\string"}

    \item Run the external (Perl) application:

      \texttt{makeglossaries \string"\jobname\string"}
  \end{itemize}

  Then rerun \LaTeX\ on this document.
  \GlossariesExtraWarning{Rerun required to build the
  glossary '#1' or check TeX's shell escape allows
  you to run \ifglsxindy xindy\else makeindex\fi}%
}
\newcommand{\GlsXtrNoGlsWarningMisMatch}{%
  You need to either replace \texttt{\string\makenoidxglossaries}
  with \texttt{\string\makeglossaries} or replace
  \texttt{\string\printglossary} (or \texttt{\string\printglossaries}) with
  \texttt{\string\printnoidxglossary}
  (or \texttt{\string\printnoidxglossaries}) and then rebuild
  this document.%
}
\newcommand{\GlsXtrNoGlsWarningBuildInfo}{%
  Try one of the following:
  \begin{itemize}
    \item Add \texttt{automake} to your package option list when you load
          \texttt{glossaries-extra.sty}. For example:

          \texttt{\string\usepackage[automake]%
              \glsopenbrace glossaries-extra\glsclosebrace}

    \item Run the external (Lua) application:

          \texttt{makeglossaries-lite.lua \string"\jobname\string"}

    \item Run the external (Perl) application:

          \texttt{makeglossaries \string"\jobname\string"}
  \end{itemize}

  Then rerun \LaTeX\ on this document.%
}
\newcommand{\GlsXtrRecordWarning}[1]{%
 \texttt{\string\printglossary} doesn't work
 with the \texttt{record=\@glsxtr@record@setting} package option
```

```
use\par\texttt{\string\printunsrtglossary[type=#1]}\par
 instead (or change the package option).%
}
\newcommand{\GlsXtrNoGlsWarningTail}{%
 This message will be removed once the problem has been fixed.%
}
\newcommand{\GlsXtrNoGlsWarningNoOut}[1]{%
  The file \texttt{#1} doesn't exist. This most likely means you haven't used
  \texttt{\string\makeglossaries} or you have used
  \texttt{\string\nofiles}. If this is just a draft version of the
  document, you can suppress this message using the
  \texttt{nomissingglstext} package option.%
}
\newcommand*{\@glsxtr@defaultnoglossarywarning}[1]{%
 \glossarysection[\glossarytoctitle]{\glossarytitle}
 \GlsXtrNoGlsWarningHead{#1}{\jobname.\csname @glotype@\@glo@type @in\endcsname}
 \par
 \glsxtrifemptyglossary{#1}%
 {%
    \GlsXtrNoGlsWarningEmptyStart\space
    \ifthenelse{\equal{#1}{main}}{\GlsXtrNoGlsWarningEmptyMain\par
    \medskip
    \noindent\texttt{\string\usepackage[nomain\ifglsacronym ,acronym\fi]%
        \glsopenbrace glossaries-extra\glsclosebrace}
    \medskip
    }%
    {\GlsXtrNoGlsWarningEmptyNotMain{#1}}%
 }%
 {%
   \IfFileExists{\jobname.\csname @glotype@\@glo@type @out\endcsname}
   {%
     \GlsXtrNoGlsWarningCheckFile
       {\jobname.\csname @glotype@\@glo@type @out\endcsname}

     \ifglsautomake

      \GlsXtrNoGlsWarningAutoMake{#1}

     \else

        \ifthenelse{\equal{#1}{main}}%
        {%
          \GlsXtrNoGlsWarningEmptyMain\par
          \medskip
          \noindent\texttt{\string\usepackage[nomain]%
            \glsopenbrace glossaries-extra\glsclosebrace}
          \medskip
        }%
        {}%
```

```
        \ifdefequal\makeglossaries\@no@makeglossaries
        {%
          \GlsXtrNoGlsWarningMisMatch
        }%
        {%
          \GlsXtrNoGlsWarningBuildInfo
        }%
      \fi
    }%
    {%
      \GlsXtrNoGlsWarningNoOut
        {\jobname.\csname @glotype@\@glo@type @out\endcsname}%
    }%
 }%
 \par
 \GlsXtrNoGlsWarningTail
}
\newcommand*{\@glsxtr@record@noglossarywarning}[1]{%
  \GlossariesExtraWarning{\string\printglossary\space doesn't work\MessageBreak
  with record=\@glsxtr@record@setting\space package option\MessageBreak(use
  \string\printunsrtglossary[type=#1])\MessageBreak
  instead (or change the package option)}%
 \glossarysection[\glossarytoctitle]{\glossarytitle}
 \GlsXtrRecordWarning{#1}
 \GlsXtrNoGlsWarningTail
}
\newcommand*{\GlsXtrDefaultResourceOptions}{}
\newcommand*{\glsxtrresourcefile}[2][]{%
  \disable@keys{glossaries-extra.sty}{record}%
  \glsxtr@writefields
  \ifdefempty\GlsXtrDefaultResourceOptions
  {%
    \protected@write\@auxout{\glsxtrresourceinit}%
    {\string\glsxtr@resource{#1}{#2}}%
  }%
  {%
   \protected@write\@auxout{\glsxtrresourceinit}%
    {\string\glsxtr@resource{\GlsXtrDefaultResourceOptions,#1}{#2}}%
  }%
  \let\@glsxtr@org@see@noindex\@gls@see@noindex
  \let\@gls@see@noindex\relax
  \IfFileExists{#2.glstex}%
  {%
    \edef\@bibgls@restoreat{\noexpand\catcode\noexpand`\noexpand\@=\number\catcode`\@}%
    \makeatletter
    \@input{#2.glstex}%
    \@bibgls@restoreat
    \@glsxtr@check@bibgls@nameref
  }%
  {%
```

```
      \GlossariesExtraWarning{No file '#2.glstex'}%
  }%
  \let\@gls@see@noindex\@glsxtr@org@see@noindex
}
\@onlypreamble\glsxtrresourcefile
\newcommand{\@glsxtr@check@bibgls@nameref}{%
  \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
    \ifdef\bibglshrefchar
    {}%
    {%
      \GlossariesExtraWarning{record=nameref requires at least
      version 1.8 of bib2gls}%
    }%
  \fi
  \let\@glsxtr@check@bibgls@nameref\relax
}
\newcommand*{\glsxtrresourceinit}{}
\newcount\glsxtrresourcecount
\newcommand*{\GlsXtrLoadResources}[1][]{%
  \ifnum\glsxtrresourcecount=0\relax
    \glsxtrresourcefile[#1]{\jobname}%
  \else
    \glsxtrresourcefile[#1]{\jobname-\the\glsxtrresourcecount}%
  \fi
  \advance\glsxtrresourcecount by 1\relax
}
\newcommand*{\glsxtr@resource}[2]{}
\newcommand*{\glsxtr@fields}[1]{}
\newcommand*{\glsxtr@texencoding}[1]{}
\newcommand*{\glsxtr@langtag}[1]{}
\newcommand*{\glsxtr@pluralsuffixes}[4]{}
\newcommand*{\glsxtr@shortcutsval}[1]{}
\newcommand*{\glsxtr@linkprefix}[1]{}
\newcommand*{\glsxtr@writefields}{%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@fields}[1]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@resource}[2]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@pluralsuffixes}[4]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@shortcutsval}[1]{}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@linkprefix}[1]{}}%
  \protected@write\@auxout{}{\string\glsxtr@fields{\@gls@keymap}}%
  \protected@write\@auxout{}%
   {\string\providecommand*{\string\glsxtr@record}[5]{}}%
  \ifx\@glsxtr@record@setting\@glsxtr@record@setting@nameref
    \protected@write\@auxout{}%
     {\string\providecommand*{\string\glsxtr@record@nameref}[8]{}}%
```

```
    \fi
    \ifdef\CurrentTrackedLanguageTag
    {%
        \protected@write\@auxout{}{%
          \string\glsxtr@langtag{\CurrentTrackedLanguageTag}}%
    }%
    {}%
    \protected@write\@auxout{}{\string\glsxtr@pluralsuffixes
      {\glspluralsuffix}{\abbrvpluralsuffix}{\acrpluralsuffix}%
      {\glsxtrabbrvpluralsuffix}}%
    \ifdef\inputencodingname
    {%
        \protected@write\@auxout{}{\string\glsxtr@texencoding{\inputencodingname}}%
    }%
    {%
        \@ifpackageloaded{fontspec}%
        {\protected@write\@auxout{}{\string\glsxtr@texencoding{utf8}}}%
        {}%
    }%
    \protected@write\@auxout{}{\string\glsxtr@shortcutsval{\@glsxtr@shortcutsval}}%
    \AtBeginDocument
      {\protected@write\@auxout{}{\string\glsxtr@linkprefix{\glolinkprefix}}}%
    \let\glsxtr@writefields\relax
    \ifglsautomake
      \IfFileExists{\jobname.aux}%
      {\immediate\write18{bib2gls \jobname}}{}%
      \ifx\@gls@doautomake\@gls@doautomake@err
          \let\@gls@doautomake\relax
      \fi
    \fi
    \@glsxtr@if@record@only
    {\ifdefstring{\glsorder}{letter}%
     {\GlossariesExtraWarningNoLine{Package option 'order=letter' isn't
      supported with 'record=\@glsxtr@record@setting'. Use 'break-at=none'
      resource option instead}}%
     {}%
    }%
    {}%
}
\newcommand*{\@gls@doautomake@err}{%
  \PackageError{glossaries}{You must use
  \string\makeglossaries\space with automake=true}
  {%
      Either remove the automake=true setting or
      add \string\makeglossaries\space to your document preamble.%
  }%
}
\newcommand*{\glsxtr@record}[5]{}
\newcommand*{\glsxtr@record@nameref}[8]{}
\newcommand*{\glsxtr@counterrecord}[3]{%
```

```
      \glsxtrfieldlistgadd{#1}{record.#2}{#3}%
}
\newcommand*{\@glsxtr@counterrecordhook}{}
\newcommand*{\GlsXtrRecordCounter}[1]{%
  \@@glsxtr@recordcounter{#1}%
}
\@onlypreamble\GlsXtrRecordCounter
\newcommand*{\@glsxtr@docounterrecord}[1]{%
  \protected@write\@auxout{}{\string\glsxtr@counterrecord
    {\@gls@label}{#1}{\csuse{the#1}}}%
}
\newcommand*{\glsxtrglossentry}[1]{%
  \glsxtrtitleorpdforheading
  {\@glsxtrglossentry{#1}}%
  {\glsentryname{#1}}%
  {\glsxtrheadname{#1}}%
}
\newrobustcmd*{\@glsxtrglossentry}[1]{%
  \glsxtrtitleorpdforheading
  {%
    \glsdoifexists{#1}%
    {%
      \begingroup
        \protected@edef\glscurrententrylabel{\glsdetoklabel{#1}}%
        \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
        \ifglshasparent{#1}%
        {\GlsXtrStandaloneSubEntryItem{#1}}%
        {\glsentryitem{#1}}%
        \GlsXtrStandaloneEntryName{#1}%
      \endgroup
    }%
  }%
  {\glsentryname{#1}}%
  {\glsxtrheadname{#1}}%
}
\newcommand*{\GlsXtrStandaloneEntryName}[1]{%
  \glstarget{#1}{\glossentryname{#1}}%
}
\newcommand{\GlsXtrStandaloneGlossaryType}{\glsentrytype{\glscurrententrylabel}}
\newcommand*{\GlsXtrStandaloneSubEntryItem}[1]{%
  \GlsXtrIfFieldEqNum{level}{#1}{1}{\glssubentryitem{#1}}{}%
}
\newcommand*{\glsxtrglossentryother}[3]{%
  \ifstrempty{#1}%
  {%
    \ifcsdef{glsxtrhead#3}%
    {%
      \glsxtrtitleorpdforheading
      {\@glsxtrglossentryother{#2}{#3}{#1}}%
      {\@gls@entry@field{#2}{#3}}%
```

```
        {\csuse{glsxtrhead#3}{#2}}%
      }%
      {%
        \glsxtrtitleorpdforheading
        {\@glsxtrglossentryother{#2}{#3}{#1}}%
        {\@gls@entry@field{#2}{#3}}%
        {\@gls@entry@field{\NoCaseChange{#2}}{#3}}%
      }%
    }%
    {%
      \glsxtrtitleorpdforheading
      {\@glsxtrglossentryother{#2}{#3}{#1}}%
      {\@gls@entry@field{#2}{#3}}%
      {#1}%
    }%
}
\newrobustcmd*{\@glsxtrglossentryother}[3]{%
  \glsxtrtitleorpdforheading
  {%
    \glsdoifexists{#1}%
    {%
      \begingroup
        \protected@edef\glscurrententrylabel{\glsdetoklabel{#1}}%
        \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
        \ifglshasparent{#1}%
        {\GlsXtrStandaloneSubEntryItem{#1}}%
        {\glsentryitem{#1}}%
        \GlsXtrStandaloneEntryOther{#1}%
      \endgroup
    }%
  }%
  {\@gls@entry@field{#1}{#2}}%
  {#3}%
}
\newcommand*{\GlsXtrStandaloneEntryOther}[2]{%
  \glstarget{#1}{\glossentrynameother{#1}{#2}}%
}
\ifdef\@printgloss@checkexists
{
  \newcommand*{\printunsrtglossary}{%
    \let\@printgloss@checkexists\@printgloss@checkexists@allowignored
    \@ifstar\s@printunsrtglossary\@printunsrtglossary
  }
}
{
  \newcommand*{\printunsrtglossary}{%
    \@ifstar\s@printunsrtglossary\@printunsrtglossary
  }
}
\newcommand*{\@printunsrtglossary}[1][]{%
```

```
    \@printglossary{type=\glsdefaulttype,#1}{\@print@unsrt@glossary}%
}
\newcommand*{\s@printunsrtglossary}[2][]{%
  \begingroup
    #2%
    \@printglossary{type=\glsdefaulttype,#1}{\@print@unsrt@glossary}%
  \endgroup
}
\newcommand*{\printunsrtglossaries}{%
  \forallglossaries{\@@glo@type}{\printunsrtglossary[type=\@@glo@type]}%
}

\newcommand*{\@print@unsrt@glossary}{%
  \glossarysection[\glossarytoctitle]{\glossarytitle}%
  \glossarypreamble
  \glsxtrifemptyglossary{\@glo@type}%
  {%
    \GlossariesExtraWarning{No entries defined in glossary '\@glo@type'}%
  }%
  {%
    \key@ifundefined{glossentry}{group}%
    {\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle}%
    {\let\@gls@getgrouptitle\@glsxtr@unsrt@getgrouptitle}%
    \def\@gls@currentlettergroup{}%
    \def\@glsxtr@doglossary{%
      \begin{theglossary}%
      \glossaryheader
      \glsresetentrylist
    }%
    \expandafter\@for\expandafter\glscurrententrylabel\expandafter
      :\expandafter=\csname glolist@\@glo@type\endcsname\do{%
      \ifdefempty{\glscurrententrylabel}
      {}%
      {%
        \let\glsxtr@process\@firstofone
        \let\printunsrtglossaryskipentry
            \@glsxtr@printunsrtglossaryskipentry
        \printunsrtglossaryentryprocesshook{\glscurrententrylabel}%
        \glsxtr@process
        {%
          \ifglsxtr@printgloss@groups
            \ifglshasparent{\glscurrententrylabel}{}%
            {%
              \@glsxtr@checkgroup\glscurrententrylabel
              \expandafter\appto\expandafter\@glsxtr@doglossary\expandafter
                {\@glsxtr@groupheading}%
            }%
          \fi
          \protected@eappto\@glsxtr@doglossary{%
            \noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}%
```

```
        }%
      }%
    }%
    \appto\@glsxtr@doglossary{\end{theglossary}}%
    \printunsrtglossarypredoglossary
    \@glsxtr@doglossary
  }%
  \glossarypostamble
}
\newcommand*{\printunsrtinnerglossary}[3][]{%
  \begingroup
  \def\@glsxtr@printglossopts{#1}%
  \def\@glo@type{\glsdefaulttype}%
  \setkeys{printgloss}[title,toctitle,style,numberedsection,sort,label]{#1}%
  \let\currentglossary\@glo@type
  #2%
  \@print@unsrt@innerglossary
  #3%
  \endgroup
}
\newenvironment{printunsrtglossarywrap}[1][]%
{%
  \def\@glsxtr@printglossopts{#1}%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glotype@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '\@glo@type'. \MessageBreak
        Using deprecated fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoctitle{\glssettoctitle{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \ifglossaryexists*{\@glo@type}%
  {%
   \ifx\glossarytitle\org@glossarytitle
   \else
     \expandafter\let\csname @glotype@\@glo@type @title\endcsname
                     \glossarytitle
   \fi
   \let\currentglossary\@glo@type
  }%
```

931

```
    {}%
    \let\org@glossaryentrynumbers\glossaryentrynumbers
    \let\glsnonextpages\@glsnonextpages
    \let\glsnextpages\@glsnextpages
    \let\nopostdesc\@nopostdesc
    \gls@dotoctitle
    \@glossarystyle
    \let\gls@org@glossaryentryfield\glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
}%
{%
    \end{theglossary}%
    \glossarypostamble
    \global\let\glossaryentrynumbers\@org@glossaryentrynumbers
    \global\let\warn@noprintglossary\relax
}
\newcommand*{\@print@unsrt@innerglossary}{%
    \glsxtrifemptyglossary{\@glo@type}%
    {%
      \GlossariesExtraWarning{No entries defined in glossary '\@glo@type'}%
    }%
    {%
      \key@ifundefined{glossentry}{group}%
      {\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle}%
      {\let\@gls@getgrouptitle\@glsxtr@unsrt@getgrouptitle}%
      \def\@gls@currentlettergroup{}%
      \def\@glsxtr@doglossary{}%
      \expandafter\@for\expandafter\glscurrententrylabel\expandafter
        :\expandafter=\csname glolist@\@glo@type\endcsname\do{%
        \ifdefempty{\glscurrententrylabel}
        {}%
        {%
          \let\glsxtr@process\@firstofone
          \let\printunsrtglossaryskipentry
              \@glsxtr@printunsrtglossaryskipentry
          \printunsrtglossaryentryprocesshook{\glscurrententrylabel}%
```

```
        \glsxtr@process
        {%
          \ifglsxtr@printgloss@groups
            \ifglshasparent{\glscurrententrylabel}{}%
            {%
              \@glsxtr@checkgroup\glscurrententrylabel
              \expandafter\appto\expandafter\@glsxtr@doglossary\expandafter
                {\@glsxtr@groupheading}%
            }%
          \fi
          \protected@eappto\@glsxtr@doglossary{%
            \noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}%
        }%
      }%
    }%
    \printunsrtglossarypredoglossary
    \@glsxtr@doglossary
  }%
}
\newcommand*{\printunsrtglossaryentryprocesshook}[1]{}
\newcommand*{\printunsrtglossaryskipentry}{%
  \PackageError{glossaries-extra}{\string\printunsrtglossaryskipentry\space
can only be used within \string\printunsrtglossaryentryprocesshook}{}%
}
\newcommand*{\@glsxtr@printunsrtglossaryskipentry}{%
  \let\glsxtr@process\@gobble
}
\newcommand*{\printunsrtglossarypredoglossary}{}
\newcommand{\@printunsrt@glossary@handler}[1]{%
  \protected@xdef\glscurrententrylabel{#1}%
  \printunsrtglossaryhandler\glscurrententrylabel
}
\newcommand{\printunsrtglossaryhandler}[1]{%
  \glsxtrunsrtdo{#1}%
}
\newrobustcmd*{\glsxtriflabelinlist}[4]{%
 \protected@edef\@glsxtr@doiflabelinlist{\noexpand\@gls@ifinlist{#1}{#2}}%
 \@glsxtr@doiflabelinlist{#3}{#4}%
}
\newcommand{\print@op@unsrtglossaryunit}[2][]{%
  \s@printunsrtglossary[type=\glsdefaulttype,#1]{%
    \printunsrtglossaryunitsetup{#2}%
  }%
}
\newcommand*{\printunsrtglossaryunitsetup}[1]{%
  \renewcommand{\printunsrtglossaryhandler}[1]{%
    \glsxtrfieldxifinlist{##1}{record.#1}{\csuse{the#1}}
    {\glsxtrunsrtdo{##1}}%
    {}%
  }%
```

```
  \ifcsundef{theH#1}%
  {%
    \renewcommand*{\@glsxtrhypernameprefix}{record.#1.\csuse{the#1}.\@gobble}%
  }%
  {%
    \renewcommand*{\@glsxtrhypernameprefix}{record.#1.\csuse{theH#1}.\@gobble}%
  }%
  \renewcommand*{\glossarysection}[2][]{}%
  \appto\glossarypostamble{\glspar\medskip\glspar}%
}
\newcommand{\print@noop@unsrtglossaryunit}[2][]{%
  \PackageError{glossaries-extra}{\string\printunsrtglossaryunit\space
   requires the record=only or record=alsoindex package option}{}%
}
\newrobustcmd*{\@glsxtr@unsrt@getgrouptitle}[2]{%
  \protected@edef\@glsxtr@titlelabel{glsxtr@grouptitle@#1}%
  \@onelevel@sanitize\@glsxtr@titlelabel
  \ifcsdef{\@glsxtr@titlelabel}
  {\letcs{#2}{\@glsxtr@titlelabel}}%
  {\def#2{#1}}%
}
\newcommand{\glsxtrunsrtdo}{\@glsxtr@noidx@do}
\newcommand*{\glsxtrgroupfield}{group}
\newcommand*{\@glsxtr@checkgroup}[1]{%
  \def\@glsxtr@groupheading{}%
  \key@ifundefined{glossentry}{group}%
  {%
    \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
  }%
  {%
    \protected@edef\@glo@thislettergrp{%
        \csuse{glo@\glsdetoklabel{#1}@\glsxtrgroupfield}}%
  }%
  \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
  {}%
  {%
    \ifdefempty{\@gls@currentlettergroup}{}%
    {\def\@glsxtr@groupheading{\glsgroupskip}}%
    \protected@eappto\@glsxtr@groupheading{%
      \noexpand\glsgroupheading{\expandonce\@glo@thislettergrp}%
    }%
  }%
  \let\@gls@currentlettergroup\@glo@thislettergrp
}
\newcommand*{\GlsXtrLocationField}{location}
\newcommand{\@glsxtr@noidx@do}[1]{%
  \ifglsentryexists{#1}%
  {%
    \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
```

```
\global\letcs{\@gls@location}{glo@\glsdetoklabel{#1}@\GlsXtrLocationField}%
\gls@level=\numexpr\csuse{glo@\glsdetoklabel{#1}@level}+\@glsxtr@leveloffset\relax
\ifnum\gls@level>0
  \let\@glsxtr@ifischild\@firstoftwo
\else
  \let\@glsxtr@ifischild\@secondoftwo
\fi
\@glsxtr@ifischild
{%
  \ifdefvoid{\@gls@location}%
  {%
    \ifdefvoid{\@gls@loclist}%
    {%
      \expandafter\subglossentry\expandafter{\number\gls@level}{#1}{}%
    }%
    {%
      \expandafter\subglossentry\expandafter{\number\gls@level}{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
  {%
    \expandafter\subglossentry\expandafter
      {\number\gls@level}{#1}{\glossaryentrynumbers{\@gls@location}}%
  }%
}%
{%
  \ifdefvoid{\@gls@location}%
  {%
    \ifdefvoid{\@gls@loclist}
    {%
      \glossentry{#1}{}%
    }%
    {%
      \glossentry{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
  {%
    \glossentry{#1}%
    {%
      \glossaryentrynumbers{\@gls@location}%
    }%
  }%
}%
}%
{}%
```

935

```
}
\newcount\@glsxtrnewgls@inner
\newcommand*{\@glsxtr@providenewgls}{%
  \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@newglslike}[2]{}}%
  \let\@glsxtr@providenewgls\relax
}
\newcommand{\glsxtridentifyglslike}[2]{%
 \ifdefequal\@glsxtr@record@setting\@glsxtr@record@setting@off
 {}%
 {%
   \@glsxtr@providenewgls
   \protected@write\@auxout{}{\string\@glsxtr@newglslike{#1}{\string#2}}%
 }%
}
\newcommand*{\@glsxtrnewgls}[4]{%
  \ifdef{#3}%
  {%
    \PackageError{glossaries-extra}{Command \string#3\space already
defined}{}%
  }%
  {%
    \glsxtridentifyglslike{#2}{#3}%
    \ifcsdef{@#4like@#2}%
    {%
      \advance\@glsxtrnewgls@inner by \@ne
      \def\@glsxtrnewgls@innercsname{@#4like\number\@glsxtrnewgls@inner @#2}%
    }%
    {\def\@glsxtrnewgls@innercsname{@#4like@#2}}%
    \expandafter\newrobustcmd\expandafter*\expandafter
     #3\expandafter{\expandafter\@gls@hyp@opt\csname\@glsxtrnewgls@innercsname\endcsname}%
    \ifstrempty{#1}%
    {%
      \expandafter\newcommand\expandafter*\csname\@glsxtrnewgls@innercsname\endcsname[2][]{%
        \new@ifnextchar[%
          {\csname @#4@\endcsname{##1}{#2##2}}%
          {\csname @#4@\endcsname{##1}{#2##2}[]}%
      }%
    }%
    {%
      \expandafter\newcommand\expandafter*\csname\@glsxtrnewgls@innercsname\endcsname[2][]{%
        \new@ifnextchar[%
          {\csname @#4@\endcsname{#1,##1}{#2##2}}%
          {\csname @#4@\endcsname{#1,##1}{#2##2}[]}%
      }%
    }%
  }%
}
\newrobustcmd*{\glsxtrnewgls}[3][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{gls}%
}
```

```
\newrobustcmd*{\glsxtrnewglslike}[6][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{gls}%
  \@glsxtrnewgls{#1}{#2}{#4}{glspl}%
  \@glsxtrnewgls{#1}{#2}{#5}{Gls}%
  \@glsxtrnewgls{#1}{#2}{#6}{Glspl}%
}
\newrobustcmd*{\glsxtrnewGLSlike}[4][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{GLS}%
  \@glsxtrnewgls{#1}{#2}{#4}{GLSpl}%
}
\newrobustcmd*{\glsxtrnewrgls}[3][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{rgls}%
}
\newrobustcmd*{\glsxtrnewrglslike}[6][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{rgls}%
  \@glsxtrnewgls{#1}{#2}{#4}{rglspl}%
  \@glsxtrnewgls{#1}{#2}{#5}{rGls}%
  \@glsxtrnewgls{#1}{#2}{#6}{rGlspl}%
}
\newrobustcmd*{\glsxtrnewrGLSlike}[4][]{%
  \@glsxtrnewgls{#1}{#2}{#3}{rGLS}%
  \@glsxtrnewgls{#1}{#2}{#4}{rGLSpl}%
}
\newcommand*{\GlsXtrTotalRecordCount}[1]{%
 \ifcsdef{glo@\glsdetoklabel{#1}@recordcount}%
 {\csname glo@\glsdetoklabel{#1}@recordcount\endcsname}%
 {0}%
}
\newcommand*{\GlsXtrRecordCount}[2]{%
 \ifcsdef{glo@\glsdetoklabel{#1}@recordcount.#2}%
 {\csname glo@\glsdetoklabel{#1}@recordcount.#2\endcsname}%
 {0}%
}
\newcommand*{\GlsXtrLocationRecordCount}[3]{%
 \ifcsdef{glo@\glsdetoklabel{#1}@recordcount.#2.\glsxtrdetoklocation{#3}}%
 {\csname glo@\glsdetoklabel{#1}@recordcount.#2.\glsxtrdetoklocation{#3}\endcsname}%
 {0}%
}
\newcommand*{\glsxtrdetoklocation}[1]{#1}
\newcommand*{\glsxtrenablerecordcount}{%
  \renewcommand*{\gls}{\rgls}%
  \renewcommand*{\Gls}{\rGls}%
  \renewcommand*{\glspl}{\rglspl}%
  \renewcommand*{\Glspl}{\rGlspl}%
  \renewcommand*{\GLS}{\rGLS}%
  \renewcommand*{\GLSpl}{\rGLSpl}%
}
\newcommand*{\glsxtrrecordtriggervalue}[1]{%
 \GlsXtrTotalRecordCount{#1}%
}
```

```
\newcommand*{\GlsXtrSetRecordCountAttribute}[2]{%
 \@for\@glsxtr@cat:=#1\do
 {%
    \ifdefempty{\@glsxtr@cat}{}%
    {%
      \glssetcategoryattribute{\@glsxtr@cat}{recordcount}{#2}%
    }%
 }%
}
\newcommand*{\glsxtrifrecordtrigger}[3]{%
 \glshasattribute{#1}{recordcount}%
 {%
    \ifnum\glsxtrrecordtriggervalue{#1}>\glsgetattribute{#1}{recordcount}\relax
     #3%
    \else
     #2%
    \fi
 }%
 {#3}%
}
\newcommand*{\@glsxtr@rglstrigger@record}[3]{%
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \let\@gls@link@label\glslabel
  \def\@glsxtr@thevalue{}%
  \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
  \def\@glsnumberformat{glstriggerrecordformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \def\@glsxtr@thevalue{}%
  \def\@glsxtr@theHvalue{\@glsxtr@thevalue}%
  \@gls@save@glslocal
  \glsxtrinitwrgloss
  \glslinkpresetkeys
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \ifdefempty{\@glsxtr@thevalue}%
  {%
    \@gls@saveentrycounter
  }%
  {%
    \let\theglsentrycounter\@glsxtr@thevalue
    \def\theHglsentrycounter{\@glsxtr@theHvalue}%
  }%
  \glslinkwrcontent
  {%
    \ifglsxtrinitwrglossbefore
      \@do@wrglossary{#2}%
    \fi
    #3%
    \ifglsxtrinitwrglossbefore
```

```
    \else
      \@do@wrglossary{#2}%
    \fi
  }%
  \@gls@restore@glslocal
  \@gls@do@glsunset{#2}%
}
\newcommand*{\glstriggerrecordformat}[1]{}
\newrobustcmd*{\rgls}{\@gls@hyp@opt\@rgls}
\newcommand*{\@rgls}[2][]{%
  \new@ifnextchar[{\@rgls@{#1}{#2}}{\@rgls@{#1}{#2}[]}%
}
\def\@rgls@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rglsformat{#2}{#3}}%
  }%
  {%
    \@gls@{#1}{#2}[#3]%
  }%
}%
\newrobustcmd*{\rglspl}{\@gls@hyp@opt\@rglspl}
\newcommand*{\@rglspl}[2][]{%
  \new@ifnextchar[{\@rglspl@{#1}{#2}}{\@rglspl@{#1}{#2}[]}%
}
\def\@rglspl@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rglsplformat{#2}{#3}}%
  }%
  {%
    \@glspl@{#1}{#2}[#3]%
  }%
}%
\newrobustcmd*{\rGls}{\@gls@hyp@opt\@rGls}
\newcommand*{\@rGls}[2][]{%
  \new@ifnextchar[{\@rGls@{#1}{#2}}{\@rGls@{#1}{#2}[]}%
}
\def\@rGls@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGlsformat{#2}{#3}}%
  }%
  {%
    \@Gls@{#1}{#2}[#3]%
  }%
}%
\newrobustcmd*{\rGlspl}{\@gls@hyp@opt\@rGlspl}
\newcommand*{\@rGlspl}[2][]{%
  \new@ifnextchar[{\@rGlspl@{#1}{#2}}{\@rGlspl@{#1}{#2}[]}%
```

```
}
\def\@rGlspl@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGlsplformat{#2}{#3}}%
  }%
  {%
    \@Glspl@{#1}{#2}[#3]%
  }%
}%
\newrobustcmd*{\rGLS}{\@gls@hyp@opt\@rGLS}
\newcommand*{\@rGLS}[2][]{%
  \new@ifnextchar[{\@rGLS@{#1}{#2}}{\@rGLS@{#1}{#2}[]}%
}
\def\@rGLS@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGLSformat{#2}{#3}}%
  }%
  {%
    \@GLS@{#1}{#2}[#3]%
  }%
}%
\newrobustcmd*{\rGLSpl}{\@gls@hyp@opt\@rGLSpl}
\newcommand*{\@rGLSpl}[2][]{%
  \new@ifnextchar[{\@rGLSpl@{#1}{#2}}{\@rGLSpl@{#1}{#2}[]}%
}
\def\@rGLSpl@#1#2[#3]{%
  \glsxtrifrecordtrigger{#2}%
  {%
    \@glsxtr@rglstrigger@record{#1}{#2}{\rGLSplformat{#2}{#3}}%
  }%
  {%
    \@GLSpl@{#1}{#2}[#3]%
  }%
}%
\newcommand*{\rglsformat}[2]{%
  \glsifregular{#1}
  {\glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}}#2%
}
\newcommand*{\rglsplformat}[2]{%
  \glsifregular{#1}
  {\glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\glsentrylongplural{#1}}{\glsentryfirstplural{#1}}}#2%
}
\newcommand*{\rGlsformat}[2]{%
  \glsifregular{#1}
  {\Glsentryfirst{#1}}%
  {\ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}}#2%
```

```
}
\newcommand*{\rGlsplformat}[2]{%
  \glsifregular{#1}
  {\Glsentryfirstplural{#1}}%
  {\ifglshaslong{#1}{\Glsentrylongplural{#1}}{\Glsentryfirstplural{#1}}}#2%
}
\newcommand*{\rGLSformat}[2]{%
 \expandafter\mfirstucMakeUppercase\expandafter{\rglsformat{#1}{#2}}%
}
\newcommand*{\rGLSplformat}[2]{%
 \expandafter\mfirstucMakeUppercase\expandafter{\rglsplformat{#1}{#2}}%
}
\newcommand{\@glsxtr@do@inc@linkcount}{%
 \glsifattribute{\glslabel}{linkcount}{true}%
 {%
   \ifcsdef{c@glsxtr@linkcount@\glslabel}{}%
   {%
     \newcounter{glsxtr@linkcount@\glslabel}%
     \glshasattribute{\glslabel}{linkcountmaster}%
     {%
       \begingroup
        \edef\@glo@tmp{\endgroup\noexpand\@addtoreset{glsxtr@linkcount@\glslabel}%
          {\glsgetattribute{\glslabel}{linkcountmaster}}}%
        \@glo@tmp
     }%
     {}%
   }%
  \glsxtrinclinkcounter{glsxtr@linkcount@\glslabel}%
 }%
 {}%
}
\newcommand*{\glsxtrinclinkcounter}[1]{\stepcounter{#1}}
\newcommand*{\GlsXtrLinkCounterValue}[1]{%
 \ifcsundef{c@glsxtr@linkcount@#1}{0}{\csname c@glsxtr@linkcount@#1\endcsname}%
}
\newcommand*{\GlsXtrTheLinkCounter}[1]{%
 \ifcsundef{theglsxtr@linkcount@#1}{0}%
 {\csname theglsxtr@linkcount@#1\endcsname}%
}
\newcommand*{\GlsXtrIfLinkCounterDef}[3]{%
 \ifcsundef{theglsxtr@linkcount@#1}{#3}{#2}%
}
\newcommand*{\GlsXtrLinkCounterName}[1]{glsxtr@linkcount@#1}
\newcommand*{\GlsXtrEnableLinkCounting}[2][]{%
 \let\glsxtr@inc@linkcount\@glsxtr@do@inc@linkcount
 \@for\@glsxtr@label:=#2\do
 {%
  \glssetcategoryattribute{\@glsxtr@label}{linkcount}{true}%
  \ifstrempty{#1}{}%
  {%
```

```
      \ifcsundef{c@#1}%
      {\@nocounterr{#1}}%
      {\glssetcategoryattribute{\@glsxtr@label}{linkcountmaster}{#1}}%
   }%
 }%
}
\@onlypreamble\GlsXtrEnableLinkCounting
\@ifpackageloaded{glossaries-accsupp}
{
  \newcommand*{\glsaccessname}[1]{%
    \glsnameaccessdisplay
    {%
      \glsentryname{#1}%
    }%
    {#1}%
  }
  \newcommand*{\Glsaccessname}[1]{%
    \glsnameaccessdisplay
    {%
      \Glsentryname{#1}%
    }%
    {#1}%
  }
  \newcommand*{\GLSaccessname}[1]{%
    \glsnameaccessdisplay
    {%
      \mfirstucMakeUppercase{\glsentryname{#1}}%
    }%
    {#1}%
  }
  \newcommand*{\glsaccesstext}[1]{%
    \glstextaccessdisplay
    {%
      \glsentrytext{#1}%
    }%
    {#1}%
  }
  \newcommand*{\Glsaccesstext}[1]{%
    \glstextaccessdisplay
    {%
      \Glsentrytext{#1}%
    }%
    {#1}%
  }
  \newcommand*{\GLSaccesstext}[1]{%
    \glstextaccessdisplay
    {%
      \mfirstucMakeUppercase{\glsentrytext{#1}}%
    }%
    {#1}%
```

```
}
\newcommand*{\glsaccessplural}[1]{%
  \glspluralaccessdisplay
  {%
    \glsentryplural{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccessplural}[1]{%
  \glspluralaccessdisplay
  {%
    \Glsentryplural{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccessplural}[1]{%
  \glspluralaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentryplural{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccessfirst}[1]{%
  \glsfirstaccessdisplay
  {%
    \glsentryfirst{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccessfirst}[1]{%
  \glsfirstaccessdisplay
  {%
    \Glsentryfirst{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccessfirst}[1]{%
  \glsfirstaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentryfirst{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccessfirstplural}[1]{%
  \glsfirstpluralaccessdisplay
  {%
    \glsentryfirstplural{#1}%
  }%
  {#1}%
}
```

```
\newcommand*{\Glsaccessfirstplural}[1]{%
  \glsfirstpluralaccessdisplay
  {%
    \Glsentryfirstplural{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccessfirstplural}[1]{%
  \glsfirstpluralaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentryfirstplural{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccesssymbol}[1]{%
  \glssymbolaccessdisplay
  {%
    \glsentrysymbol{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccesssymbol}[1]{%
  \glssymbolaccessdisplay
  {%
    \Glsentrysymbol{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccesssymbol}[1]{%
  \glssymbolaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentrysymbol{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccesssymbolplural}[1]{%
  \glssymbolpluralaccessdisplay
  {%
    \glsentrysymbolplural{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccesssymbolplural}[1]{%
  \glssymbolpluralaccessdisplay
  {%
    \Glsentrysymbolplural{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccesssymbolplural}[1]{%
```

```latex
    \glssymbolpluralaccessdisplay
    {%
      \mfirstucMakeUppercase{\glsentrysymbolplural{#1}}%
    }%
    {#1}%
}
\newcommand*{\glsaccessdesc}[1]{%
  \glsdescriptionaccessdisplay
  {%
    \glsentrydesc{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccessdesc}[1]{%
  \glsdescriptionaccessdisplay
  {%
    \Glsentrydesc{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccessdesc}[1]{%
  \glsdescriptionaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentrydesc{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccessdescplural}[1]{%
  \glsdescriptionpluralaccessdisplay
  {%
    \glsentrydescplural{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccessdescplural}[1]{%
  \glsdescriptionpluralaccessdisplay
  {%
    \Glsentrydescplural{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccessdescplural}[1]{%
  \glsdescriptionpluralaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentrydescplural{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccessshort}[1]{%
  \glsshortaccessdisplay
```

```
  {%
    \glsentryshort{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccessshort}[1]{%
  \glsshortaccessdisplay
  {%
    \Glsentryshort{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccessshort}[1]{%
  \glsshortaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentryshort{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccessshortpl}[1]{%
  \glsshortpluralaccessdisplay
  {%
    \glsentryshortpl{#1}%
  }%
  {#1}%
}
\newcommand*{\Glsaccessshortpl}[1]{%
  \glsshortpluralaccessdisplay
  {%
    \Glsentryshortpl{#1}%
  }%
  {#1}%
}
\newcommand*{\GLSaccessshortpl}[1]{%
  \glsshortpluralaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentryshortpl{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccesslong}[1]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}%
}

\newcommand*{\Glsaccesslong}[1]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}%
}
\newcommand*{\GLSaccesslong}[1]{%
  \glslongaccessdisplay
  {%
```

```
    \mfirstucMakeUppercase{\glsentrylong{#1}}%
  }%
  {#1}%
}
\newcommand*{\glsaccesslongpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}%
}


\newcommand*{\Glsaccesslongpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}%
}
\newcommand*{\GLSaccesslongpl}[1]{%
  \glslongpluralaccessdisplay
  {%
    \mfirstucMakeUppercase{\glsentrylongpl{#1}}%
  }%
  {#1}%
}
\define@key{glsxtrabbrv}{access}{%
  \def\@gls@nameaccess{#1}%
}
\define@key{glsxtrabbrv}{textaccess}{%
  \def\@gls@textaccess{#1}%
}
\define@key{glsxtrabbrv}{pluralaccess}{%
  \def\@gls@pluralaccess{#1}%
}
\define@key{glsxtrabbrv}{firstaccess}{%
  \def\@gls@firstaccess{#1}%
}
\define@key{glsxtrabbrv}{firstpluralaccess}{%
  \def\@gls@firstpluralaccess{#1}%
}
\define@key{glsxtrabbrv}{shortaccess}{%
  \def\@gls@shortaccess{#1}%
}
\define@key{glsxtrabbrv}{shortpluralaccess}{%
  \def\@gls@shortaccesspl{#1}%
}
\define@key{glsxtrabbrv}{longaccess}{%
  \def\@gls@longaccess{#1}%
}
\define@key{glsxtrabbrv}{shortlonglaccess}{%
  \def\@gls@longaccesspl{#1}%
}
\newcommand*{\@gls@initaccesskeys}{%
  \def\@gls@nameaccess{}%
  \def\@gls@textaccess{}%
  \def\@gls@pluralaccess{}%
  \def\@gls@firstaccess{}%
```

```
    \def\@gls@firstpluralaccess{}%
    \def\@gls@shortaccess{}%
    \def\@gls@shortaccesspl{}%
    \def\@gls@longaccess{}%
    \def\@gls@longaccesspl{}%
  }
\newcommand*{\@gls@ifaccessattribute@set}[3]{%
  \glsifcategoryattribute{\glscategorylabel}{access#1}{true}%
  {#2}%
  {%
    \glsifcategoryattribute{\glscategorylabel}{access#1}{false}%
    {#3}%
    {%
      \glsifcategoryattribute{\glscategorylabel}{#1}{true}%
      {#2}%
      {#3}%
    }%
  }%
}
  \def\glsdefaultshortaccess#1#2{#1 (#2)}
  \newcommand{\glsxtrassignactualsetup}{%
   \let\@\empty
   \let\emph\@firstofone
   \let\textbf\@firstofone
   \let\textmd\@firstofone
   \let\textit\@firstofone
   \let\textsl\@firstofone
   \let\textsc\@firstofone
   \let\textrm\@firstofone
   \let\textsf\@firstofone
   \let\texttt\@firstofone
  }
  \ifdef\pdfstringdef
  {
     \newcommand{\@gls@assign@actual}{%
      \begingroup
       \glsxtrassignactualsetup
       \pdfstringdef\@gls@actualshort{\glsxtrorgshort}%
       \pdfstringdef\@gls@actuallong{\glsxtrorglong}%
       \pdfstringdef\@gls@actualshortpl{\@gls@shortpl}%
       \pdfstringdef\@gls@actuallongpl{\@gls@longpl}%
      \protected@edef\@gls@tmp{\endgroup
         \def\noexpand\@gls@actualshort{\expandonce\@gls@actualshort}%
         \def\noexpand\@gls@actuallong{\expandonce\@gls@actuallong}%
         \def\noexpand\@gls@actualshortpl{\expandonce\@gls@actualshortpl}%
         \def\noexpand\@gls@actuallongpl{\expandonce\@gls@actuallongpl}%
      }%
      \@gls@tmp
    }
  }
```

```
{
    \newcommand{\@gls@assign@actual}{%
     \begingroup
      \glsxtrassignactualsetup
      \protected@edef\@gls@tmp{\endgroup
        \def\noexpand\@gls@actualshort{\glsxtrorgshort}%
        \def\noexpand\@gls@actuallong{\glsxtrorglong}%
        \def\noexpand\@gls@actualshortpl{\@gls@shortpl}%
        \def\noexpand\@gls@actuallongpl{\@gls@longpl}%
      }%
      \@gls@tmp
    }
}
\newcommand{\@gls@setup@default@access}{%
  \@gls@assign@actual
  \ifdefempty\@gls@shortaccess
  {%
    \@gls@ifaccessattribute@set{insertdots}%
    {%
      \expandafter\@glsxtr@insertdots\expandafter\@gls@actualshort\expandafter
        {\@gls@actualshort}%
    }%
    {}%
    \ifdefempty\@gls@longaccess
    {%
      \protected@edef\@gls@shortaccess{\glsdefaultshortaccess
        {\expandonce\@gls@actuallong}{\expandonce\@gls@actualshort}}%
    }%
    {%
      \protected@edef\@gls@shortaccess{\glsdefaultshortaccess
        {\expandonce\@gls@longaccess}{\expandonce\@gls@actualshort}}%
    }%
    \eappto\ExtraCustomAbbreviationFields{shortaccess={\@gls@shortaccess},}%
    \ifdefempty\@gls@shortaccesspl
    {%
      \@gls@ifaccessattribute@set{aposplural}%
      {%
        \expandafter\def\expandafter\@gls@shortaccesspl\expandafter{%
          \@gls@actualshort'\glsxtrabbrvpluralsuffix}%
      }%
      {%
        \@gls@ifaccessattribute@set{noshortplural}%
        {%
          \let\@gls@shortaccesspl\@gls@shortaccess
        }%
        {%
          \let\@gls@shortaccesspl\@gls@actualshortpl
        }%
      }%
    }%
    \ifdefempty\@gls@longaccesspl
```

949

```
        {%
          \protected@edef\@gls@shortaccesspl{\glsdefaultshortaccess
            {\expandonce\@gls@actuallongpl}{\expandonce\@gls@actualshortpl}}%
        }%
        {%
          \protected@edef\@gls@shortaccesspl{\glsdefaultshortaccess
            {\expandonce\@gls@longaccesspl}{\expandonce\@gls@actualshort}}%
        }%
        \eappto\ExtraCustomAbbreviationFields{shortpluralaccess={\@gls@shortaccesspl},}%
        }%
        {}%
    }%
    {%
      \ifdefempty\@gls@shortaccesspl
      {\let\@gls@shortaccesspl\@gls@shortaccess}%
      {}%
    }%
    \ifdefempty\@gls@nameaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{nameshortaccess}{true}%
      {%
        \eappto\ExtraCustomAbbreviationFields{access={\@gls@shortaccess},}%
      }%
      {}%
    }%
    {}%
    \ifdefempty\@gls@textaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{textshortaccess}{true}%
      {%
        \eappto\ExtraCustomAbbreviationFields{textaccess={\@gls@shortaccess},}%
      }%
      {}%
    }%
    {}%
    \ifdefempty\@gls@pluralaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{textshortaccess}{true}%
      {%
        \eappto\ExtraCustomAbbreviationFields{%
            pluralaccess={\@gls@shortaccesspl},%
        }%
      }%
      {}%
    }%
    {}%
    \ifdefempty\@gls@firstaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{firstshortaccess}{true}%
      {%
```

```
            \eappto\ExtraCustomAbbreviationFields{firstaccess={\@gls@shortaccess},}%
      }%
      {}%
    }%
    {}%
    \ifdefempty\@gls@firstpluralaccess
    {%
      \glsifcategoryattribute{\glscategorylabel}{firstshortaccess}{true}%
      {%
        \eappto\ExtraCustomAbbreviationFields{%
          firstpluralaccess={\@gls@shortaccesspl},%
        }%
      }%
      {}%
    }%
    {}%
}
\newcommand*{\glsxtrprovideaccsuppcmd}[2]{%
  \ifcsundef{glsxtr#1#2accsupp}%
  {\csdef{glsxtr#1#2accsupp}{\glsshortaccsupp}}%
  {}%
}
\newcommand*{\glsxtrAccSuppAbbrSetNoLongAttrs}[1]{%
 \glssetcategoryattribute{#1}{nameshortaccess}{true}%
 \glssetcategoryattribute{#1}{firstshortaccess}{true}%
 \glssetcategoryattribute{#1}{textshortaccess}{true}%
 \glsxtrprovideaccsuppcmd{#1}{name}%
 \glsxtrprovideaccsuppcmd{#1}{first}%
 \glsxtrprovideaccsuppcmd{#1}{firstpl}%
 \glsxtrprovideaccsuppcmd{#1}{text}%
 \glsxtrprovideaccsuppcmd{#1}{plural}%
}
\newcommand*{\glsxtrAccSuppAbbrSetFirstLongAttrs}[1]{%
 \glssetcategoryattribute{#1}{nameshortaccess}{true}%
 \glssetcategoryattribute{#1}{textshortaccess}{true}%
 \glsxtrprovideaccsuppcmd{#1}{name}%
 \glsxtrprovideaccsuppcmd{#1}{text}%
 \glsxtrprovideaccsuppcmd{#1}{plural}%
}
\newcommand*{\glsxtrAccSuppAbbrSetTextShortAttrs}[1]{%
 \glssetcategoryattribute{#1}{textshortaccess}{true}%
 \glsxtrprovideaccsuppcmd{#1}{text}%
 \glsxtrprovideaccsuppcmd{#1}{plural}%
}
\newcommand*{\glsxtrAccSuppAbbrSetNameShortAttrs}[1]{%
 \glssetcategoryattribute{#1}{nameshortaccess}{true}%
 \glsxtrprovideaccsuppcmd{#1}{name}%
}
\newcommand*{\glsxtrAccSuppAbbrSetNameLongAttrs}[1]{%
 \glssetcategoryattribute{#1}{firstshortaccess}{true}%
```

```
    \glssetcategoryattribute{#1}{textshortaccess}{true}%
    \glsxtrprovideaccsuppcmd{#1}{first}%
    \glsxtrprovideaccsuppcmd{#1}{firstpl}%
    \glsxtrprovideaccsuppcmd{#1}{text}%
    \glsxtrprovideaccsuppcmd{#1}{plural}%
  }
}
{
  \newcommand*{\glsaccessname}[1]{\glsentryname{#1}}
  \newcommand*{\Glsaccessname}[1]{\Glsentryname{#1}}
  \newcommand*{\GLSaccessname}[1]{%
   \protect\mfirstucMakeUppercase{\glsentryname{#1}}}
  \newcommand*{\glsaccesstext}[1]{\glsentrytext{#1}}
  \newcommand*{\Glsaccesstext}[1]{\Glsentrytext{#1}}
  \newcommand*{\GLSaccesstext}[1]{%
   \protect\mfirstucMakeUppercase{\glsentrytext{#1}}}
  \newcommand*{\glsaccessplural}[1]{\glsentryplural{#1}}
  \newcommand*{\Glsaccessplural}[1]{\Glsentryplural{#1}}
  \newcommand*{\GLSaccessplural}[1]{%
   \protect\mfirstucMakeUppercase{\glsentryplural{#1}}}
  \newcommand*{\glsaccessfirst}[1]{\glsentryfirst{#1}}
  \newcommand*{\Glsaccessfirst}[1]{\Glsentryfirst{#1}}
  \newcommand*{\GLSaccessfirst}[1]{%
   \protect\mfirstucMakeUppercase{\glsentryfirst{#1}}}
  \newcommand*{\glsaccessfirstplural}[1]{\glsentryfirstplural{#1}}
  \newcommand*{\Glsaccessfirstplural}[1]{\Glsentryfirstplural{#1}}
  \newcommand*{\GLSaccessfirstplural}[1]{%
   \protect\mfirstucMakeUppercase{\glsentryfirstplural{#1}}}
  \newcommand*{\glsaccesssymbol}[1]{\glsentrysymbol{#1}}
  \newcommand*{\Glsaccesssymbol}[1]{\Glsentrysymbol{#1}}
  \newcommand*{\GLSaccesssymbol}[1]{%
   \protect\mfirstucMakeUppercase{\glsentrysymbol{#1}}}
  \newcommand*{\glsaccesssymbolplural}[1]{\glsentrysymbolplural{#1}}
  \newcommand*{\Glsaccesssymbolplural}[1]{\Glsentrysymbolplural{#1}}
  \newcommand*{\GLSaccesssymbolplural}[1]{%
   \protect\mfirstucMakeUppercase{\glsentrysymbolplural{#1}}}
  \newcommand*{\glsaccessdesc}[1]{\glsentrydesc{#1}}
  \newcommand*{\Glsaccessdesc}[1]{\Glsentrydesc{#1}}
  \newcommand*{\GLSaccessdesc}[1]{%
   \protect\mfirstucMakeUppercase{\glsentrydesc{#1}}}
  \newcommand*{\glsaccessdescplural}[1]{\glsentrydescplural{#1}}
  \newcommand*{\Glsaccessdescplural}[1]{\Glsentrydescplural{#1}}
  \newcommand*{\GLSaccessdescplural}[1]{%
   \protect\mfirstucMakeUppercase{\glsentrydescplural{#1}}}
  \newcommand*{\glsaccessshort}[1]{\glsentryshort{#1}}
  \newcommand*{\Glsaccessshort}[1]{\Glsentryshort{#1}}
  \newcommand*{\GLSaccessshort}[1]{%
   \protect\mfirstucMakeUppercase{\glsentryshort{#1}}}
  \newcommand*{\glsaccessshortpl}[1]{\glsentryshortpl{#1}}
  \newcommand*{\Glsaccessshortpl}[1]{\Glsentryshortpl{#1}}
```

```
  \newcommand*{\GLSaccessshortpl}[1]{%
   \protect\mfirstucMakeUppercase{\glsentryshortpl{#1}}}
  \newcommand*{\glsaccesslong}[1]{\glsentrylong{#1}}
  \newcommand*{\Glsaccesslong}[1]{\Glsentrylong{#1}}
  \newcommand*{\GLSaccesslong}[1]{%
   \protect\mfirstucMakeUppercase{\glsentrylong{#1}}}
  \newcommand*{\glsaccesslongpl}[1]{\glsentrylongpl{#1}}
  \newcommand*{\Glsaccesslongpl}[1]{\Glsentrylongpl{#1}}
  \newcommand*{\GLSaccesslongpl}[1]{%
   \protect\mfirstucMakeUppercase{\glsentrylongpl{#1}}}
  \newcommand*{\@gls@initaccesskeys}{}
  \newcommand{\@gls@setup@default@access}{}
  \newcommand*{\glsxtrAccSuppAbbrSetNoLongAttrs}[1]{}
  \newcommand*{\glsxtrAccSuppAbbrSetFirstLongAttrs}[1]{}
  \newcommand*{\glsxtrAccSuppAbbrSetTextShortAttrs}[1]{}
  \newcommand*{\glsxtrAccSuppAbbrSetNameShortAttrs}[1]{}
  \newcommand*{\glsxtrAccSuppAbbrSetNameLongAttrs}[1]{}
}
\glsaddstoragekey{category}{general}{\glscategory}
\newcommand{\glsifcategory}[4]{%
 \ifglsfieldeq{#1}{category}{#2}{#3}{#4}%
}
\newcommand*{\glssetcategoryattribute}[3]{%
  \csdef{@glsxtr@categoryattr@@#1@#2}{#3}%
}
\newcommand*{\glssetcategoriesattribute}[3]{%
  \@for\@gls@thiscatlabel:=#1\do{%
    \csgdef{@glsxtr@categoryattr@@\@gls@thiscatlabel @#2}{#3}%
  }%
}
\newcommand*{\glssetcategoriesattributes}[3]{%
 {%
    \@for\@gls@thisattrlabel:=#2\do{%
      \glssetcategoriesattribute{#1}{\@gls@thisattrlabel}{#3}%
    }%
 }%
}
\newcommand*{\glsgetcategoryattribute}[2]{%
  \csuse{@glsxtr@categoryattr@@#1@#2}%
}
\newcommand*{\glsunsetcategoryattribute}[2]{%
  \csundef{@glsxtr@categoryattr@@#1@#2}%
}
\newcommand*{\glshascategoryattribute}[4]{%
  \ifcsvoid{@glsxtr@categoryattr@@#1@#2}{#4}{#3}%
}
\newcommand*{\glssetattribute}[3]{%
  \glssetcategoryattribute{\glscategory{#1}}{#2}{#3}%
}
\newcommand*{\glsgetattribute}[2]{%
```

```
    \glsgetcategoryattribute{\glscategory{#1}}{#2}%
}
\newcommand*{\glshasattribute}[4]{%
  \ifglsentryexists{#1}%
  {\glshascategoryattribute{\glscategory{#1}}{#2}{#3}{#4}}%
  {#4}%
}
\newcommand{\glsifcategoryattribute}[5]{%
 \ifcsundef{@glsxtr@categoryattr@@#1@#2}%
 {#5}%
 {\ifcsstring{@glsxtr@categoryattr@@#1@#2}{#3}{#4}{#5}}%
}
\newcommand{\glsifattribute}[5]{%
  \ifglsentryexists{#1}%
  {\glsifcategoryattribute{\glscategory{#1}}{#2}{#3}{#4}{#5}}%
  {#5}%
}
\glssetcategoryattribute{general}{regular}{true}
\glssetcategoryattribute{acronym}{regular}{true}
\newcommand*{\glssetregularcategory}[1]{%
 \glssetcategoryattribute{#1}{regular}{true}%
}
\newcommand{\glsifregularcategory}[3]{%
  \glsifcategoryattribute{#1}{regular}{true}{#2}{#3}%
}
\newcommand{\glsifnotregularcategory}[3]{%
  \glsifcategoryattribute{#1}{regular}{false}{#2}{#3}%
}
\newcommand{\glsifregular}[3]{%
  \glsifregularcategory{\glscategory{#1}}{#2}{#3}%
}
\newcommand{\glsifnotregular}[3]{%
  \glsifnotregularcategory{\glscategory{#1}}{#2}{#3}%
}
\newcommand{\glsforeachincategory}[5][\@glo@types]{%
  \forallglossaries[#1]{#3}%
  {%
     \forglsentries[#3]{#4}%
     {%
       \glsifcategory{#4}{#2}{#5}{}%
     }%
  }%
}
\newcommand{\glsforeachwithattribute}[6][\@glo@types]{%
  \forallglossaries[#1]{#4}%
  {%
     \forglsentries[#4]{#5}%
     {%
       \glsifattribute{#5}{#2}{#3}{#6}{}%
     }%
```

```
  }%
}
\ifdef\newterm
{%
  \renewcommand*{\newterm}[2][]{%
    \newglossaryentry{#2}%
    {type={index},category=index,name={#2},%
     description={\glsxtrpostdescription\nopostdesc},#1}%
  }
  \glssetcategoryattribute{index}{regular}{true}
  \newcommand*{\glsxtrpostdescindex}{}
}
{}
\ifdef\printsymbols
{%
  \newcommand*{\glsxtrnewsymbol}[3][]{%
    \newglossaryentry{#2}{name={#3},sort={#2},type=symbols,category=symbol,#1}%
  }
  \glssetcategoryattribute{symbol}{regular}{true}
  \newcommand*{\glsxtrpostdescsymbol}{}
}
{}
\ifdef\printnumbers
{%
\ifdef\printnumbers
  \newcommand*{\glsxtrnewnumber}[3][]{%
    \newglossaryentry{#2}{name={#3},sort={#2},type=numbers,category=number,#1}%
  }
  \glssetcategoryattribute{number}{regular}{true}
  \newcommand*{\glsxtrpostdescnumber}{}
}
{}
\newcommand*{\glsxtrsetcategory}[2]{%
  \@for\@glsxtr@label:=#1\do
  {%
    \glsfieldxdef{\@glsxtr@label}{category}{#2}%
  }%
}
\newcommand*{\glsxtrsetcategoryforall}[2]{%
  \forallglossaries[#1]{\@glsxtr@type}{%
    \forglsentries[\@glsxtr@type]{\@glsxtr@label}%
    {%
      \glsfieldxdef{\@glsxtr@label}{category}{#2}%
    }%
  }%
}
\newcommand*{\glsxtrfieldtitlecase}[2]{%
  \expandafter\glsxtrfieldtitlecasecs\expandafter
    {\csname glo@\glsdetoklabel{#1}@#2\endcsname}%
}
```

```
\ifdef\glscapitalisewords
{
 \newcommand*{\glsxtrfieldtitlecasecs}[1]{%
   \expandafter\glscapitalisewords\expandafter{#1}}
}
{
 \newcommand*{\glsxtrfieldtitlecasecs}[1]{\xcapitalisewords{#1}}
}
\@ifpackageloaded{glossaries-accsupp}
{
   \renewcommand*{\glossentrydesc}[1]{%
     \glsdoifexistsorwarn{#1}%
     {%
       \glssetabbrvfmt{\glscategory{#1}}%
       \glshasattribute{#1}{glossdescfont}%
       {%
         \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossdescfont}}%
         \ifcsdef{\@glsxtr@attrval}%
         {%
           \letcs{\@glsxtr@glossdescfont}{\@glsxtr@attrval}%
         }%
         {%
           \GlossariesExtraWarning{Unknown control sequence name
           `\@glsxtr@attrval' supplied in glossdescfont attribute
           for entry `#1'. Ignoring}%
           \let\@glsxtr@glossdescfont\@firstofone
         }%
       }%
       {\let\@glsxtr@glossdescfont\@firstofone}%
       \glsifattribute{#1}{glossdesc}{firstuc}%
       {%
         \@glsxtr@glossdescfont{\Glsaccessdesc{#1}}%
       }%
       {%
         \glsifattribute{#1}{glossdesc}{title}%
         {%
           \@glsxtr@do@titlecaps@warn
           \glsdescriptionaccessdisplay
           {%
             \@glsxtr@glossdescfont{\glsxtrfieldtitlecase{#1}{desc}}%
           }%
           {#1}%
         }%
         {%
           \@glsxtr@glossdescfont{\glsaccessdesc{#1}}%
         }%
       }%
     }%
   }
}
```

```
{
  \renewcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossdescfont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossdescfont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossdescfont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          '\@glsxtr@attrval' supplied in glossdescfont attribute
          for entry '#1'. Ignoring}%
          \let\@glsxtr@glossdescfont\@firstofone
        }%
      }%
      {\let\@glsxtr@glossdescfont\@firstofone}%
      \glsifattribute{#1}{glossdesc}{firstuc}%
      {%
        \@glsxtr@glossdescfont{\Glsentrydesc{#1}}%
      }%
      {%
        \glsifattribute{#1}{glossdesc}{title}%
        {%
          \@glsxtr@do@titlecaps@warn
          \@glsxtr@glossdescfont{\glsxtrfieldtitlecase{#1}{desc}}%
        }%
        {%
          \@glsxtr@glossdescfont{\glsentrydesc{#1}}%
        }%
      }%
    }%
  }
}
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\glossentryname}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
        }%
```

```
      {%
        \GlossariesExtraWarning{Unknown control sequence name
        '\@glsxtr@attrval' supplied in glossnamefont attribute
        for entry '#1'. Reverting to default \string\glsnamefont}%
        \let\@glsxtr@glossnamefont\glsnamefont
      }%
    }%
    {\let\@glsxtr@glossnamefont\glsnamefont}%
    \glsifattribute{#1}{glossname}{firstuc}%
    {%
      \glsnameaccessdisplay
      {%
        \@glsxtr@glossnamefont{\Glsentryname{#1}}%
      }%
      {#1}%
    }%
    {%
      \glsifattribute{#1}{glossname}{title}%
      {%
        \@glsxtr@do@titlecaps@warn
        \glsnameaccessdisplay
        {%
          \@glsxtr@glossnamefont{\glsxtrfieldtitlecase{#1}{name}}%
        }%
        {#1}%
      }%
      {%
        \glsifattribute{#1}{glossname}{uc}%
        {%
          \glsnameaccessdisplay
          {%
            \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
            \@glsxtr@glossnamefont{\mfirstucMakeUppercase{\glo@name}}%
          }%
          {#1}%
        }%
        {%
          \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
          \glsnameaccessdisplay
          {%
            \expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}%
          }%
          {#1}%
        }%
      }%
    }%
    \glsxtrpostnamehook{#1}%
  }%
  }
}
```

```
{
  \renewcommand*{\glossentryname}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          '\@glsxtr@attrval' supplied in glossnamefont attribute
          for entry '#1'. Reverting to default \string\glsnamefont}%
          \let\@glsxtr@glossnamefont\glsnamefont
        }%
      }%
      {\let\@glsxtr@glossnamefont\glsnamefont}%
      \glsifattribute{#1}{glossname}{firstuc}%
      {%
        \@glsxtr@glossnamefont{\Glsentryname{#1}}%
      }%
      {%
        \glsifattribute{#1}{glossname}{title}%
        {%
          \@glsxtr@do@titlecaps@warn
          \@glsxtr@glossnamefont{\glsxtrfieldtitlecase{#1}{name}}%
        }%
        {%
          \glsifattribute{#1}{glossname}{uc}%
          {%
            \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
            \@glsxtr@glossnamefont{\mfirstucMakeUppercase{\glo@name}}%
          }%
          {%
            \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
            \expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}%
          }%
        }%
      }%
      \glsxtrpostnamehook{#1}%
    }%
  }
}
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\Glossentryname}[1]{%
    \@glsdoifexistsorwarn{#1}%
```

959

```
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          `\@glsxtr@attrval' supplied in glossnamefont attribute
          for entry `#1'. Reverting to default \string\glsnamefont}%
          \let\@glsxtr@glossnamefont\glsnamefont
        }%
      }%
      {\let\@glsxtr@glossnamefont\glsnamefont}%
      \glsnameaccessdisplay
      {%
        \@glsxtr@glossnamefont{\Glsentryname{#1}}%
      }%
      {#1}%
      \glsxtrpostnamehook{#1}%
    }%
  }
}
{
  \renewcommand*{\Glossentryname}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          `\@glsxtr@attrval' supplied in glossnamefont attribute
          for entry `#1'. Reverting to default \string\glsnamefont}%
          \let\@glsxtr@glossnamefont\glsnamefont
        }%
      }%
      {\let\@glsxtr@glossnamefont\glsnamefont}%
      \@glsxtr@glossnamefont{\Glsentryname{#1}}%
      \glsxtrpostnamehook{#1}%
    }%
  }
```

```
}
\newcommand*{\glsxtrpostnamehook}[1]{%
  \let\@glsnumberformat\@glsxtr@defaultnumberformat
  \glsxtrdoautoindexname{#1}{indexname}%
  \glsextrapostnamehook{#1}%
  \csuse{glsxtrpostname\glscategory{#1}}%
}
\newcommand*{\glsextrapostnamehook}[1]{}
\newcommand*{\glsdefpostname}[2]{%
  \csdef{glsxtrpostname#1}{#2}%
}
\@ifpackageloaded{glossaries-accsupp}
{
  \newcommand*{\glsxtr@setaccessdisplay}[1]{%
     \ifcsdef{gls#1accessdisplay}%
     {\letcs\@glsxtr@accessdisplay{gls#1accessdisplay}}%
     {%
       \protected@edef\@gls@thisval{#1}%
       \@for\@gls@map:=\@gls@keymap\do{%
         \protected@edef\@this@key{\expandafter\@secondoftwo\@gls@map}%
         \ifdefequal{\@this@key}{\@gls@thisval}%
         {%
           \protected@edef\@gls@thisval{\expandafter\@firstoftwo\@gls@map}%
           \@endfortrue
         }%
         {}%
       }%
       \ifcsdef{gls\@gls@thisval accessdisplay}%
       {\letcs\@glsxtr@accessdisplay{gls\@gls@thisval accessdisplay}}%
       {\let\@glsxtr@accessdisplay\@firstoftwo}%
     }%
  }
}
{%
  \newcommand*{\glsxtr@setaccessdisplay}[1]{%
   \let\@glsxtr@accessdisplay\@firstoftwo}
}
\newrobustcmd*{\glossentrynameother}[2]{%
  \@glsdoifexistsorwarn{#1}%
  {%
    \glsxtr@setaccessdisplay{#2}%
    \glssetabbrvfmt{\glscategory{#1}}%
    \glshasattribute{#1}{glossnamefont}%
    {%
      \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glossnamefont}}%
      \ifcsdef{\@glsxtr@attrval}%
      {%
        \letcs{\@glsxtr@glossnamefont}{\@glsxtr@attrval}%
      }%
      {%
```

```
      \GlossariesExtraWarning{Unknown control sequence name
      '\@glsxtr@attrval' supplied in glossnamefont attribute
      for entry '#1'. Reverting to default \string\glsnamefont}%
      \let\@glsxtr@glossnamefont\glsnamefont
    }%
  }%
  {\let\@glsxtr@glossnamefont\glsnamefont}%
  \glsifattribute{#1}{glossname}{firstuc}%
  {%
    \@glsxtr@accessdisplay
    {\@glsxtr@glossnamefont{\@Gls@entry@field{#1}{#2}}}%
    {#1}%
  }%
  {%
    \glsifattribute{#1}{glossname}{title}%
    {%
      \@glsxtr@do@titlecaps@warn
      \@glsxtr@accessdisplay
      {\@glsxtr@glossnamefont{\glsxtrfieldtitlecase{#1}{#2}}}%
      {#1}%
    }%
    {%
      \glsifattribute{#1}{glossname}{uc}%
      {%
        \letcs{\glo@name}{glo@\glsdetoklabel{#1}@#2}%
        \@glsxtr@accessdisplay
        {\@glsxtr@glossnamefont{\mfirstucMakeUppercase{\glo@name}}}%
        {#1}%
      }%
      {%
        \letcs{\glo@name}{glo@\glsdetoklabel{#1}@#2}%
        \@glsxtr@accessdisplay
        {\expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}}%
        {#1}%
      }%
    }%
  }%
      \glsxtrpostnamehook{#1}%
  }%
}
\newif\if@glsxtr@format@override
\@glsxtr@format@overridefalse
\@ifpackageloaded{hyperref}
{
  \ifHy@hyperindex
    \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
      \@glsxtr@format@overridetrue
      \appto\theindex{\let\glshypernumber\@firstofone}%
    }
  \else
```

```
    \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
       \@glsxtr@format@overridetrue
       \appto\theindex{\let\glshypernumber\hyperpage}%
    }
  \fi
}
{
  \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
    \@glsxtr@format@overridetrue
  }
}
\@onlypreamble\GlsXtrEnableIndexFormatOverride
\newcommand*{\glsxtrdoautoindexname}[2]{%
  \glshasattribute{#1}{#2}%
  {%
    \@glsxtr@autoindex@setname{#1}%
    \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{#2}}%
    \if@glsxtr@format@override
      \ifx\@glsnumberformat\@glsxtr@defaultnumberformat
      \else
        \let\@glsxtr@attrval\@glsnumberformat
      \fi
    \fi
    \ifdefstring{\@glsxtr@attrval}{true}%
    {}%
    {\protected@eappto\@glo@name{\@glsxtr@autoindex@encap\@glsxtr@attrval}}%
    \expandafter\glsxtrautoindex\expandafter{\@glo@name}%
  }%
  {}%
}
\newcommand*{\glsxtrautoindex}{\index}
\newcommand{\glsxtrautoindexesc}{%
  \@gls@checkmkidxchars\@glo@sort
  \@glsxtr@autoindex@doextra@esc\@glo@sort
}
\newcommand*{\@glsxtr@autoindex@setname}[1]{%
  \protected@edef\@glo@name{\glsxtrautoindexentry{#1}}%
  \glsxtrautoindexassignsort{\@glo@sort}{#1}%
  \glsxtrautoindexesc
  \epreto\@glo@name{\@glo@sort\@glsxtr@autoindex@at}%
}
\newcommand*{\glsxtrautoindexentry}[1]{\string\glsentryname{#1}}
\newcommand*{\glsxtrautoindexassignsort}[2]{%
  \glsletentryfield{#1}{#2}{sort}%
}
\newcommand*{\@glsxtr@autoindex@doextra@esc}[1]{%
  \ifx\@glsxtr@autoindex@esc\@gls@quotechar
  \else
    \def\@gls@checkedmkidx{}%
    \edef\@@glsxtr@checkspch{%
```

```
          \noexpand\@glsxtr@autoindex@escquote\expandonce{#1}%
            \noexpand\@empty\@glsxtr@autoindex@esc\noexpand\@nnil
            \@glsxtr@autoindex@esc\noexpand\@empty\noexpand\@glsxtr@endescspch}%
      \@@glsxtr@checkspch
      \let#1\@gls@checkedmkidx\relax
    \fi
    \ifx\@glsxtr@autoindex@at\@gls@actualchar
    \else
      \def\@gls@checkedmkidx{}%
      \edef\@@glsxtr@checkspch{%
        \noexpand\@glsxtr@autoindex@escat\expandonce{#1}%
          \noexpand\@empty\@glsxtr@autoindex@at\noexpand\@nnil
          \@glsxtr@autoindex@at\noexpand\@empty\noexpand\@glsxtr@endescspch}%
      \@@glsxtr@checkspch
      \let#1\@gls@checkedmkidx\relax
    \fi
    \ifx\@glsxtr@autoindex@level\@gls@levelchar
    \else
      \def\@gls@checkedmkidx{}%
      \edef\@@glsxtr@checkspch{%
        \noexpand\@glsxtr@autoindex@esclevel\expandonce{#1}%
          \noexpand\@empty\@glsxtr@autoindex@level\noexpand\@nnil
          \@glsxtr@autoindex@level\noexpand\@empty\noexpand\@glsxtr@endescspch}%
      \@@glsxtr@checkspch
      \let#1\@gls@checkedmkidx\relax
    \fi
    \ifx\@glsxtr@autoindex@encap\@gls@encapchar
    \else
      \def\@gls@checkedmkidx{}%
      \edef\@@glsxtr@checkspch{%
        \noexpand\@glsxtr@autoindex@escencap\expandonce{#1}%
          \noexpand\@empty\@glsxtr@autoindex@encap\noexpand\@nnil
          \@glsxtr@autoindex@encap\noexpand\@empty\noexpand\@glsxtr@endescspch}%
      \@@glsxtr@checkspch
      \let#1\@gls@checkedmkidx\relax
    \fi
}
\newcommand*{\@glsxtr@autoindex@at}{}
\newcommand*{\GlsXtrSetActualChar}[1]{%
  \gdef\@glsxtr@autoindex@at{#1}%
  \def\@glsxtr@autoindex@escat##1#1##2#1##3\@glsxtr@endescspch{%
    \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escat}{##1}{##2}{##3}%
  }%
}
\@onlypreamble\GlsXtrSetActualChar
\makeatother
\GlsXtrSetActualChar{@}
\makeatletter
\newcommand*{\@glsxtr@autoindex@encap}{}
\newcommand*{\GlsXtrSetEncapChar}[1]{%
```

```
    \gdef\@glsxtr@autoindex@encap{#1}%
    \def\@glsxtr@autoindex@escencap##1#1##2#1##3\@glsxtr@endescspch{%
      \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escencap}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetEncapChar{|}
\@onlypreamble\GlsXtrSetEncapChar
\newcommand*{\@glsxtr@autoindex@level}{}
\newcommand*{\GlsXtrSetLevelChar}[1]{%
  \gdef\@glsxtr@autoindex@level{#1}%
  \def\@glsxtr@autoindex@esclevel##1#1##2#1##3\@glsxtr@endescspch{%
    \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@esclevel}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetLevelChar{!}
\@onlypreamble\GlsXtrSetLevelChar
\newcommand*{\@glsxtr@autoindex@esc}{"}
\newcommand*{\GlsXtrSetEscChar}[1]{%
  \gdef\@glsxtr@autoindex@esc{#1}%
  \def\@glsxtr@autoindex@escquote##1#1##2#1##3\@glsxtr@endescspch{%
    \@@glsxtr@autoindex@escspch{#1}{\@glsxtr@autoindex@escquote}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetEscChar{"}
\@onlypreamble\GlsXtrSetEscChar
\ifdef\actualchar
 {\expandafter\GlsXtrSetActualChar\expandafter{\actualchar}}
 {}
\ifdef\quotechar
 {\expandafter\GlsXtrSetEscChar\expandafter{\quotechar}}
 {}
\ifdef\levelchar
 {\expandafter\GlsXtrSetLevelChar\expandafter{\levelchar}}
 {}
\ifdef\encapchar
 {\expandafter\GlsXtrSetEncapChar\expandafter{\encapchar}}
 {}
\def\@glsxtr@gobbleto@endescspch#1\@glsxtr@endescspch{}
\newcommand*{\@@glsxtr@autoindex@escspch}[5]{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#3}%
  \ifx\@nnil#3\relax
    \def\@@glsxtr@checkspch{\@glsxtr@gobbleto@endescspch#5\@glsxtr@endescspch}%
  \else
    \ifx\@nnil#4\relax
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
     \def\@@glsxtr@checkspch{\@glsxtr@gobbleto@endescspch
        #4#5\@glsxtr@endescspch}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
```

```
      \@glsxtr@autoindex@esc#1}%
        \def\@@glsxtr@checkspch{#2#5#1\@nnil#1\@glsxtr@endescspch}%
    \fi
  \fi
  \@@glsxtr@checkspch
}
\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glssetabbrvfmt{\glscategory{#1}}%
    \Glsaccessdesc{#1}%
  }%
}
\ifdef\texorpdfstring
{
  \renewcommand*{\glossentrysymbol}[1]{%
    \texorpdfstring{\@glossentrysymbol{#1}}{\glsentrypdfsymbol{#1}}%
  }
}
{
  \renewcommand*{\glossentrysymbol}[1]{\@glossentrysymbol{#1}}
}
\newcommand{\glsentrypdfsymbol}[1]{\glsentrysymbol{#1}}
\newrobustcmd*{\@glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \begingroup
      \glssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glosssymbolfont}%
      {%
        \protected@edef\@glsxtr@attrval{\glsgetattribute{#1}{glosssymbolfont}}%
        \ifcsdef{\@glsxtr@attrval}%
        {%
          \letcs{\@glsxtr@glosssymbolfont}{\@glsxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
          '\@glsxtr@attrval' supplied in glosssymbolfont attribute
          for entry '#1'. Ignoring}%
          \let\@glsxtr@glosssymbolfont\@firstofone
        }%
      }%
      {\let\@glsxtr@glosssymbolfont\@firstofone}%
      \@glsxtr@glosssymbolfont{\glsaccesssymbol{#1}}%
    \endgroup
  }%
}
\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
```

```
      \glssetabbrvfmt{\glscategory{#1}}%
      \Glsaccesssymbol{#1}%
  }%
}
\newcommand*{\GlsXtrEnableInitialTagging}{%
  \@ifstar\s@glsxtr@enabletagging\@glsxtr@enabletagging
}
\@onlypreamble\GlsXtrEnableInitialTagging
\newcommand*{\s@glsxtr@enabletagging}[2]{%
  \undef#2%
  \@glsxtr@enabletagging{#1}{#2}%
}
\newcommand*{\@glsxtr@enabletagging}[2]{%
  \@for\@glsxtr@cat:=#1\do
  {%
    \ifdefempty\@glsxtr@cat
    {}%
    {\glssetcategoryattribute{\@glsxtr@cat}{tagging}{true}}%
  }%
  \newrobustcmd*#2[1]{##1}%
  \def\@glsxtr@taggingcs{#2}%
  \renewcommand*\@glsxtr@activate@initialtagging{%
    \let#2\@glsxtr@tag
  }%
  \ifundef\@gls@preglossaryhook
  {\GlossariesExtraWarning{Initial tagging requires at least
    glossaries.sty v4.19 to work correctly}}%
  {}%
}
\ifundef\mfu@checkword@do
{
  \newcommand*{\mfu@checkword@do}[1]{%
   \ifdefstring{\mfu@checkword@arg}{#1}%
   {%
     \let\@mfu@domakefirstuc\@firstofone
     \listbreak
   }%
   {}%
  }
  \ifundef\mfu@checkword
  {
    \newcommand{\@glsxtr@do@titlecaps@warn}{%
     \GlossariesExtraWarning{mfirstuc.sty too old. Title Caps
       support not available}%
     \let\@glsxtr@do@titlecaps@warn\relax
    }
  }
  {
    \renewcommand*{\mfu@checkword}[1]{%
      \def\mfu@checkword@arg{#1}%
```

```
      \let\@mfu@domakefirstuc\makefirstuc
      \forlistloop\mfu@checkword@do\@mfu@nocaplist
    }
  }
}
{}% no patch required
\newcommand*{\@glsxtr@do@titlecaps@warn}{}
\newcommand*\@glsxtr@activate@initialtagging{}
\newrobustcmd*{\@glsxtr@tag}[1]{%
  \glsifattribute{\glscurrententrylabel}{tagging}{true}%
  {\glsxtrtagfont{#1}}{#1}%
}
\newcommand*{\glsxtrtagfont}[1]{\underline{#1}}
\ifdef\@gls@preglossaryhook
{
  \renewcommand*{\@gls@preglossaryhook}{%
    \@glsxtr@activate@initialtagging
    \ifundef\@glsxtr@org@postdescription
    {%
      \let\@glsxtr@org@postdescription\glspostdescription
      \renewcommand*{\glspostdescription}{%
        \ifglsentryexists{\glscurrententrylabel}%
        {%
          \glsxtrpostdescription
          \@glsxtr@org@postdescription
        }%
        {}%
      }%
    }%
    {}%
    \glossxtrsetpopts
  }%
}
{}
\newcommand*{\glsxtrpostdescription}{%
  \csuse{glsxtrpostdesc\glscategory{\glscurrententrylabel}}%
}
\newcommand*{\glsxtrpostdescgeneral}{}
\newcommand*{\glsxtrpostdescterm}{}
\newcommand*{\glsxtrpostdescacronym}{}
\newcommand*{\glsxtrpostdescabbreviation}{}
\newcommand*{\glsdefpostdesc}[2]{%
  \csdef{glsxtrpostdesc#1}{#2}%
}
\renewcommand*{\glspostlinkhook}{%
 \ifglsentryexists{\glslabel}{\glsxtrpostlinkhook}{}%
}
\newcommand*{\glsxtrpostlinkhook}{%
 \glsxtrdiscardperiod{\glslabel}%
 {\glsxtrpostlinkendsentence}%
```

```
  {\glsxtrifcustomdiscardperiod
   {\glsxtrifperiod{\glsxtrpostlinkendsentence}{\glsxtrpostlink}}%
   {\glsxtrpostlink}%
 }%
}
\newcommand*{\glsxtrifcustomdiscardperiod}[2]{#2}
\newcommand*{\glsxtrpostlink}{%
 \csuse{glsxtrpostlink\glscategory{\glslabel}}%
}
\newcommand*{\glsdefpostlink}[2]{%
  \ifthenelse{\equal{#1}{}}%
  {\PackageError{glossaries-extra}
    {Invalid empty category label in \string\glsdefpostlink}{}}%
  {\csdef{glsxtrpostlink#1}{#2}}%
}
\newcommand*{\glsxtrpostlinkendsentence}{%
 \ifcsdef{glsxtrpostlink\glscategory{\glslabel}}
 {%
    \csuse{glsxtrpostlink\glscategory{\glslabel}}%
    .\spacefactor\sfcode`\. \relax
 }%
 {%
    \spacefactor\sfcode`\. \relax
 }%
}
\newcommand*{\glsxtrpostlinkAddDescOnFirstUse}{%
  \glsxtrifwasfirstuse{\space\glsxtrparen{\glsaccessdesc{\glslabel}}}{}%
}
\newcommand*{\glsxtrpostlinkAddSymbolOnFirstUse}{%
  \glsxtrifwasfirstuse
  {%
    \ifglshassymbol{\glslabel}%
    {\space\glsxtrparen{\glsaccesssymbol{\glslabel}}}%
    {}%
  }%
  {}%
}
\newcommand*{\glsxtrpostlinkAddSymbolDescOnFirstUse}{%
  \glsxtrifwasfirstuse
  {%
    \space\glsxtrparen
    {%
      \ifglshassymbol{\glslabel}%
      {\glsaccesssymbol{\glslabel}, }%
      {}%
      \glsaccessdesc{\glslabel}%
    }%
  }%
  {}%
}
```

```
\newcommand*{\glsxtrdiscardperiod}[3]{%
 \glsxtrifwasfirstuse
 {%
   \glsifattribute{#1}{retainfirstuseperiod}{true}%
   {#3}%
   {%
     \glsifattribute{#1}{discardperiod}{true}%
     {%
       \glsifplural
       {%
         \glsifattribute{#1}{pluraldiscardperiod}{true}%
         {\glsxtrifperiod{#2}{#3}}%
         {#3}%
       }%
       {%
         \glsxtrifperiod{#2}{#3}%
       }%
     }%
     {#3}%
   }%
 }%
 {%
   \glsifattribute{#1}{discardperiod}{true}%
   {%
     \glsifplural
     {%
       \glsifattribute{#1}{pluraldiscardperiod}{true}%
       {\glsxtrifperiod{#2}{#3}}%
       {#3}%
     }%
     {%
       \glsxtrifperiod{#2}{#3}%
     }%
   }%
   {#3}%
 }%
}
\newcommand*{\glsxtrifperiod}[1]{\new@ifnextchar.{\@firstoftwo{#1}}}
\newcommand*{\glsxtr@punclist}{.,:;?!}
\newcommand*{\glsxtraddpunctuationmark}[1]{\appto\glsxtr@punclist{#1}}
\newcommand*{\glsxtrsetpunctuationmarks}[1]{\def\glsxtr@punclist{#1}}
\newcommand*{\glsxtrifnextpunc}[2]{%
  \def\reserved@a{#1}%
  \def\reserved@b{#2}%
  \futurelet\@glspunc@token\glsxtr@ifnextpunc
}
\newcommand*{\glsxtr@ifnextpunc}{%
 \glsxtr@ifpunctoken{\@glspunc@token}{\let\reserved@b\reserved@a}{}%
 \reserved@b
}
```

```
\newcommand*{\glsxtr@ifpunctoken}[1]{%
  \expandafter\@glsxtr@ifpunctoken\expandafter#1\glsxtr@punclist\@nnil
}
\def\@glsxtr@ifpunctoken#1#2{%
  \let\reserved@d=#2%
  \ifx\reserved@d\@nnil
    \let\glsxtr@next\@glsxtr@notfoundinlist
  \else
    \ifx#1\reserved@d
     \let\glsxtr@next\@glsxtr@foundinlist
    \else
     \let\glsxtr@next\@glsxtr@ifpunctoken
    \fi
  \fi
  \glsxtr@next#1%
}
\def\@glsxtr@foundinlist#1\@nnil{\@firstoftwo}
\def\@glsxtr@notfoundinlist#1{\@secondoftwo}
\newcommand{\glsxtrdopostpunc}[1]{%
  \glsxtrifnextpunc{\@glsxtr@swaptwo{#1}}{#1}%
}
\newcommand{\@glsxtr@swaptwo}[2]{#2#1}
\define@key{glsxtrabbrv}{category}{%
 \protected@edef\glscategorylabel{#1}%
}
\define@key{glsxtrabbrv}{shortplural}{%
  \def\@gls@shortpl{#1}%
}
\define@key{glsxtrabbrv}{longplural}{%
  \def\@gls@longpl{#1}%
}
\newtoks\glsshortpltok
\newtoks\glslongpltok
\newcommand*{\@glsxtr@insertdots}[2]{%
  \def#1{}%
  \@glsxtr@insert@dots#1#2\@nnil
}
\newcommand*{\@glsxtr@insert@dots}[2]{%
  \ifx\@nnil#2\relax
   \let\@glsxtr@insert@dots@next\@gobble
  \else
   \ifx\relax#2\relax
   \else
     \appto#1{#2.}%
   \fi
   \let\@glsxtr@insert@dots@next\@glsxtr@insert@dots
  \fi
  \@glsxtr@insert@dots@next#1%
}
\newcommand*{\glsxtrwordsep}{\space}
```

```
\newcommand*{\glsxtrword}[1]{#1}
\newcommand*{\@glsxtr@markwordseps}[2]{%
  \def#1{}%
  \@glsxtr@mark@wordseps#1#2 \@nnil
}
\def\@glsxtr@mark@wordseps#1#2 #3{%
  \ifdefempty{#1}%
  {\def#1{\protect\glsxtrword{#2}}}%
  {\appto#1{\protect\glsxtrwordsep\protect\glsxtrword{#2}}}%
  \ifx\@nnil#3\relax
   \let\@glsxtr@mark@wordseps@next\relax
  \else
   \def\@glsxtr@mark@wordseps@next{%
     \@glsxtr@mark@wordseps#1#3}%
  \fi
  \@glsxtr@mark@wordseps@next
}
\newcommand*{\newabbreviation}[4][]{%
  \glsxtr@newabbreviation{#1}{#2}{#3}{#4}%
}
\newcommand*{\glsxtr@newabbreviation}[4]{%
  \glskeylisttok{#1}%
  \glslabeltok{#2}%
  \glsshorttok{#3}%
  \glslongtok{#4}%
  \def\glsxtrorgshort{#3}%
  \def\glsxtrorglong{#4}%
  \def\ExtraCustomAbbreviationFields{}%
  \@gls@initaccesskeys
  \def\glscategorylabel{abbreviation}%
  \setkeys*{glsxtrabbrv}[shortplural,longplural]{#1}%
  \ifcsdef{@glsabbrv@current@\glscategorylabel}%
  {%
    \let\@glsxtr@orgwarndep\GlsXtrWarnDeprecatedAbbrStyle
    \let\GlsXtrWarnDeprecatedAbbrStyle\@gobbletwo
    \glsxtr@applyabbrvstyle{\csname @glsabbrv@current@\glscategorylabel\endcsname}%
    \let\GlsXtrWarnDeprecatedAbbrStyle\@glsxtr@orgwarndep
  }%
  {%
    \glsxtr@applyabbrvstyle{\@glsabbrv@current@abbreviation}%
  }%
  \def\@gls@longpl{#4\glspluralsuffix}%
  \let\@gls@default@longpl\@gls@longpl
  \glsifcategoryattribute{\glscategorylabel}{markwords}{true}%
  {%
    \@glsxtr@markwordseps\@gls@long{#4}%
    \expandafter\def\expandafter\@gls@longpl\expandafter
     {\@gls@long\glspluralsuffix}%
    \let\@gls@default@longpl\@gls@longpl
    \expandafter\glslongtok\expandafter{\@gls@long}%
```

```
}%
{}%
\glsifcategoryattribute{\glscategorylabel}{markshortwords}{true}%
{%
  \@glsxtr@markwordseps\@gls@short{#3}%
}%
{%
  \glsifcategoryattribute{\glscategorylabel}{insertdots}{true}%
  {%
    \@glsxtr@insertdots\@gls@short{#3}%
    \appto\@gls@short{\@}%
  }%
  {\def\@gls@short{#3}}%
}%
\glsifcategoryattribute{\glscategorylabel}{aposplural}{true}%
{%
  \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
    '\abbrvpluralsuffix}%
}%
{%
  \glsifcategoryattribute{\glscategorylabel}{noshortplural}{true}%
  {%
    \let\@gls@shortpl\@gls@short
  }%
    {%
    \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
      \abbrvpluralsuffix}%
  }%
}%
\expandafter\glsshorttok\expandafter{\@gls@short}%
\glsxtrnewabbrevpresetkeyhook{#1}{#2}{#3}%
\setkeys*{glsxtrabbrv}[category]{#1}%
  \let\@gls@org@longpl\@gls@longpl
  \let\@gls@org@shortpl\@gls@shortpl
\ifx\@gls@default@longpl\@gls@longpl
\else
  \glsifcategoryattribute{\glscategorylabel}{markwords}{true}%
  {%
    \expandafter\@glsxtr@markwordseps\expandafter\@gls@longpl\expandafter
      {\@gls@longpl}%
  }%
  {}%
\fi
\expandafter\glsshortpltok\expandafter{\@gls@shortpl}%
\expandafter\glslongpltok\expandafter{\@gls@longpl}%
\@gls@setup@default@access
\newabbreviationhook
\protected@edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
```

```
      type=\glsxtrabbrvtype,%
      category=abbreviation,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshortpltok},%
      long={\the\glslongtok},%
      longplural={\the\glslongpltok},%
      name={\the\glsshorttok},%
      \CustomAbbreviationFields,%
      \ExtraCustomAbbreviationFields
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
  \@glsxtr@addabbreviationlist{\glsentrytype{\the\glslabeltok}}%
  \GlsXtrPostNewAbbreviation
}
\newcommand*{\glsxtrnewabbrevpresetkeyhook}[3]{}
\newcommand*{\GlsXtrPostNewAbbreviation}{}
\newcommand*{\newabbreviationhook}{}
\newcommand*{\CustomAbbreviationFields}{}
\newcommand*{\glsxtrparen}[1]{(#1)}
\newcommand*{\glsxtrfullformat}[2]{%
  \glsfirstlongfont{\glsaccesslong{#1}}#2\glsxtrfullsep{#1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{#1}}}%
}
\newcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongfont{\Glsaccesslong{#1}}#2\glsxtrfullsep{#1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{#1}}}%
}
\newcommand*{\glsxtrfullplformat}[2]{%
  \glsfirstlongfont{\glsaccesslongpl{#1}}#2\glsxtrfullsep{#1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{#1}}}%
}
\newcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstlongfont{\Glsaccesslongpl{#1}}#2\glsxtrfullsep{#1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{#1}}}%
}
\newcommand*{\glsxtrfullsep}[1]{\space}
\newcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}
\newcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}
\newcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}
\newcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}
\renewcommand*{\glsentryfull}[1]{\glsxtrinlinefullformat{#1}{}}
\renewcommand*{\Glsentryfull}[1]{\Glsxtrinlinefullformat{#1}{}}
\renewcommand*{\glsentryfullpl}[1]{\glsxtrinlinefullplformat{#1}{}}
\renewcommand*{\Glsentryfullpl}[1]{\Glsxtrinlinefullplformat{#1}{}}
\newcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{#1}}
\newcommand*{\glsfirstabbrvdefaultfont}[1]{\glsabbrvdefaultfont{#1}}
\newcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{#1}}
\newcommand*{\glsabbrvdefaultfont}[1]{#1}
```

```
\newcommand*{\glslongfont}[1]{\glslongdefaultfont{#1}}
\newcommand*{\glslongdefaultfont}[1]{#1}
\newcommand*{\glsfirstlongfont}[1]{\glslongfont{#1}}
\newcommand*{\glsfirstlongdefaultfont}[1]{\glslongdefaultfont{#1}}
\newcommand*{\glsxtrabbrvpluralsuffix}{\glspluralsuffix}
\newcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}
\newrobustcmd*{\glsxtrfull}{\@gls@hyp@opt\ns@glsxtrfull}
\newcommand*\ns@glsxtrfull[2][]{%
  \new@ifnextchar[{\@glsxtr@full{#1}{#2}}%
                 {\@glsxtr@full{#1}{#2}[]}%
}
\def\@glsxtr@full#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{\glsxtrinlinefullformat{#2}{#3}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\glsxtrsetupfulldefs}{%
  \let\glsxtrifwasfirstuse\@firstoftwo
}
\newrobustcmd*{\Glsxtrfull}{\@gls@hyp@opt\ns@Glsxtrfull}
\newcommand*\ns@Glsxtrfull[2][]{%
  \new@ifnextchar[{\@Glsxtr@full{#1}{#2}}%
                 {\@Glsxtr@full{#1}{#2}[]}%
}
\def\@Glsxtr@full#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{\Glsxtrinlinefullformat{#2}{#3}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSxtrfull}{\@gls@hyp@opt\ns@GLSxtrfull}
\newcommand*\ns@GLSxtrfull[2][]{%
```

```
        \new@ifnextchar[{\@GLSxtr@full{#1}{#2}}%
                       {\@GLSxtr@full{#1}{#2}[]}%
}
\def\@GLSxtr@full#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{\mfirstucMakeUppercase{\glsxtrinlinefullformat{#2}{#3}}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsxtrfullpl}{\@gls@hyp@opt\ns@glsxtrfullpl}
\newcommand*\ns@glsxtrfullpl[2][]{%
  \new@ifnextchar[{\@glsxtr@fullpl{#1}{#2}}%
                 {\@glsxtr@fullpl{#1}{#2}[]}%
}
\def\@glsxtr@fullpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{\glsxtrinlinefullplformat{#2}{#3}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glsxtrfullpl}{\@gls@hyp@opt\ns@Glsxtrfullpl}
\newcommand*\ns@Glsxtrfullpl[2][]{%
  \new@ifnextchar[{\@Glsxtr@fullpl{#1}{#2}}%
                 {\@Glsxtr@fullpl{#1}{#2}[]}%
}
\def\@Glsxtr@fullpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
```

```
    \let\glsinsert\@empty
    \def\glscustomtext{\Glsxtrinlinefullplformat{#2}{#3}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSxtrfullpl}{\@gls@hyp@opt\ns@GLSxtrfullpl}
\newcommand*\ns@GLSxtrfullpl[2][]{%
  \new@ifnextchar[{\@GLSxtr@fullpl{#1}{#2}}%
                 {\@GLSxtr@fullpl{#1}{#2}[]}%
}
\def\@GLSxtr@fullpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsxtrinlinefullplformat{#2}{#3}}}%
    \glsxtrsetupfulldefs
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsxtrshort}{\@gls@hyp@opt\ns@glsxtrshort}
\newcommand*{\ns@glsxtrshort}[2][]{%
  \new@ifnextchar[{\@glsxtrshort{#1}{#2}}{\@glsxtrshort{#1}{#2}[]}%
}
\def\@glsxtrshort#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsabbrvfont{\glsaccessshort{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glsxtrshort}{\@gls@hyp@opt\ns@Glsxtrshort}
```

```
\newcommand*{\ns@Glsxtrshort}[2][]{%
  \new@ifnextchar[{\@Glsxtrshort{#1}{#2}}{\@Glsxtrshort{#1}{#2}[]}%
}
\def\@Glsxtrshort#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsabbrvfont{\Glsaccessshort{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSxtrshort}{\@gls@hyp@opt\ns@GLSxtrshort}
\newcommand*{\ns@GLSxtrshort}[2][]{%
  \new@ifnextchar[{\@GLSxtrshort{#1}{#2}}{\@GLSxtrshort{#1}{#2}[]}%
}
\def\@GLSxtrshort#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase
      {\glsabbrvfont{\glsaccessshort{#2}\ifglsxtrinsertinside#3\fi}%
        \ifglsxtrinsertinside\else#3\fi
      }%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsxtrlong}{\@gls@hyp@opt\ns@glsxtrlong}
\newcommand*{\ns@glsxtrlong}[2][]{%
  \new@ifnextchar[{\@glsxtrlong{#1}{#2}}{\@glsxtrlong{#1}{#2}[]}%
}
\def\@glsxtrlong#1#2[#3]{%
```

```
    \@glsxtr@record{#1}{#2}{glslink}%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glsxtrifwasfirstuse\@secondoftwo
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@firstofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \glslongfont{\glsaccesslong{#2}\ifglsxtrinsertinside#3\fi}%
        \ifglsxtrinsertinside\else#3\fi
      }%
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\Glsxtrlong}{\@gls@hyp@opt\ns@Glsxtrlong}
\newcommand*{\ns@Glsxtrlong}[2][]{%
  \new@ifnextchar[{\@Glsxtrlong{#1}{#2}}{\@Glsxtrlong{#1}{#2}[]}%
}
\def\@Glsxtrlong#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glslongfont{\Glsaccesslong{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSxtrlong}{\@gls@hyp@opt\ns@GLSxtrlong}
\newcommand*{\ns@GLSxtrlong}[2][]{%
  \new@ifnextchar[{\@GLSxtrlong{#1}{#2}}{\@GLSxtrlong{#1}{#2}[]}%
}
\def\@GLSxtrlong#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
```

```
    \def\glscustomtext{%
     \mfirstucMakeUppercase
     {\glslongfont{\glsaccesslong{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
     }%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsxtrshortpl}{\@gls@hyp@opt\ns@glsxtrshortpl}
\newcommand*{\ns@glsxtrshortpl}[2][]{%
  \new@ifnextchar[{\@glsxtrshortpl{#1}{#2}}{\@glsxtrshortpl{#1}{#2}[]}%
}
\def\@glsxtrshortpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsabbrvfont{\glsaccessshortpl{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glsxtrshortpl}{\@gls@hyp@opt\ns@Glsxtrshortpl}
\newcommand*{\ns@Glsxtrshortpl}[2][]{%
  \new@ifnextchar[{\@Glsxtrshortpl{#1}{#2}}{\@Glsxtrshortpl{#1}{#2}[]}%
}
\def\@Glsxtrshortpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsabbrvfont{\Glsaccessshortpl{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
    }%
```

```
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSxtrshortpl}{\@gls@hyp@opt\ns@GLSxtrshortpl}
\newcommand*{\ns@GLSxtrshortpl}[2][]{%
  \new@ifnextchar[{\@GLSxtrshortpl{#1}{#2}}{\@GLSxtrshortpl{#1}{#2}[]}%
}
\def\@GLSxtrshortpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase
      {\glsabbrvfont{\glsaccessshortpl{#2}\ifglsxtrinsertinside#3\fi}%
       \ifglsxtrinsertinside\else#3\fi
      }%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsxtrlongpl}{\@gls@hyp@opt\ns@glsxtrlongpl}
\newcommand*{\ns@glsxtrlongpl}[2][]{%
  \new@ifnextchar[{\@glsxtrlongpl{#1}{#2}}{\@glsxtrlongpl{#1}{#2}[]}%
}
\def\@glsxtrlongpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glslongfont{\glsaccesslongpl{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glsxtrlongpl}{\@gls@hyp@opt\ns@Glsxtrlongpl}
```

```
\newcommand*{\ns@Glsxtrlongpl}[2][]{%
  \new@ifnextchar[{\@Glsxtrlongpl{#1}{#2}}{\@Glsxtrlongpl{#1}{#2}[]}%
}
\def\@Glsxtrlongpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glslongfont{\Glsaccesslongpl{#2}\ifglsxtrinsertinside#3\fi}%
      \ifglsxtrinsertinside\else#3\fi
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSxtrlongpl}{\@gls@hyp@opt\ns@GLSxtrlongpl}
\newcommand*{\ns@GLSxtrlongpl}[2][]{%
  \new@ifnextchar[{\@GLSxtrlongpl{#1}{#2}}{\@GLSxtrlongpl{#1}{#2}[]}%
}
\def\@GLSxtrlongpl#1#2[#3]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase
      {\glslongfont{\glsaccesslongpl{#2}\ifglsxtrinsertinside#3\fi}%
       \ifglsxtrinsertinside\else#3\fi
      }%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\glssetabbrvfmt}[1]{%
  \ifcsdef{@glsabbrv@current@#1}%
  {\glsxtr@applyabbrvfmt{\csname @glsabbrv@current@#1\endcsname}}%
  {\glsxtr@applyabbrvfmt{\@glsabbrv@current@abbreviation}}%
}
\newrobustcmd*{\glsuseabbrvfont}[2]{{\glssetabbrvfmt{#2}\glsabbrvfont{#1}}}
\newrobustcmd*{\glsuselongfont}[2]{{\glssetabbrvfmt{#2}\glslongfont{#1}}}
```

```
\newcommand*{\glsxtrgenabbrvfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsxtrsubsequentplfmt{\glslabel}{\glsinsert}%
        }%
        {%
          \Glsxtrsubsequentplfmt{\glslabel}{\glsinsert}%
        }%
        {%
          \mfirstucMakeUppercase
            {\glsxtrsubsequentplfmt{\glslabel}{\glsinsert}}%
        }%
      }%
      {%
        \glscapscase
        {%
          \glsxtrsubsequentfmt{\glslabel}{\glsinsert}%
        }%
        {%
          \Glsxtrsubsequentfmt{\glslabel}{\glsinsert}%
        }%
        {%
          \mfirstucMakeUppercase
            {\glsxtrsubsequentfmt{\glslabel}{\glsinsert}}%
        }%
      }%
    }%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsxtrfullplformat{\glslabel}{\glsinsert}%
        }%
        {%
          \Glsxtrfullplformat{\glslabel}{\glsinsert}%
        }%
        {%
          \mfirstucMakeUppercase
            {\glsxtrfullplformat{\glslabel}{\glsinsert}}%
        }%
      }%
      {%
        \glscapscase
```

983

```
        {%
          \glsxtrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
          \Glsxtrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
          \mfirstucMakeUppercase
           {\glsxtrfullformat{\glslabel}{\glsinsert}}%
        }%
      }%
    }%
  }%
  {%
    \glscustomtext
  }%
}
\newcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsabbrvfont{\glsaccessshort{#1}\ifglsxtrinsertinside #2\fi}%
  \ifglsxtrinsertinside \else#2\fi
}
\let\glsxtrdefaultsubsequentfmt\glsxtrsubsequentfmt
\newcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsabbrvfont{\glsaccessshortpl{#1}\ifglsxtrinsertinside #2\fi}%
  \ifglsxtrinsertinside \else#2\fi
}
\let\glsxtrdefaultsubsequentplfmt\glsxtrsubsequentplfmt
\newcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glsabbrvfont{\Glsaccessshort{#1}\ifglsxtrinsertinside #2\fi}%
  \ifglsxtrinsertinside \else#2\fi
}
\let\Glsxtrdefaultsubsequentfmt\Glsxtrsubsequentfmt
\newcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glsabbrvfont{\Glsaccessshortpl{#1}\ifglsxtrinsertinside #2\fi}%
  \ifglsxtrinsertinside \else#2\fi
}
\let\Glsxtrdefaultsubsequentplfmt\Glsxtrsubsequentplfmt
\newcommand*{\setabbreviationstyle}[2][abbreviation]{%
  \ifcsundef{@glsabbrv@dispstyle@setup@#2}
  {%
    \PackageError{glossaries-extra}{Undefined abbreviation style '#2'}{}%
  }%
  {%
    \ifcsstring{@glsabbrv@current@#1}{#2}%
    {%
    }%
    {%
      \def\@glsxtr@dostylewarn{}%
      \glsforeachincategory{#1}{\@gls@type}{\@gls@label}%
      {%
```

```
      \def\@glsxtr@dostylewarn{\GlossariesWarning{Abbreviation
        style has been switched \MessageBreak
        for category '#1', \MessageBreak
        but there have already been entries \MessageBreak
        defined for this category. Unwanted \MessageBreak
        side-effects may result}}%
      \@endfortrue
    }%
    \@glsxtr@dostylewarn
    \csdef{@glsabbrv@current@#1}{#2}%
    \protected@edef\glscategorylabel{#1}%
    \glsxtr@applyabbrvstyle{#2}%
  }%
  }%
}
\newcommand*{\glsxtr@applyabbrvstyle}[1]{%
  \csuse{@glsabbrv@dispstyle@setup@#1}%
  \csuse{@glsabbrv@dispstyle@fmts@#1}%
}
\newcommand*{\glsxtr@applyabbrvfmt}[1]{%
  \csuse{@glsabbrv@dispstyle@fmts@#1}%
}
\newcommand*{\newabbreviationstyle}[3]{%
  \ifcsdef{@glsabbrv@dispstyle@setup@#1}
  {%
    \PackageError{glossaries-extra}{Abbreviation style '#1' already
     defined}{}%
  }%
  {%
    \csdef{@glsabbrv@dispstyle@setup@#1}{%
     \renewcommand*{\GlsXtrPostNewAbbreviation}{}%
     #2}%
    \csdef{@glsabbrv@dispstyle@fmts@#1}{%
    \renewcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}%
    \renewcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}%
    \renewcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}%
    \renewcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}%
    \let\glsxtrsubsequentfmt\glsxtrdefaultsubsequentfmt
    \let\glsxtrsubsequentplfmt\glsxtrdefaultsubsequentplfmt
    \let\Glsxtrsubsequentfmt\Glsxtrdefaultsubsequentfmt
    \let\Glsxtrsubsequentplfmt\Glsxtrdefaultsubsequentplfmt
    #3}%
  }%
}
\newcommand*{\renewabbreviationstyle}[3]{%
  \ifcsundef{@glsabbrv@dispstyle@setup@#1}
  {%
    \PackageError{glossaries-extra}{Abbreviation style '#1' not defined}{}%
  }%
  {%
```

```
      \csdef{@glsabbrv@dispstyle@setup@#1}{%
       \renewcommand*{\GlsXtrPostNewAbbreviation}{}%
       #2}%
      \csdef{@glsabbrv@dispstyle@fmts@#1}{%
      \renewcommand*{\glsxtrinlinefullformat}{\glsxtrfullformat}%
      \renewcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}%
      \renewcommand*{\glsxtrinlinefullplformat}{\glsxtrfullplformat}%
      \renewcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}%
      #3}%
   }%
}
\newcommand*{\letabbreviationstyle}[2]{%
  \csletcs{@glsabbrv@dispstyle@setup@#1}{@glsabbrv@dispstyle@setup@#2}%
  \csletcs{@glsabbrv@dispstyle@fmts@#1}{@glsabbrv@dispstyle@fmts@#2}%
}
\newcommand*{\@glsxtr@deprecated@abbrstyle}[2]{%
  \csdef{@glsabbrv@dispstyle@setup@#1}{%
    \GlsXtrWarnDeprecatedAbbrStyle{#1}{#2}%
    \csuse{@glsabbrv@dispstyle@setup@#2}%
  }%
  \csletcs{@glsabbrv@dispstyle@fmts@#1}{@glsabbrv@dispstyle@fmts@#2}%
}
\newcommand*{\GlsXtrWarnDeprecatedAbbrStyle}[2]{%
  \GlossariesExtraWarning{Deprecated abbreviation style name '#1',
  use '#2' instead}%
}
\newcommand*{\GlsXtrUseAbbrStyleSetup}[1]{%
  \ifcsundef{@glsabbrv@dispstyle@setup@#1}%
  {%
     \PackageError{glossaries-extra}%
     {Unknown abbreviation style definitions '#1'}{}%
  }%
  {%
     \csname @glsabbrv@dispstyle@setup@#1\endcsname
  }%
}
\newcommand*{\GlsXtrUseAbbrStyleFmts}[1]{%
  \ifcsundef{@glsabbrv@dispstyle@fmts@#1}%
  {%
     \PackageError{glossaries-extra}%
     {Unknown abbreviation style formats '#1'}{}%
  }%
  {%
     \csname @glsabbrv@dispstyle@fmts@#1\endcsname
  }%
}
\newif\ifglsxtrinsertinside
\glsxtrinsertinsidefalse
\newcommand*{\glsxtrlongshortname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
```

```
}
\newabbreviationstyle{long-short}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvfont{\the\glsshortpltok}}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
```

```
    }%
}
\setabbreviationstyle{long-short}
\newcommand*{\glsxtrlongshortdescsort}{%
 \expandonce\glsxtrorglong\space (\expandonce\glsxtrorgshort)%
}
\newcommand*{\glsxtrlongshortdescname}{%
  \protect\glslongfont{\the\glslongtok}
  \glsxtrparen{\protect\glsabbrvfont{\the\glsshorttok}}%
}
\newabbreviationstyle{long-short-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortdescname},
    sort={\glsxtrlongshortdescsort},%
    first={\protect\glsfirstlongfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-short}%
}
\newcommand*{\glsxtrshortlongname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
}
\newabbreviationstyle{short-long}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
```

```
      \protect\glsxtrfullsep{\the\glslabeltok}%
      \glsxtrparen{\protect\glsfirstlongfont{\the\glslongpltok}}},%
     text={\protect\glsabbrvfont{\the\glsshorttok}},%
     plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
     \glshasattribute{\the\glslabeltok}{regular}%
     {%
       \glssetattribute{\the\glslabeltok}{regular}{false}%
     }%
     {}%
   }%
}%
{%
   \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
   \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
   \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
   \renewcommand*{\glsxtrfullformat}[2]{%
     \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
     \glsxtrfullsep{##1}%
     \glsxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
   }%
   \renewcommand*{\glsxtrfullplformat}[2]{%
     \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
     \glsxtrfullsep{##1}%
     \glsxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
   }%
   \renewcommand*{\Glsxtrfullformat}[2]{%
     \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
     \glsxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
   }%
   \renewcommand*{\Glsxtrfullplformat}[2]{%
     \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
     \glsxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
   }%
}
\newcommand*{\glsxtrshortlongdescsort}{\the\glsshorttok}
\newcommand*{\glsxtrshortlongdescname}{%
  \protect\glsabbrvfont{\the\glsshorttok}
  \glsxtrparen{\protect\glslongfont{\the\glslongtok}}%
}
\newabbreviationstyle{short-long-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
```

```
      name={\glsxtrshortlongdescname},
      sort={\glsxtrshortlongdescsort},
      first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \glsxtrparen{\protect\glsfirstlongfont{\the\glslongtok}}},%
      firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \glsxtrparen{\protect\glsfirstlongfont{\the\glslongpltok}}},%
      text={\protect\glsabbrvfont{\the\glsshorttok}},%
      plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
    }%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-long}%
}
\newcommand*{\glsfirstlongfootnotefont}[1]{\glslongfootnotefont{#1}}%
\newcommand*{\glslongfootnotefont}[1]{\glslongdefaultfont{#1}}%
\newcommand*{\glsxtrabbrvfootnote}[2]{\footnote{#2}}
\newcommand*{\glsxtrfootnotename}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
}
\newabbreviationstyle{footnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
```

```
    }%
  }%
  {%
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
    \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
```

```
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
}
\letabbreviationstyle{short-footnote}{footnote}
\newcommand*{\glsxtrfootnotedescname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\protect\glslongfont{\the\glslongtok}}%
}
\newcommand*{\glsxtrfootnotedescsort}{\the\glsshorttok}
\newabbreviationstyle{short-footnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
    sort={\glsxtrfootnotedescsort},
    first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
       {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
       {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{footnote}%
}
\letabbreviationstyle{footnote-desc}{short-footnote-desc}
\newabbreviationstyle{postfootnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
```

```
      \glsxtrifwasfirstuse
      {%
        \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
        {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
  \renewcommand*{\glsxtrsetupfulldefs}{%
    \let\glsxtrifwasfirstuse\@secondoftwo
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
```

```latex
      \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
   }%
   \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
   }%
}
\letabbreviationstyle{short-postfootnote}{postfootnote}
\newabbreviationstyle{short-postfootnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
    sort={\glsxtrfootnotedescsort},
    first={\protect\glsfirstabbrvfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
        {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
  \renewcommand*{\glsxtrsetupfulldefs}{%
    \let\glsxtrifwasfirstuse\@secondoftwo
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{postfootnote}%
}
\letabbreviationstyle{postfootnote-desc}{short-postfootnote-desc}
\newcommand*{\glsxtrshortnolongname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
}
\newabbreviationstyle{short}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrshortnolongname},
  sort={\the\glsshorttok},
  first={\protect\glsfirstabbrvfont{\the\glsshorttok}},
  firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}},
  text={\protect\glsabbrvfont{\the\glsshorttok}},
  plural={\protect\glsabbrvfont{\the\glsshortpltok}},
  description={\the\glslongtok}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{regular}{true}}}%
}%
{%
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \protect\glsfirstabbrvfont{\glsaccessshort{##1}%
    \ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \protect\glsfirstabbrvfont{\glsaccessshortpl{##1}%
   \ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \protect\glsfirstabbrvfont{\glsaccessshort{##1}%
    \ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfont{\Glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \protect\glsfirstabbrvfont{\glsaccessshortpl{##1}%
    \ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfont{\Glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
```

```
      \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
}
\setabbreviationstyle[acronym]{short}
\letabbreviationstyle{short-nolong}{short}
\newabbreviationstyle{short-nolong-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{short-nolong}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong}%
}
\newcommand*{\glsxtrshortdescname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\protect\glslongfont{\the\glslongtok}}%
}
\newabbreviationstyle{short-desc}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortdescname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvfont{\the\glsshorttok}},
    firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}},
    text={\protect\glsabbrvfont{\the\glsshorttok}},
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
```

```
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi
    }%
}
\letabbreviationstyle{short-nolong-desc}{short-desc}
\newabbreviationstyle{short-nolong-desc-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{short-nolong-desc}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong-desc}%
}
```

```
\newabbreviationstyle{nolong-short}%
{%
  \GlsXtrUseAbbrStyleSetup{short-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongfont{\glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongfont{\glsaccesslongpl{##1}%
     \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongfont{\glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\Glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongfont{\glsaccesslongpl{##1}%
       \ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\Glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{nolong-short-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{nolong-short}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{nolong-short}%
}
\newcommand*{\glsxtrlongnoshortdescname}{%
  \protect\glslongfont{\the\glslongtok}%
}
\newabbreviationstyle{long-desc}%
{%
```

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glsxtrlongnoshortdescname},
  sort={\the\glslongtok},
  first={\protect\glsfirstlongfont{\the\glslongtok}},
  firstplural={\protect\glsfirstlongfont{\the\glslongpltok}},
  text={\glslongfont{\the\glslongtok}},
  plural={\glslongfont{\the\glslongpltok}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glslongfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glslongfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glslongfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glslongfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsfirstlongfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsfirstlongfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstlongfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstlongfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
```

```
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\letabbreviationstyle{long-noshort-desc}{long-desc}
\newabbreviationstyle{long-noshort-desc-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-noshort-desc}%
}
\newcommand*{\glsxtrlongnoshortname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
}
\newabbreviationstyle{long}%
{%
  \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongfont{\the\glslongtok}},
    firstplural={\protect\glsfirstlongfont{\the\glslongpltok}},
    text={\glslongfont{\the\glslongtok}},
    plural={\glslongfont{\the\glslongpltok}},%
    description={\the\glslongtok}%
  }%
```

```
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-desc}%
}
\letabbreviationstyle{long-noshort}{long}
\newabbreviationstyle{long-noshort-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-noshort}%
}
\newcommand*{\glsxtrscfont}[1]{\textsc{#1}}
\newcommand*{\glsabbrvscfont}{\glsxtrscfont}
\newcommand*{\glsxtrfirstscfont}[1]{\glsabbrvscfont{#1}}
\newcommand*{\glsfirstabbrvscfont}{\glsxtrfirstscfont}
\newcommand*{\glsxtrscsuffix}{\protect\glstextup{\glsxtrabbrvpluralsuffix}}
\newabbreviationstyle{long-short-sc}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvscfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvscfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvscfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}},%
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
```

```
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
    \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
    }%
}
\newabbreviationstyle{long-short-sc-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortdescname},
    sort={\glsxtrlongshortdescsort},%
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvscfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvscfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvscfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
```

```
{%
  \GlsXtrUseAbbrStyleFmts{long-short-sc}%
}
\newabbreviationstyle{short-sc-long}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvscfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
```

```
      \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
   }%
}
\newabbreviationstyle{short-sc-long-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},
    sort={\glsxtrshortlongdescsort},
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvscfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-long}%
}
\newabbreviationstyle{short-sc}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortnolongname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},
    firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},
    text={\protect\glsabbrvscfont{\the\glsshorttok}},
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}},
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
```

```
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \protect\glsfirstabbrvscfont{\glsaccessshort{##1}%
        \ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}%
       \ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \protect\glsfirstabbrvscfont{\Glsaccessshort{##1}%
        \ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \protect\glsfirstabbrvscfont{\Glsaccessshortpl{##1}%
        \ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
}
\letabbreviationstyle{short-sc-nolong}{short-sc}
\newabbreviationstyle{short-sc-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortdescname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},
```

1005

```
      firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},
      text={\protect\glsabbrvscfont{\the\glsshorttok}},
      plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
  }%
}
\letabbreviationstyle{short-sc-nolong-desc}{short-sc-desc}
```

```
\newabbreviationstyle{nolong-short-sc}%
{%
  \GlsXtrUseAbbrStyleSetup{short-sc-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-nolong}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glsaccesslongpl{##1}%
     \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}%
      \ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{long-noshort-sc}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}},
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},
    text={\protect\glslongdefaultfont{\the\glslongtok}},
    plural={\protect\glslongdefaultfont{\the\glslongpltok}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
```

```
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
  \ifglsxtrinsertinside \else##2\fi
}%
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
   \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
  \ifglsxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
```

```
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\@glsxtr@deprecated@abbrstyle{long-sc}{long-noshort-sc}
\newabbreviationstyle{long-noshort-sc-desc}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
```

```
    }%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
}
\@glsxtr@deprecated@abbrstyle{long-desc-sc}{long-noshort-sc-desc}
\newabbreviationstyle{short-sc-footnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvscfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
```

```
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
}
\@glsxtr@deprecated@abbrstyle{footnote-sc}{short-sc-footnote}
\newabbreviationstyle{short-sc-footnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
```

```
        sort={\glsxtrfootnotedescsort},
        first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
         \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
            {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
        firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
         \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
            {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
        text={\protect\glsabbrvscfont{\the\glsshorttok}},%
        plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
     }%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-footnote}%
}
\newabbreviationstyle{short-sc-postfootnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvscfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
        {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
  \renewcommand*{\glsxtrsetupfulldefs}{%
    \let\glsxtrifwasfirstuse\@secondoftwo
  }%
```

1012

```
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvscfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
}
\@glsxtr@deprecated@abbrstyle{postfootnote-sc}{short-sc-postfootnote}
\newabbreviationstyle{short-sc-postfootnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
```

```
      sort={\glsxtrfootnotedescsort},
      first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},%
      firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},%
      text={\protect\glsabbrvscfont{\the\glsshorttok}},%
      plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \csdef{glsxtrpostlink\glscategorylabel}{%
        \glsxtrifwasfirstuse
        {%
          \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
          {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
        }%
        {}%
      }%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
    \renewcommand*{\glsxtrsetupfulldefs}{%
      \let\glsxtrifwasfirstuse\@secondoftwo
    }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-postfootnote}%
}
\newcommand*{\glsxtrsmfont}[1]{\textsmaller{#1}}
\newcommand*{\glsabbrvsmfont}{\glsxtrsmfont}
\newcommand*{\glsxtrfirstsmfont}[1]{\glsabbrvsmfont{#1}}
\newcommand*{\glsfirstabbrvsmfont}{\glsxtrfirstsmfont}
\newcommand*{\glsxtrsmsuffix}{\glsxtrabbrvpluralsuffix}
\newabbreviationstyle{long-short-sm}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}},%
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
```

```
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{long-short-sm-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortdescname},
    sort={\glsxtrlongshortdescsort},%
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvsmfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
```

```
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-short-sm}%
}
\newabbreviationstyle{short-sm-long}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
```

```
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
}
\newabbreviationstyle{short-sm-long-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},
    sort={\glsxtrshortlongdescsort},
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sm-long}%
}
\newabbreviationstyle{short-sm}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortnolongname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}},
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}},
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}},
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
```

```
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \protect\glsfirstabbrvsmfont{\glsaccessshort{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}%
     \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstabbrvsmfont{\Glsaccessshort{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstabbrvsmfont{\Glsaccessshortpl{##1}%
      \ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\letabbreviationstyle{short-sm-nolong}{short-sm}
\newabbreviationstyle{short-sm-desc}%
```

```
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortdescname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}},
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}},
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
  }%
```

```
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
}
\letabbreviationstyle{short-sm-nolong-desc}{short-sm-desc}
\newabbreviationstyle{nolong-short-sm}%
{%
  \GlsXtrUseAbbrStyleSetup{short-sm-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sm-nolong}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glsaccesslongpl{##1}%
     \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}%
        \ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{long-noshort-sm}%
{%
  \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}},
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},
    text={\protect\glslongdefaultfont{\the\glslongtok}},
    plural={\protect\glslongdefaultfont{\the\glslongpltok}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
```

```
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
```

```
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\@glsxtr@deprecated@abbrstyle{long-sm}{long-noshort-sm}
\newabbreviationstyle{long-noshort-sm-desc}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
```

```
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\@glsxtr@deprecated@abbrstyle{long-desc-sm}{long-noshort-sm-desc}
\newabbreviationstyle{short-sm-footnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
       {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
       {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
```

```
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
  }%
```

```
}
\@glsxtr@deprecated@abbrstyle{footnote-sm}{short-sm-footnote}
\newabbreviationstyle{short-sm-footnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
    sort={\glsxtrfootnotedescsort},
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sm-footnote}%
}
\newabbreviationstyle{short-sm-postfootnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
        {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
```

```
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
    \renewcommand*{\glsxtrsetupfulldefs}{%
      \let\glsxtrifwasfirstuse\@secondoftwo
    }%
  }%
  {%
    \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
    \renewcommand*{\abbrvpluralsuffix}{\glsxtrsmsuffix}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
```

```
}
\@glsxtr@deprecated@abbrstyle{postfootnote-sm}{short-sm-postfootnote}
\newabbreviationstyle{short-sm-postfootnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
    sort={\glsxtrfootnotedescsort},
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
        {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
  \renewcommand*{\glsxtrsetupfulldefs}{%
    \let\glsxtrifwasfirstuse\@secondoftwo
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sm-postfootnote}%
}
\newcommand*{\glsabbrvemfont}[1]{\emph{#1}}%
\newcommand*{\glsfirstabbrvemfont}[1]{\glsabbrvemfont{#1}}%
\newcommand*{\glsxtremsuffix}{\glsxtrabbrvpluralsuffix}
\newcommand*{\glsfirstlongemfont}[1]{\glslongemfont{#1}}%
\newcommand*{\glslongemfont}[1]{\emph{#1}}%
\newabbreviationstyle{long-short-em}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
```

```
      \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
      text={\protect\glsabbrvemfont{\the\glsshorttok}},%
      plural={\protect\glsabbrvemfont{\the\glsshortpltok}},%
      description={\the\glslongtok}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{long-short-em-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortdescname},
    sort={\glsxtrlongshortdescsort},%
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
```

```
      \protect\glsxtrfullsep{\the\glslabeltok}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
     text={\protect\glsabbrvemfont{\the\glsshorttok}},%
     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}%
   }%
   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
     \glshasattribute{\the\glslabeltok}{regular}%
     {%
       \glssetattribute{\the\glslabeltok}{regular}{false}%
     }%
     {}%
   }%
}%
{%
   \GlsXtrUseAbbrStyleFmts{long-short-em}%
}
\newabbreviationstyle{long-em-short-em}%
{%
   \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
   \renewcommand*{\CustomAbbreviationFields}{%
     name={\glsxtrlongshortname},
     sort={\the\glsshorttok},
     first={\protect\glsfirstlongemfont{\the\glslongtok}%
      \protect\glsxtrfullsep{\the\glslabeltok}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
     firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}%
      \protect\glsxtrfullsep{\the\glslabeltok}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
     text={\protect\glsabbrvemfont{\the\glsshorttok}},%
     plural={\protect\glsabbrvemfont{\the\glsshortpltok}},%
     description={\protect\glslongemfont{\the\glslongtok}}}%
   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
     \glshasattribute{\the\glslabeltok}{regular}%
     {%
       \glssetattribute{\the\glslabeltok}{regular}{false}%
     }%
     {}%
   }%
}%
{%
   \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
   \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
   \renewcommand*{\glsxtrfullformat}[2]{%
     \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi
     \glsxtrfullsep{##1}%
     \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
```

```
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongemfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{long-em-short-em-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortdescname},
    sort={\glsxtrlongshortdescsort},%
    first={\protect\glsfirstlongemfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-em-short-em}%
}
\newabbreviationstyle{short-em-long}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
```

```
          first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
           \protect\glsxtrfullsep{\the\glslabeltok}%
           \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
          firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
           \protect\glsxtrfullsep{\the\glslabeltok}%
           \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
          text={\protect\glsabbrvemfont{\the\glsshorttok}},%
          plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
}
\newabbreviationstyle{short-em-long-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},
```

```
        sort={\glsxtrshortlongdescsort},
        first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
          \protect\glsxtrfullsep{\the\glslabeltok}%
          \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
        firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
          \protect\glsxtrfullsep{\the\glslabeltok}%
          \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
        text={\protect\glsabbrvemfont{\the\glsshorttok}},%
        plural={\protect\glsabbrvemfont{\the\glsshortpltok}}%
     }%
     \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
     }%
}%
{%
   \GlsXtrUseAbbrStyleFmts{short-em-long}%
}
\newabbreviationstyle{short-em-long-em}%
{%
   \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
   \renewcommand*{\CustomAbbreviationFields}{%
     name={\glsxtrshortlongname},
     sort={\the\glsshorttok},
     description={\protect\glslongemfont{\the\glslongtok}},%
     first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \glsxtrparen{\protect\glsfirstlongemfont{\the\glslongtok}}},%
     firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
       \protect\glsxtrfullsep{\the\glslabeltok}%
       \glsxtrparen{\protect\glsfirstlongemfont{\the\glslongpltok}}},%
     text={\protect\glsabbrvemfont{\the\glsshorttok}},%
     plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
   \renewcommand*{\GlsXtrPostNewAbbreviation}{%
     \glshasattribute{\the\glslabeltok}{regular}%
     {%
       \glssetattribute{\the\glslabeltok}{regular}{false}%
     }%
     {}%
   }%
}%
{%
   \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
   \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
   \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
   \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
   \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
```

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongemfont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongemfont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongemfont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongemfont{\glsaccesslongpl{##1}}}%
    }%
}
\newabbreviationstyle{short-em-long-em-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},%
    sort={\glsxtrshortlongdescsort},%
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongemfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlongemfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-long-em}%
}
\newabbreviationstyle{short-em}%
```

1033

```
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortnolongname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}},
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}},
    text={\protect\glsabbrvemfont{\the\glsshorttok}},
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}},
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \protect\glsfirstabbrvemfont{\glsaccessshort{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}%
     \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstabbrvemfont{\Glsaccessshort{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstabbrvemfont{\Glsaccessshortpl{##1}%
        \ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
```

```
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
}
\letabbreviationstyle{short-em-nolong}{short-em}
\newabbreviationstyle{short-em-desc}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortdescname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}},
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}},
    text={\protect\glsabbrvemfont{\the\glsshorttok}},
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
```

```
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
}
\letabbreviationstyle{short-em-nolong-desc}{short-em-desc}
\newabbreviationstyle{nolong-short-em}%
{%
  \GlsXtrUseAbbrStyleSetup{short-em-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-nolong}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glsaccesslongpl{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslong{##1}%
      \ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}%
        \ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{long-noshort-em}%
```

```
{%
  \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}},
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},
    text={\protect\glslongdefaultfont{\the\glslongtok}},
    plural={\protect\glslongdefaultfont{\the\glslongpltok}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
```

```
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\@glsxtr@deprecated@abbrstyle{long-em}{long-noshort-em}
\newabbreviationstyle{long-em-noshort-em}%
{%
  \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongemfont{\the\glslongtok}},
    firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}},
    text={\protect\glslongemfont{\the\glslongtok}},
    plural={\protect\glslongemfont{\the\glslongpltok}},%
    description={\protect\glslongemfont{\the\glslongtok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glslongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
```

```
    \glslongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongemfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongemfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongemfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\newabbreviationstyle{long-em-noshort-em-noreg}%
{%
```

```
    \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
    \GlsXtrUseAbbrStyleSetup{long-em-noshort-em}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
  {%
    \GlsXtrUseAbbrStyleFmts{long-em-noshort-em}%
  }
  \newabbreviationstyle{long-noshort-em-desc}%
  {%
    \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
  }%
  {%
    \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
    \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
    \renewcommand*{\glsxtrsubsequentfmt}[2]{%
      \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
      \ifglsxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
      \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
      \ifglsxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
      \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
      \ifglsxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
      \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
      \ifglsxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
```

```
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
     \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
}
\@glsxtr@deprecated@abbrstyle{long-desc-em}{long-noshort-em-desc}
\newabbreviationstyle{long-em-noshort-em-desc}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortdescname},
    sort={\the\glslongtok},
    first={\protect\glsfirstlongemfont{\the\glslongtok}},
    firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}},
    text={\glslongemfont{\the\glslongtok}},
    plural={\glslongemfont{\the\glslongpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glslongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
    \ifglsxtrinsertinside \else##2\fi
  }%
```

```
    \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
      \glslongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
      \ifglsxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
      \glslongemfont{\Glsaccesslong{##1}\ifglsxtrinsertinside ##2\fi}%
      \ifglsxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
      \glslongemfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside ##2\fi}%
      \ifglsxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstlongemfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstlongemfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstlongemfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
}
\newabbreviationstyle{long-em-noshort-em-desc-noreg}%
```

```
{%
  \GlsXtrUseAbbrStyleSetup{long-em-noshort-em-desc}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-em-noshort-em-desc}%
}
\newabbreviationstyle{short-em-footnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \protect\glsxtrabbrvfootnote{##1}%
      {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
```

```
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \protect\glsxtrabbrvfootnote{##1}%
        {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
  }
\@glsxtr@deprecated@abbrstyle{footnote-em}{short-em-footnote}
\newabbreviationstyle{short-em-footnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
    sort={\glsxtrfootnotedescsort},
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
       {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
     \protect\glsxtrabbrvfootnote{\the\glslabeltok}%
```

```
      {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-footnote}%
}
\newabbreviationstyle{short-em-postfootnote}%
{%
  \glsxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
        {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
  \renewcommand*{\glsxtrsetupfulldefs}{%
    \let\glsxtrifwasfirstuse\@secondoftwo
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtremsuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
```

1045

```
    \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
       \ifglsxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
      \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
    }%
}
\@glsxtr@deprecated@abbrstyle{postfootnote-em}{short-em-postfootnote}
\newabbreviationstyle{short-em-postfootnote-desc}%
{%
  \glsxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrfootnotedescname},
    sort={\glsxtrfootnotedescsort},
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
```

1046

```
        \csdef{glsxtrpostlink\glscategorylabel}{%
          \glsxtrifwasfirstuse
          {%
            \glsxtrdopostpunc{\protect\glsxtrabbrvfootnote{\glslabel}%
            {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
          }%
          {}%
        }%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
    }%
    \renewcommand*{\glsxtrsetupfulldefs}{%
      \let\glsxtrifwasfirstuse\@secondoftwo
    }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-postfootnote}%
}
\newcommand*{\glsxtruserfield}{useri}
\ifdef\glscurrentfieldvalue
{
  \newcommand*{\glsxtruserparen}[2]{%
    \glsxtrfullsep{#2}%
    \glsxtrparen
     {#1\ifglshasfield{\glsxtruserfield}{#2}{, \glscurrentfieldvalue}{}}%
  }
}
{
  \newcommand*{\glsxtruserparen}[2]{%
    \glsxtrfullsep{#2}%
    \glsxtrparen
      {#1\ifglshasfield{\glsxtruserfield}{#2}{, \@glo@thisvalue}{}}%
  }
}
\newcommand*{\glsabbrvuserfont}[1]{\glsabbrvdefaultfont{#1}}
\newcommand*{\glsfirstabbrvuserfont}[1]{\glsabbrvuserfont{#1}}
\newcommand*{\glslonguserfont}[1]{\glslongdefaultfont{#1}}
\newcommand*{\glsfirstlonguserfont}[1]{\glslonguserfont{#1}}
\newcommand*{\glsxtrusersuffix}{\glsxtrabbrvpluralsuffix}
\newcommand*{\glsuserdescription}[2]{\glslonguserfont{#1}}
\newabbreviationstyle{long-short-user}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlonguserfont{\the\glslongtok}%
```

```
      \protect\glsxtruserparen{\protect\glsfirstabbrvuserfont{\the\glsshorttok}}%
       {\the\glslabeltok}},%
      firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}%
       \protect\glsxtruserparen
        {\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%
      text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
      plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
      description={\protect\glsuserdescription{\the\glslongtok}%
       {\the\glslabeltok}}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
  }%
}
\newabbreviationstyle{long-postshort-user}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
```

```
        first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
        firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
        text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
        plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
        description={\protect\glsuserdescription{\the\glslongtok}%
          {\the\glslabeltok}}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \csdef{glsxtrpostlink\glscategorylabel}{%
          \glsxtrifwasfirstuse
          {%
            \glsxtruserparen
              {\glsfirstabbrvuserfont{\glsentryshort{\glslabel}}}%
              {\glslabel}%
          }%
          {}%
        }%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
    }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
  }%
```

```
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
    }%
}
\newcommand*{\glsabbrvscuserfont}{\glsabbrvscfont}%
\newcommand*{\glsfirstabbrvscuserfont}{\glsabbrvscuserfont}%
\newcommand*{\glsxtrscusersuffix}{\glsxtrscsuffix}
\newcommand*{\glsxtrlongshortscusername}{%
  \protect\glsabbrvscuserfont{\the\glsshorttok}%
}
\newabbreviationstyle{long-postshort-sc-user}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortscusername},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
    text={\protect\glsabbrvscuserfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscuserfont{\the\glsshortpltok}},%
    description={\protect\glsuserdescription{\the\glslongtok}%
      {\the\glslabeltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtruserparen
          {\glsfirstabbrvscuserfont{\glsentryshort{\glslabel}}}%
          {\glslabel}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
```

1050

```
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscusersuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscuserfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscuserfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvscuserfont{\glsaccessshort{##1}}}{##1}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvscuserfont{\glsaccessshortpl{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvscuserfont{\glsaccessshort{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvscuserfont{\glsaccessshortpl{##1}}}{##1}%
  }%
}
\newcommand*{\glsxtrlongshortuserdescname}{%
  \protect\glslonguserfont{\the\glslongtok}%
  \protect\glsxtruserparen
   {\protect\glsabbrvuserfont{\the\glsshorttok}}{\the\glslabeltok}%
}
\newabbreviationstyle{long-postshort-user-desc}%
```

1051

```
      {%
        \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
        \renewcommand*{\CustomAbbreviationFields}{%
          name={\glsxtrlongshortuserdescname},
          sort={\the\glslongtok},
          first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
          firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
          text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
          plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}%
        }%
        \renewcommand*{\GlsXtrPostNewAbbreviation}{%
          \csdef{glsxtrpostlink\glscategorylabel}{%
            \glsxtrifwasfirstuse
            {%
              \glsxtruserparen
                {\glsfirstabbrvuserfont{\glsentryshort{\glslabel}}}%
                {\glslabel}%
            }%
            {}%
          }%
          \glshasattribute{\the\glslabeltok}{regular}%
          {%
            \glssetattribute{\the\glslabeltok}{regular}{false}%
          }%
          {}%
        }%
      }%
      {%
        \GlsXtrUseAbbrStyleFmts{long-postshort-user}%
      }
      \newcommand*{\glsxtrlongshortscuserdescname}{%
        \protect\glslonguserfont{\the\glslongtok}%
        \protect\glsxtruserparen
          {\protect\glsabbrvscuserfont{\the\glsshorttok}}{\the\glslabeltok}%
      }
      \newabbreviationstyle{long-postshort-sc-user-desc}%
      {%
        \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
        \renewcommand*{\CustomAbbreviationFields}{%
          name={\glsxtrlongshortscuserdescname},
          sort={\the\glslongtok},
          first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
          firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
          text={\protect\glsabbrvscuserfont{\the\glsshorttok}},%
          plural={\protect\glsabbrvscuserfont{\the\glsshortpltok}}%
        }%
        \renewcommand*{\GlsXtrPostNewAbbreviation}{%
          \csdef{glsxtrpostlink\glscategorylabel}{%
            \glsxtrifwasfirstuse
            {%
```

```
        \glsxtruserparen
          {\glsfirstabbrvscuserfont{\glsentryshort{\glslabel}}}%
            {\glslabel}%
        }%
        {}%
      }%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-postshort-sc-user}%
}
\newabbreviationstyle{short-postlong-user}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
    text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvuserfont{\the\glsshortpltok}},%
    description={\protect\glsuserdescription{\the\glslongtok}%
     {\the\glslabeltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtruserparen
          {\glsfirstlonguserfont{\glsentrylong{\glslabel}}}%
            {\glslabel}%
        }%
        {}%
      }%
      \glshasattribute{\the\glslabeltok}{regular}%
      {%
        \glssetattribute{\the\glslabeltok}{regular}{false}%
      }%
      {}%
    }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
```

```
    \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
    \renewcommand*{\glsxtrfullformat}[2]{%
      \glsfirstabbrvuserfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrfullplformat}[2]{%
      \glsfirstabbrvuserfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvuserfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvuserfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
    }%
}
\newcommand*{\glsxtrshortlonguserdescname}{%
 \protect\glsabbrvuserfont{\the\glsshorttok}%
 \protect\glsxtruserparen
   {\protect\glslonguserfont{\the\glslongpltok}}%
   {\the\glslabeltok}%
}
\newabbreviationstyle{short-postlong-user-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlonguserdescname},
    sort={\the\glsshorttok},
```

```
          first={\protect\glsfirstlonguserfont{\the\glslongtok}},%
          firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}},%
          text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
          plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}%
      }%
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
          \csdef{glsxtrpostlink\glscategorylabel}{%
              \glsxtrifwasfirstuse
              {%
                  \glsxtruserparen
                      {\glsfirstlonguserfont{\glsentrylong{\glslabel}}}%
                      {\glslabel}%
              }%
              {}%
          }%
          \glshasattribute{\the\glslabeltok}{regular}%
          {%
              \glssetattribute{\the\glslabeltok}{regular}{false}%
          }%
          {}%
      }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-postlong-user}%
}
\newabbreviationstyle{long-short-user-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortuserdescname},
    sort={\glsxtrlongshortdescsort},%
    first={\protect\glsfirstlonguserfont{\the\glslongtok}%
     \protect\glsxtruserparen{\protect\glsfirstabbrvuserfont{\the\glsshorttok}}%
       {\the\glslabeltok}},%
    firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}%
     \protect\glsxtruserparen
       {\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-short-user}%
```

```
}
\newabbreviationstyle{short-long-user}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    description={\protect\glsuserdescription{\the\glslongtok}%
      {\the\glslabeltok}},%
    first={\protect\glsfirstabbrvuserfont{\the\glsshorttok}%
     \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongtok}}%
        {\the\glslabeltok}},%
    firstplural={\protect\glsfirstabbrvuserfont{\the\glsshortpltok}%
     \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongpltok}}%
        {\the\glslabeltok}},%
    text={\protect\glsabbrvuserfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvuserfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrusersuffix}%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvuserfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstabbrvuserfont{\glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstabbrvuserfont{\glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslong{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
    \glsxtruserparen{\glsfirstlonguserfont{\glsaccesslongpl{##1}}}{##1}%
```

```
    }%
}
\newabbreviationstyle{short-long-user-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlonguserdescname},
    sort={\glsxtrshortlongdescsort},%
    first={\protect\glsfirstabbrvuserfont{\the\glsshorttok}%
     \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongtok}}%
        {\the\glslabeltok}},%
    firstplural={\protect\glsfirstabbrvuserfont{\the\glsshortpltok}%
     \protect\glsxtruserparen{\protect\glsfirstlonguserfont{\the\glslongpltok}}%
        {\the\glslabeltok}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-long-user}%
}
\newrobustcmd*{\glsxtrifhyphenstart}[3]{%
  \ifx\glsinsert#1\relax
   \expandafter\@glsxtrifhyphenstart#1\relax\relax
     \@end@glsxtrifhyphenstart{#2}{#3}%
  \else
   \@glsxtrifhyphenstart#1\relax\relax\@end@glsxtrifhyphenstart{#2}{#3}%
  \fi
}
\def\@glsxtrifhyphenstart#1#2\@end@glsxtrifhyphenstart#3#4{%
  \ifx-#1\relax#3\else #4\fi
}
\newcommand*{\glsxtrlonghyphenshort}[4]{%
 {%
   \glsxtrifhyphenstart{#4}{\def\glsxtrwordsep{-}}{}%
   \glsfirstlonghyphenfont{#2\ifglsxtrinsertinside{#4}\fi}%
   \ifglsxtrinsertinside\else{#4}\fi
   \glsxtrfullsep{#1}%
   \glsxtrparen{\glsfirstabbrvhyphenfont{#3\ifglsxtrinsertinside{#4}\fi}%
    \ifglsxtrinsertinside\else{#4}\fi}%
 }%
}
\newcommand*{\glsabbrvhyphenfont}{\glsabbrvdefaultfont}%
```

```
\newcommand*{\glsfirstabbrvhyphenfont}{\glsabbrvhyphenfont}%
\newcommand*{\glslonghyphenfont}{\glslongdefaultfont}%
\newcommand*{\glsfirstlonghyphenfont}{\glslonghyphenfont}%
\newcommand*{\glsxtrhyphensuffix}{\glsxtrabbrvpluralsuffix}
\newabbreviationstyle{long-hyphen-short-hyphen}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlonghyphenfont{\the\glslongtok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
    description={\protect\glslonghyphenfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrhyphensuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrlonghyphenshort{##1}{\glsaccesslong{##1}}{\glsaccessshort{##1}}{##2}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsxtrlonghyphenshort{##1}{\glsaccesslongpl{##1}}%
      {\glsaccessshortpl{##1}}{##2}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsxtrlonghyphenshort{##1}{\Glsaccesslong{##1}}{\glsaccessshort{##1}}{##2}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsxtrlonghyphenshort{##1}{\Glsaccesslongpl{##1}}%
      {\glsaccessshortpl{##1}}{##2}%
  }%
}
\newabbreviationstyle{long-hyphen-short-hyphen-desc}%
{%
```

```
      \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
      \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtrlongshortdescname},
        sort={\glsxtrlongshortdescsort},
        first={\protect\glsfirstlonghyphenfont{\the\glslongtok}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}}},%
        firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}%
         \protect\glsxtrfullsep{\the\glslabeltok}%
         \glsxtrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}}},%
        text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
        plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
      }%
      \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
          \glssetattribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
      }%
  }%
  {%
    \GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%
  }
\newcommand*{\glsxtrlonghyphennoshort}[3]{%
 {%
    \glsxtrifhyphenstart{#3}{\def\glsxtrwordsep{-}}{}%
    \glsfirstlonghyphenfont{#2\ifglsxtrinsertinside{#3}\fi}%
    \ifglsxtrinsertinside\else{#3}\fi
 }%
}
\newabbreviationstyle{long-hyphen-noshort-desc-noreg}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortdescname},
    sort={\expandonce\glsxtrorglong},
    first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
    text={\protect\glslonghyphenfont{\the\glslongtok}},%
    plural={\protect\glslonghyphenfont{\the\glslongpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
```

```
\GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%
\renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
\renewcommand*\glsabbrvfont[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
\renewcommand*{\glsxtrsubsequentfmt}[2]{%
  \glsxtrlonghyphennoshort{##1}{\glsaccesslong{##1}}{##2}%
}%
\renewcommand*{\glsxtrsubsequentplfmt}[2]{%
  \glsxtrlonghyphennoshort{##1}{\glsaccesslongpl{##1}}{##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glsxtrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glsxtrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
}%
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\glsaccesslong{##1}}{##2}%
  \glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\glsaccesslongpl{##1}}{##2}%
  \glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
  \glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
  \glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\glsaccesslong{##1}}{##2}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\glsaccesslongpl{##1}}{##2}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\Glsaccesslong{##1}}{##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsxtrlonghyphennoshort{##1}{\Glsaccesslongpl{##1}}{##2}%
}%
```

```
}
\newabbreviationstyle{long-hyphen-noshort-noreg}%
{%
  \glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongnoshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
    text={\protect\glslonghyphenfont{\the\glslongtok}},%
    plural={\protect\glslonghyphenfont{\the\glslongpltok}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-noshort-desc-noreg}%
}
\newcommand*{\glsxtrlonghyphen}[3]{%
 {%
    \glsxtrifhyphenstart{#3}{\def\glsxtrwordsep{-}}{}%
    \glsfirstlonghyphenfont{#1}%
 }%
}
\newcommand*{\glsxtrposthyphenshort}[2]{%
 {%
    \glsxtrifhyphenstart{#2}{\def\glsxtrwordsep{-}}{}%
    \ifglsxtrinsertinside{\glsfirstlonghyphenfont{#2}}\else{#2}\fi
    \glsxtrfullsep{#1}%
    \glsxtrparen
    {\glsfirstabbrvhyphenfont{\glsentryshort{#1}\ifglsxtrinsertinside{#2}\fi}%
     \ifglsxtrinsertinside\else{#2}\fi
    }%
 }%
}
\newcommand*{\glsxtrposthyphensubsequent}[2]{%
  \glsabbrvfont{\ifglsxtrinsertinside {#2}\fi}%
  \ifglsxtrinsertinside \else{#2}\fi
}
\newabbreviationstyle{long-hyphen-postshort-hyphen}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortname},
```

```
      sort={\the\glsshorttok},
      first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
      firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
      text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
      plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
      description={\protect\glslonghyphenfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrposthyphenshort{\glslabel}{\glsinsert}%
      }%
      {%
        \glsxtrposthyphensubsequent{\glslabel}{\glsinsert}%
      }%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsabbrvfont{\glsaccessshort{##1}}%
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glsabbrvfont{\glsaccessshortpl{##1}}%
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glsabbrvfont{\Glsaccessshort{##1}}%
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glsabbrvfont{\Glsaccessshortpl{##1}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrlonghyphen{\glsaccesslong{##1}}{##1}{##2}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsxtrlonghyphen{\glsaccesslongpl{##1}}{##1}{##2}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsxtrlonghyphen{\Glsaccesslong{##1}}{##1}{##2}%
  }%
```

```
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsxtrlonghyphen{\Glsaccesslongpl{##1}}{##1}{##2}%
}%
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsfirstlonghyphenfont{\glsaccesslong{##1}%
    \ifglsxtrinsertinside{##2}\fi}%
  \ifglsxtrinsertinside \else{##2}\fi
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsfirstlonghyphenfont{\glsaccesslongpl{##1}%
    \ifglsxtrinsertinside{##2}\fi}%
  \ifglsxtrinsertinside \else{##2}\fi
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstlonghyphenfont{\Glsaccesslong{##1}%
    \ifglsxtrinsertinside{##2}\fi}%
  \ifglsxtrinsertinside \else{##2}\fi
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstlonghyphenfont{\Glsaccesslongpl{##1}%
    \ifglsxtrinsertinside{##2}\fi}%
  \ifglsxtrinsertinside \else{##2}\fi
}%
}
\newabbreviationstyle{long-hyphen-postshort-hyphen-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrlongshortdescname},
    sort={\glsxtrlongshortdescsort},%
    first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrposthyphenshort{\glslabel}{\glsinsert}%
      }%
      {%
        \glsxtrposthyphensubsequent{\glslabel}{\glsinsert}%
      }%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
```

```
    }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-postshort-hyphen}%
}
\newcommand*{\glsxtrshorthyphenlong}[4]{%
 {%
    \glsxtrifhyphenstart{#4}{\def\glsxtrwordsep{-}}{}%
    \glsfirstabbrvhyphenfont{#2\ifglsxtrinsertinside{#4}\fi}%
    \ifglsxtrinsertinside\else{#4}\fi
    \glsxtrfullsep{#1}%
    \glsxtrparen{\glsfirstlonghyphenfont{#3\ifglsxtrinsertinside{#4}\fi}%
     \ifglsxtrinsertinside\else{#4}\fi}%
 }%
}
\newabbreviationstyle{short-hyphen-long-hyphen}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
    description={\protect\glslonghyphenfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrhyphensuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrshorthyphenlong{##1}{\glsaccessshort{##1}}{\glsaccesslong{##1}}{##2}%
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsxtrshorthyphenlong{##1}%
     {\glsaccessshortpl{##1}}{\glsaccesslongpl{##1}}{##2}%
```

```
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsxtrshorthyphenlong{##1}{\glsaccessshort{##1}}{\Glsaccesslong{##1}}{##2}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsxtrshorthyphenlong{##1}%
     {\glsaccessshortpl{##1}}{\Glsaccesslongpl{##1}}{##2}%
  }%
}
\newabbreviationstyle{short-hyphen-long-hyphen-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},
    sort={\glsxtrshortlongdescsort},
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
     \protect\glsxtrfullsep{\the\glslabeltok}%
     \glsxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-hyphen-long-hyphen}%
}
\newcommand*{\glsxtrshorthyphen}[3]{%
 {%
   \glsxtrifhyphenstart{#3}{\def\glsxtrwordsep{-}}{}%
   \glsfirstabbrvhyphenfont{#1}%
 }%
}
\newcommand*{\glsxtrposthyphenlong}[2]{%
 {%
   \glsxtrifhyphenstart{#2}{\def\glsxtrwordsep{-}}{}%
   \ifglsxtrinsertinside{\glsfirstabbrvhyphenfont{#2}}\else{#2}\fi
   \glsxtrfullsep{#1}%
   \glsxtrparen
   {\glsfirstlonghyphenfont{\glsentrylong{#1}\ifglsxtrinsertinside{#2}\fi}%
    \ifglsxtrinsertinside\else{#2}\fi
   }%
```

```
 }%
}
\newabbreviationstyle{short-hyphen-postlong-hyphen}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
    description={\protect\glslonghyphenfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrposthyphenlong{\glslabel}{\glsinsert}%
      }%
      {%
        \glsxtrposthyphensubsequent{\glslabel}{\glsinsert}%
      }%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrabbrvpluralsuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glsabbrvfont{\glsaccessshort{##1}}%
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glsabbrvfont{\glsaccessshortpl{##1}}%
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glsabbrvfont{\Glsaccessshort{##1}}%
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glsabbrvfont{\Glsaccessshortpl{##1}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsxtrshorthyphen{\glsaccessshort{##1}}{##1}{##2}%
```

```
    }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsxtrshorthyphen{\glsaccessshortpl{##1}}{##1}{##2}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsxtrshorthyphen{\Glsaccessshort{##1}}{##1}{##2}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsxtrshorthyphen{\Glsaccessshortpl{##1}}{##1}{##2}%
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvhyphenfont{\glsaccessshort{##1}%
      \ifglsxtrinsertinside{##2}\fi}%
    \ifglsxtrinsertinside \else{##2}\fi
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvhyphenfont{\glsaccessshortpl{##1}%
      \ifglsxtrinsertinside{##2}\fi}%
    \ifglsxtrinsertinside \else{##2}\fi
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvhyphenfont{\Glsaccessshort{##1}%
      \ifglsxtrinsertinside{##2}\fi}%
    \ifglsxtrinsertinside \else{##2}\fi
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvhyphenfont{\Glsaccessshortpl{##1}%
      \ifglsxtrinsertinside{##2}\fi}%
    \ifglsxtrinsertinside \else{##2}\fi
  }%
}
\newabbreviationstyle{short-hyphen-postlong-hyphen-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrshortlongdescname},
    sort={\glsxtrshortlongdescsort},%
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glsxtrpostlink\glscategorylabel}{%
      \glsxtrifwasfirstuse
      {%
        \glsxtrposthyphenlong{\glslabel}{\glsinsert}%
      }%
      {%
        \glsxtrposthyphensubsequent{\glslabel}{\glsinsert}%
```

```
      }%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-hyphen-postlong-hyphen}%
}
\newcommand*{\glsabbrvonlyfont}{\glsabbrvdefaultfont}%
\newcommand*{\glsfirstabbrvonlyfont}{\glsabbrvonlyfont}%
\newcommand*{\glslongonlyfont}{\glslongdefaultfont}%
\newcommand*{\glsfirstlongonlyfont}{\glslongonlyfont}%
\newcommand*{\glsxtronlysuffix}{\glsxtrabbrvpluralsuffix}
\newcommand*{\glsxtronlyname}{%
  \protect\glsabbrvonlyfont{\the\glsshorttok}%
}
\newabbreviationstyle{long-only-short-only}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtronlyname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
    text={\protect\glsabbrvonlyfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvonlyfont{\the\glsshortpltok}},%
    description={\protect\glslongonlyfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtronlysuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvonlyfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvonlyfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongonlyfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
```

1068

```
        \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
        \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
        \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
        \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
        \glsxtrfullsep{##1}%
        \glsxtrparen{\protect\glsfirstabbrvonlyfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
        \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
        \glsxtrfullsep{##1}%
        \glsxtrparen{\protect\glsfirstabbrvonlyfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
        \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
        \glsxtrfullsep{##1}%
        \glsxtrparen{\protect\glsfirstabbrvonlyfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
        \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
        \glsxtrfullsep{##1}%
        \glsxtrparen{\protect\glsfirstabbrvonlyfont{\Glsaccessshortpl{##1}}}%
    }%
}
\newcommand*{\glsxtronlydescsort}{\the\glslongtok}
\newcommand*{\glsxtronlydescname}{%
    \protect\glslongfont{\the\glslongtok}%
}
\newabbreviationstyle{long-only-short-only-desc}%
{%
    \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtronlydescname},
        sort={\glsxtronlydescsort},%
        first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
        firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
        text={\protect\glsabbrvonlyfont{\the\glsshorttok}},%
        plural={\protect\glsabbrvonlyfont{\the\glsshortpltok}}%
```

```
    }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-only-short-only}%
}
\newcommand*{\glsabbrvsconlyfont}{\glsabbrvscfont}%
\newcommand*{\glsfirstabbrvsconlyfont}{\glsabbrvsconlyfont}%
\newcommand*{\glsxtrsconlysuffix}{\glsxtrscsuffix}
\newcommand*{\glsxtrsconlyname}{%
  \protect\glsabbrvsconlyfont{\the\glsshorttok}%
}
\newabbreviationstyle{long-only-short-sc-only}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrsconlyname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
    text={\protect\glsabbrvsconlyfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsconlyfont{\the\glsshortpltok}},%
    description={\protect\glslongonlyfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrsconlysuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvsconlyfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvsconlyfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongonlyfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
    \ifglsxtrinsertinside\else##2\fi
```

```
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
      \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
      \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glsxtrinlinefullformat}[2]{%
      \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvsconlyfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\glsxtrinlinefullplformat}[2]{%
      \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvsconlyfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
      \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvsconlyfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
      \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglsxtrinsertinside##2\fi}%
      \ifglsxtrinsertinside\else##2\fi
      \glsxtrfullsep{##1}%
      \glsxtrparen{\protect\glsfirstabbrvsconlyfont{\Glsaccessshortpl{##1}}}%
    }%
}
\newcommand*{\glsxtrsconlydescsort}{\glsxtronlydescsort}
\newcommand*{\glsxtrsconlydescname}{\glsxtronlydescname}
\newabbreviationstyle{long-only-short-sc-only-desc}%
{%
  \glsxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrsconlydescname},
    sort={\glsxtrsconlydescsort},%
    first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
    text={\protect\glsabbrvsconlyfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsconlyfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
```

```
      \glssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-only-short-sc-only}%
}
\let\@glsxtr@org@markright\markright
\renewcommand*{\markright}[1]{%
 \glsxtrmarkhook
 \@glsxtr@org@markright{\@glsxtrinmark#1\@glsxtrnotinmark}%
 \glsxtrrestoremarkhook
}
\let\@glsxtr@org@markboth\markboth
\renewcommand*{\markboth}[2]{%
 \glsxtrmarkhook
 \@glsxtr@org@markboth
   {\@glsxtrinmark#1\@glsxtrnotinmark}%
   {\@glsxtrinmark#2\@glsxtrnotinmark}%
 \glsxtrrestoremarkhook
}
\let\@glsxtr@org@@starttoc\@starttoc
\renewcommand*{\@starttoc}[1]{%
 \glsxtrmarkhook
 \@glsxtrinmark
 \@glsxtr@org@@starttoc{#1}%
 \@glsxtrnotinmark
 \glsxtrrestoremarkhook
}
\newcommand*{\glsxtrRevertMarks}{%
  \let\markright\@glsxtr@org@markright
  \let\markboth\@glsxtr@org@markboth
  \let\@starttoc\@glsxtr@org@@starttoc
}
\newcommand*{\glsxtrRevertTocMarks}{%
  \let\@starttoc\@glsxtr@org@@starttoc
}
\newcommand*{\glsxtrifinmark}[2]{#2}
\newrobustcmd*{\@glsxtrinmark}{%
  \let\glsxtrifinmark\@firstoftwo
}
\newrobustcmd*{\@glsxtrnotinmark}{%
  \let\glsxtrifinmark\@secondoftwo
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsxtrtitleorpdforheading}[3]{\texorpdfstring{#1}{#2}}
}
{
```

```
    \newcommand*{\glsxtrtitleorpdforheading}[3]{#1}
}
\newcommand*{\glsxtrmarkhook}{%
  \let\@glsxtr@org@MakeUppercase\MakeUppercase
  \let\@glsxtr@org@glsxtrtitleorpdforheading\glsxtrtitleorpdforheading
  \let\@glsxtr@org@glsxtrtitleshort\glsxtrtitleshort
  \let\@glsxtr@org@glsxtrtitleshortpl\glsxtrtitleshortpl
  \let\@glsxtr@org@Glsxtrtitleshort\Glsxtrtitleshort
  \let\@glsxtr@org@Glsxtrtitleshortpl\Glsxtrtitleshortpl
  \let\@glsxtr@org@glsxtrtitlename\glsxtrtitlename
  \let\@glsxtr@org@Glsxtrtitlename\Glsxtrtitlename
  \let\@glsxtr@org@glsxtrtitletext\glsxtrtitletext
  \let\@glsxtr@org@Glsxtrtitletext\Glsxtrtitletext
  \let\@glsxtr@org@glsxtrtitleplural\glsxtrtitleplural
  \let\@glsxtr@org@Glsxtrtitleplural\Glsxtrtitleplural
  \let\@glsxtr@org@glsxtrtitlefirst\glsxtrtitlefirst
  \let\@glsxtr@org@Glsxtrtitlefirst\Glsxtrtitlefirst
  \let\@glsxtr@org@glsxtrtitlefirstplural\glsxtrtitlefirstplural
  \let\@glsxtr@org@Glsxtrtitlefirstplural\Glsxtrtitlefirstplural
  \let\@glsxtr@org@glsxtrtitlelong\glsxtrtitlelong
  \let\@glsxtr@org@glsxtrtitlelongpl\glsxtrtitlelongpl
  \let\@glsxtr@org@Glsxtrtitlelong\Glsxtrtitlelong
  \let\@glsxtr@org@Glsxtrtitlelongpl\Glsxtrtitlelongpl
  \let\@glsxtr@org@glsxtrtitlefull\glsxtrtitlefull
  \let\@glsxtr@org@glsxtrtitlefullpl\glsxtrtitlefullpl
  \let\@glsxtr@org@Glsxtrtitlefull\Glsxtrtitlefull
  \let\@glsxtr@org@Glsxtrtitlefullpl\Glsxtrtitlefullpl
  \let\glsxtrifinmark\@firstoftwo
  \let\MakeUppercase\MakeTextUppercase
  \let\glsxtrtitleorpdforheading\@thirdofthree
  \let\glsxtrtitleshort\glsxtrheadshort
  \let\glsxtrtitleshortpl\glsxtrheadshortpl
  \let\Glsxtrtitleshort\Glsxtrheadshort
  \let\Glsxtrtitleshortpl\Glsxtrheadshortpl
  \let\glsxtrtitlename\glsxtrheadname
  \let\Glsxtrtitlename\Glsxtrheadname
  \let\glsxtrtitletext\glsxtrheadtext
  \let\Glsxtrtitletext\Glsxtrheadtext
  \let\glsxtrtitleplural\glsxtrheadplural
  \let\Glsxtrtitleplural\Glsxtrheadplural
  \let\glsxtrtitlefirst\glsxtrheadfirst
  \let\Glsxtrtitlefirst\Glsxtrheadfirst
  \let\glsxtrtitlefirstplural\glsxtrheadfirstplural
  \let\Glsxtrtitlefirstplural\Glsxtrheadfirstplural
  \let\glsxtrtitlelong\glsxtrheadlong
  \let\glsxtrtitlelongpl\glsxtrheadlongpl
  \let\Glsxtrtitlelong\Glsxtrheadlong
  \let\Glsxtrtitlelongpl\Glsxtrheadlongpl
  \let\glsxtrtitlefull\glsxtrheadfull
  \let\glsxtrtitlefullpl\glsxtrheadfullpl
```

```
    \let\Glsxtrtitlefull\Glsxtrheadfull
    \let\Glsxtrtitlefullpl\Glsxtrheadfullpl
}
\newcommand*{\glsxtrrestoremarkhook}{%
  \let\glsxtrifinmark\@secondoftwo
  \let\MakeUppercase\@glsxtr@org@MakeUppercase
  \let\glsxtrtitleorpdforheading\@glsxtr@org@glsxtrtitleorpdforheading
  \let\glsxtrtitleshort\@glsxtr@org@glsxtrtitleshort
  \let\glsxtrtitleshortpl\@glsxtr@org@glsxtrtitleshortpl
  \let\Glsxtrtitleshort\@glsxtr@org@Glsxtrtitleshort
  \let\Glsxtrtitleshortpl\@glsxtr@org@Glsxtrtitleshortpl
  \let\glsxtrtitlename\@glsxtr@org@glsxtrtitlename
  \let\Glsxtrtitlename\@glsxtr@org@Glsxtrtitlename
  \let\glsxtrtitletext\@glsxtr@org@glsxtrtitletext
  \let\Glsxtrtitletext\@glsxtr@org@Glsxtrtitletext
  \let\glsxtrtitleplural\@glsxtr@org@glsxtrtitleplural
  \let\Glsxtrtitleplural\@glsxtr@org@Glsxtrtitleplural
  \let\glsxtrtitlefirst\@glsxtr@org@glsxtrtitlefirst
  \let\Glsxtrtitlefirst\@glsxtr@org@Glsxtrtitlefirst
  \let\glsxtrtitlefirstplural\@glsxtr@org@glsxtrtitlefirstplural
  \let\Glsxtrtitlefirstplural\@glsxtr@org@Glsxtrtitlefirstplural
  \let\glsxtrtitlelong\@glsxtr@org@glsxtrtitlelong
  \let\glsxtrtitlelongpl\@glsxtr@org@glsxtrtitlelongpl
  \let\Glsxtrtitlelong\@glsxtr@org@Glsxtrtitlelong
  \let\Glsxtrtitlelongpl\@glsxtr@org@Glsxtrtitlelongpl
  \let\glsxtrtitlefull\@glsxtr@org@glsxtrtitlefull
  \let\glsxtrtitlefullpl\@glsxtr@org@glsxtrtitlefullpl
  \let\Glsxtrtitlefull\@glsxtr@org@Glsxtrtitlefull
  \let\Glsxtrtitlefullpl\@glsxtr@org@Glsxtrtitlefullpl
}
\newcommand*{\glsxtrheadshort}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrshort[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsxtrshort[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitleshort}[1]{%
  \glsxtrshort[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadshortpl}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
```

```
      \GLSxtrshortpl[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsxtrshortpl[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitleshortpl}[1]{%
  \glsxtrshortpl[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadshort}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrshort[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsxtrshort[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitleshort}[1]{%
  \Glsxtrshort[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitleshort}[1]{%
  \GLSxtrshort[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadshortpl}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrshortpl[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsxtrshortpl[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitleshortpl}[1]{%
  \Glsxtrshortpl[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitleshortpl}[1]{%
  \GLSxtrshortpl[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadname}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
```

```
    {%
      \GLSname[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsname[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitlename}[1]{%
  \glsname[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadname}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSname[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsname[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitlename}[1]{%
  \Glsname[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitlename}[1]{%
  \GLSname[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadtext}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLStext[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glstext[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitletext}[1]{%
  \glstext[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadtext}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLStext[noindex,hyper=false]{#1}[]%
```

```
    }%
    {%
      \Glstext[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitletext}[1]{%
  \Glstext[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitletext}[1]{%
  \GLStext[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadplural}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSplural[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsplural[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitleplural}[1]{%
  \glsplural[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadplural}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSplural[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsplural[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitleplural}[1]{%
  \Glsplural[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitleplural}[1]{%
  \GLSplural[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadfirst}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
```

```
        \GLSfirst[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsfirst[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitlefirst}[1]{%
  \glsfirst[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadfirst}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSfirst[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsfirst[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitlefirst}[1]{%
  \Glsfirst[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitlefirst}[1]{%
  \GLSfirst[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadfirstplural}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSfirstplural[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsfirstplural[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitlefirstplural}[1]{%
  \glsfirstplural[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadfirstplural}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSfirstplural[noindex,hyper=false]{#1}[]%
    }%
```

```
    {%
      \Glsfirstplural[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitlefirstplural}[1]{%
  \Glsfirstplural[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitlefirstplural}[1]{%
  \GLSfirstplural[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadlong}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrlong[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsxtrlong[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitlelong}[1]{%
  \glsxtrlong[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadlongpl}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsxtrlongpl[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitlelongpl}[1]{%
  \glsxtrlongpl[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadlong}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrlong[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsxtrlong[noindex,hyper=false]{#1}[]%
```

```
    }%
  }%
}
\newrobustcmd*{\Glsxtrtitlelong}[1]{%
  \Glsxtrlong[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitlelong}[1]{%
  \GLSxtrlong[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadlongpl}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsxtrlongpl[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitlelongpl}[1]{%
  \Glsxtrlongpl[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitlelongpl}[1]{%
  \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadfull}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrfull[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glsxtrfull[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitlefull}[1]{%
  \glsxtrfull[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glsxtrheadfullpl}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrfullpl[noindex,hyper=false]{#1}[]%
    }%
    {%
```

```
      \glsxtrfullpl[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\glsxtrtitlefullpl}[1]{%
  \glsxtrfullpl[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadfull}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrfull[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsxtrfull[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitlefull}[1]{%
  \Glsxtrfull[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitlefull}[1]{%
  \GLSxtrfull[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadfullpl}[1]{%
 \protect\NoCaseChange
 {%
    \glsifattribute{#1}{headuc}{true}%
    {%
      \GLSxtrfullpl[noindex,hyper=false]{#1}[]%
    }%
    {%
      \Glsxtrfullpl[noindex,hyper=false]{#1}[]%
    }%
 }%
}
\newrobustcmd*{\Glsxtrtitlefullpl}[1]{%
  \Glsxtrfullpl[noindex,hyper=false]{#1}[]%
}
\newrobustcmd*{\GLSxtrtitlefullpl}[1]{%
  \GLSxtrfullpl[noindex,hyper=false]{#1}[]%
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtshort}[1]{%
    \texorpdfstring
      {\glsxtrtitleshort{#1}}%
      {\glsentryshort{#1}}%
  }
```

```
}
{
  \newcommand*{\glsfmtshort}[1]{%
   \glsxtrtitleshort{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtshortpl}[1]{%
    \texorpdfstring
      {\glsxtrtitleshortpl{#1}}%
      {\glsentryshortpl{#1}}%
  }
}
{
  \newcommand*{\glsfmtshortpl}[1]{%
   \glsxtrtitleshortpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtshort}[1]{%
    \texorpdfstring
      {\Glsxtrtitleshort{#1}}%
      {\glsentryshort{#1}}%
  }
}
{
  \newcommand*{\Glsfmtshort}[1]{%
   \Glsxtrtitleshort{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtshortpl}[1]{%
    \texorpdfstring
    {\Glsxtrtitleshortpl{#1}}%
    {\glsentryshortpl{#1}}%
  }
}
{
  \newcommand*{\Glsfmtshortpl}[1]{%
   \Glsxtrtitleshortpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtname}[1]{%
    \texorpdfstring
    {\glsxtrtitlename{#1}}%
    {\glsentryname{#1}}%
  }
}
{
```

```
  \newcommand*{\glsfmtname}[1]{%
    \glsxtrtitlename{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtname}[1]{%
    \texorpdfstring
    {\Glsxtrtitlename{#1}}%
    {\glsentryname{#1}}%
  }
}
{
  \newcommand*{\Glsfmtname}[1]{%
    \Glsxtrtitlename{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmtname}[1]{%
    \texorpdfstring
    {\GLSxtrtitlename{#1}}%
    {\glsentryname{#1}}%
  }
}
{
  \newcommand*{\GLSfmtname}[1]{%
    \GLSxtrtitlename{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmttext}[1]{%
    \texorpdfstring
    {\glsxtrtitletext{#1}}%
    {\glsentrytext{#1}}%
  }
}
{
  \newcommand*{\glsfmttext}[1]{%
    \glsxtrtitletext{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmttext}[1]{%
    \texorpdfstring
    {\Glsxtrtitletext{#1}}%
    {\glsentrytext{#1}}%
  }
}
{
  \newcommand*{\Glsfmttext}[1]{%
    \Glsxtrtitletext{#1}}
```

```
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmttext}[1]{%
    \texorpdfstring
    {\GLSxtrtitletext{#1}}%
    {\glsentrytext{#1}}%
  }
}
{
  \newcommand*{\GLSfmttext}[1]{%
   \GLSxtrtitletext{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtplural}[1]{%
    \texorpdfstring
    {\glsxtrtitleplural{#1}}%
    {\glsentryplural{#1}}%
  }
}
{
  \newcommand*{\glsfmtplural}[1]{%
   \glsxtrtitleplural{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtplural}[1]{%
    \texorpdfstring
    {\Glsxtrtitleplural{#1}}%
    {\glsentryplural{#1}}%
  }
}
{
  \newcommand*{\Glsfmtplural}[1]{%
   \Glsxtrtitleplural{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmtplural}[1]{%
    \texorpdfstring
    {\GLSxtrtitleplural{#1}}%
    {\glsentryplural{#1}}%
  }
}
{
  \newcommand*{\GLSfmtplural}[1]{%
   \GLSxtrtitleplural{#1}}
}
\ifdef\texorpdfstring
```

```
{
  \newcommand*{\glsfmtfirst}[1]{%
    \texorpdfstring
    {\glsxtrtitlefirst{#1}}%
    {\glsentryfirst{#1}}%
  }
}
{
  \newcommand*{\glsfmtfirst}[1]{%
   \glsxtrtitlefirst{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtfirst}[1]{%
    \texorpdfstring
    {\Glsxtrtitlefirst{#1}}%
    {\glsentryfirst{#1}}%
  }
}
{
  \newcommand*{\Glsfmtfirst}[1]{%
   \Glsxtrtitlefirst{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmtfirst}[1]{%
    \texorpdfstring
    {\GLSxtrtitlefirst{#1}}%
    {\glsentryfirst{#1}}%
  }
}
{
  \newcommand*{\GLSfmtfirst}[1]{%
   \Glsxtrtitlefirst{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtfirstpl}[1]{%
    \texorpdfstring
    {\glsxtrtitlefirstplural{#1}}%
    {\glsentryfirstplural{#1}}%
  }
}
{
  \newcommand*{\glsfmtfirstpl}[1]{%
   \glsxtrtitlefirstplural{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtfirstpl}[1]{%
```

```
      \texorpdfstring
      {\Glsxtrtitlefirstplural{#1}}%
      {\glsentryfirstplural{#1}}%
   }
}
{
  \newcommand*{\Glsfmtfirstpl}[1]{%
   \Glsxtrtitlefirstplural{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmtfirstpl}[1]{%
     \texorpdfstring
     {\GLSxtrtitlefirstplural{#1}}%
     {\glsentryfirstplural{#1}}%
   }
}
{
  \newcommand*{\GLSfmtfirstpl}[1]{%
   \GLSxtrtitlefirstplural{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtlong}[1]{%
     \texorpdfstring
     {\glsxtrtitlelong{#1}}%
     {\glsentrylong{#1}}%
   }
}
{
  \newcommand*{\glsfmtlong}[1]{%
   \glsxtrtitlelong{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtlong}[1]{%
     \texorpdfstring
     {\Glsxtrtitlelong{#1}}%
     {\glsentrylong{#1}}%
   }
}
{
  \newcommand*{\Glsfmtlong}[1]{%
   \Glsxtrtitlelong{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmtlong}[1]{%
     \texorpdfstring
     {\GLSxtrtitlelong{#1}}%
```

```
    {\glsentrylong{#1}}%
  }
}
{
  \newcommand*{\GLSfmtlong}[1]{%
   \GLSxtrtitlelong{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtlongpl}[1]{%
   \texorpdfstring
   {\glsxtrtitlelongpl{#1}}%
   {\glsentrylongpl{#1}}%
  }
}
{
  \newcommand*{\glsfmtlongpl}[1]{%
   \glsxtrtitlelongpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtlongpl}[1]{%
   \texorpdfstring
   {\Glsxtrtitlelongpl{#1}}%
   {\glsentrylongpl{#1}}%
  }
}
{
  \newcommand*{\Glsfmtlongpl}[1]{%
   \Glsxtrtitlelongpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmtlongpl}[1]{%
   \texorpdfstring
   {\GLSxtrtitlelongpl{#1}}%
   {\glsentrylongpl{#1}}%
  }
}
{
  \newcommand*{\GLSfmtlongpl}[1]{%
   \GLSxtrtitlelongpl{#1}}
}
\newcommand*{\glspdffmtfull}[1]{\glsentrylong{#1} (\glsentryshort{#1})}%
\newcommand*{\glspdffmtfullpl}[1]{\glsentrylongpl{#1} (\glsentryshortpl{#1})}%
\ifdef\texorpdfstring
{
  \newcommand*{\glsfmtfull}[1]{%
   \texorpdfstring
   {\glsxtrtitlefull{#1}}%
```

```
        {\glspdffmtfull{#1}}%
      }
    }
    {
      \newcommand*{\glsfmtfull}[1]{%
       \glsxtrtitlefull{#1}}
    }
    \ifdef\texorpdfstring
    {
      \newcommand*{\Glsfmtfull}[1]{%
       \texorpdfstring
       {\Glsxtrtitlefull{#1}}%
       {\glspdffmtfull{#1}{}}%
      }
    }
    {
      \newcommand*{\Glsfmtfull}[1]{%
       \Glsxtrtitlefull{#1}}
    }
    \ifdef\texorpdfstring
    {
      \newcommand*{\GLSfmtfull}[1]{%
       \texorpdfstring
       {\GLSxtrtitlefull{#1}}%
       {\glspdffmtfull{#1}}%
      }
    }
    {
      \newcommand*{\GLSfmtfull}[1]{%
       \GLSxtrtitlefull{#1}}
    }
    \ifdef\texorpdfstring
    {
      \newcommand*{\glsfmtfullpl}[1]{%
       \texorpdfstring
       {\glsxtrtitlefullpl{#1}}%
       {\glspdffmtfullpl{#1}}%
      }
    }
    {
      \newcommand*{\glsfmtfullpl}[1]{%
       \glsxtrtitlefullpl{#1}}
    }
    \ifdef\texorpdfstring
    {
      \newcommand*{\Glsfmtfullpl}[1]{%
       \texorpdfstring
       {\Glsxtrtitlefullpl{#1}}%
       {\glspdffmtfullpl{#1}{}}%
      }
```

```
}
{
  \newcommand*{\Glsfmtfullpl}[1]{%
    \Glsxtrtitlefullpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\GLSfmtfullpl}[1]{%
    \texorpdfstring
    {\GLSxtrtitlefullpl{#1}}%
    {\glspdffmtfullpl{#1}{}}%
  }
}
{
  \newcommand*{\GLSfmtfullpl}[1]{%
    \GLSxtrtitlefullpl{#1}}
}
\newcommand*{\multiglossaryentrysetup}[1]{\setkeys{glsxtrcombined}{#1}}
\newcommand*{\@gls@combined@indexmain}{1}
\define@choicekey{glsxtrcombined}{indexmain}%
 [\@gls@combined@indexmain@val\@gls@combined@indexmain]
 {false,true,first}[true]{}
\newcommand*{\@gls@combined@indexothers}{2}
\define@choicekey{glsxtrcombined}{indexothers}%
 [\@gls@combined@indexothers@val\@gls@combined@indexothers]
 {false,true,first}[true]{}
\newcommand*{\@gls@combined@hyper}{3}
\define@choicekey{glsxtrcombined}{hyper}%
 [\@gls@combined@hyper@val\@gls@combined@hyper]
 {none,allmain,mainonly,individual,otheronly,notmainfirst,nototherfirst,notfirst}{}
\newcommand*{\@gls@combined@encapmain}{glsnumberformat}
\define@key{glsxtrcombined}{encapmain}{%
 \renewcommand*{\@gls@combined@encapmain}{#1}%
}
\newcommand*{\@gls@combined@encapothers}{glsnumberformat}
\define@key{glsxtrcombined}{encapothers}{%
 \renewcommand*{\@gls@combined@encapothers}{#1}%
}
\newcommand*{\@gls@combined@textformat}{@firstofone}
\define@key{glsxtrcombined}{textformat}{%
 \renewcommand*{\@gls@combined@textformat}{#1}%
}
\newcommand*{\@gls@combined@category}{}
\define@key{glsxtrcombined}{category}{%
 \renewcommand*{\@gls@combined@category}{#1}%
}
\define@key{glsxtrcombinedpreopts}{category}{%
 \renewcommand*{\@gls@combined@category}{#1}%
}
\newcommand*{\@gls@combined@mglsopts}{}
```

```
\define@key{glsxtrcombined}{mglsopts}{%
 \renewcommand*{\@gls@combined@mglsopts}{#1}%
}
\define@key{glsxtrcombinedpreopts}{mglsopts}{%
 \@gls@combined@mglsopts@do
 {%
   \renewcommand*{\@gls@combined@mglsopts}{#1}%
 }%
}
\newcommand*{\@gls@combined@mglsopts@do}[1]{#1}
\newcommand*{\mgls@disable@mglsopts}{%
 \let\@gls@combined@mglsopts@do\@gls@combined@mglsopts@do@not
}
\newcommand*{\mgls@enable@mglsopts}{%
 \let\@gls@combined@mglsopts@do\@firstofone
}
\newcommand*{\@gls@combined@mglsopts@do@not}[1]{%
  \PackageError{glossaries-extra}{'mglsopts' key not permitted inside
    'setup' value}{}%
}
\newcommand*{\@gls@combined@firstprefix}{}
\define@key{glsxtrcombined}{firstprefix}{%
 \renewcommand*{\@gls@combined@firstprefix}{#1}%
}
\newcommand*{\@gls@combined@usedprefix}{}
\define@key{glsxtrcombined}{usedprefix}{%
 \renewcommand*{\@gls@combined@usedprefix}{#1}%
}
\newcommand*{\@gls@combined@firstsuffix}{}
\define@key{glsxtrcombined}{firstsuffix}{%
 \renewcommand*{\@gls@combined@firstsuffix}{#1}%
}
\newcommand*{\@gls@combined@usedsuffix}{}
\define@key{glsxtrcombined}{usedsuffix}{%
 \renewcommand*{\@gls@combined@usedsuffix}{#1}%
}
\define@boolkey{glsxtrcombined}{firstskipmain}[true]{}
\KV@glsxtrcombined@firstskipmainfalse
\define@boolkey{glsxtrcombined}{firstskipothers}[true]{}
\KV@glsxtrcombined@firstskipothersfalse
\define@boolkey{glsxtrcombined}{usedskipmain}[true]{}
\KV@glsxtrcombined@usedskipmainfalse
\define@boolkey{glsxtrcombined}{usedskipothers}[true]{}
\KV@glsxtrcombined@usedskipothersfalse
\newcommand*{\@gls@combined@postlinks@nr}{0}
\define@choicekey{glsxtrcombined}{postlinks}%
 [\@gls@combined@postlinks@val\@gls@combined@postlinks@nr]
 {none,all,notlast,mainnotlast,mainonly,othernotlast,otheronly}{}
\newcommand*{\@gls@combined@mpostlink@nr}{1}
\define@choicekey{glsxtrcombined}{mpostlink}%
```

```
 [\@gls@combined@mpostlink@val\@gls@combined@mpostlink@nr]
 {false,true,firstonly,usedonly}[true]{}
\newcommand*{\@gls@combined@mpostlinkelement@nr}{0}
\define@choicekey{glsxtrcombined}{mpostlinkelement}%
 [\@gls@combined@mpostlinkelement@val\@gls@combined@mpostlinkelement@nr]
 {last,main,custom}{}
\newcommand*{\glsxtrifmulti}[3]{\ifcsdef{@gls@combined@#1@main}{#2}{#3}}
\newcommand*{\glsxtrmultimain}[1]{\csuse{@gls@combined@#1@main}}
\newcommand*{\glsxtrmultilist}[1]{\csuse{@gls@combined@#1@list}}
\newcommand*{\glsxtrmultitotalelements}[1]{\csuse{@gls@combined@#1@total}}
\newcommand*{\glsxtrmultimainindex}[1]{\csuse{@gls@combined@#1@mainindex}}
\newcommand*{\glsxtrmultilastotherindex}[1]{\csuse{@gls@combined@#1@lastotherindex}}
\newif\ifmultiglossaryentryglobal
\multiglossaryentryglobalfalse
\newcount\mglselementindex
\newrobustcmd{\multiglossaryentry}[1][]{%
 \def\@gls@combined@current@opts{#1}%
 \ifnum\@glsxtr@docdefval=1\relax
  \let\@multi@glossentry@donext\@defmultiglossaryentry
 \else
  \let\@multi@glossentry@donext\@multiglossaryentry
 \fi
 \@multi@glossentry@donext
}
\newcommand*{\@multiglossaryentry}[1]{%
  \def\@gls@combined@current@label{#1}%
  \@multi@glossaryentry
}
\newcommand*{\@multi@glossaryentry}[2][]{%
  \ifcsdef{@gls@combined@\@gls@combined@current@label @main}%
  {\PackageError{glossaries-extra}%
    {Multi-entry label '\@gls@combined@current@label' already defined}%
    {}%
  }%
  {%
    \@multi@glossary@entry{#1}{#2}%
  }%
}
\newcommand*{\@defmultiglossaryentry}[1]{%
  \def\@gls@combined@current@label{#1}%
  \@def@multi@glossaryentry
}
\newcommand*{\@def@multi@glossaryentry}[2][]{%
  \let\@def@multi@glossaryentry@do\@multi@glossary@entry
  \ifundef\@glsxtr@docdefs@multilist
  {%
    \gdef\@glsxtr@docdefs@multilist{}%
    \listxadd
     {\@glsxtr@docdefs@multilist}{\expandonce\@gls@combined@current@label}%
  }%
```

1091

```
  {%
    \xifinlist{\@gls@combined@current@label}{\@glsxtr@docdefs@multilist}%
    {%
      \PackageError{glossaries-extra}%
      {Multi-entry label '\@gls@combined@current@label' already defined}%
      {}%
      \let\@def@multi@glossaryentry@do\@gobbletwo
    }%
    {%
      \listxadd
       {\@glsxtr@docdefs@multilist}{\expandonce\@gls@combined@current@label}%
    }%
  }%
  \@def@multi@glossaryentry@do{#1}{#2}%
}
\newcommand*{\@multi@glossary@doifexists}{\glsdoifexists}
\newrobustcmd{\providemultiglossaryentry}[2][]{%
 \def\@gls@combined@current@opts{#1}%
 \def\@gls@combined@current@label{#2}%
 \ifcsdef{@gls@combined@\@gls@combined@current@label @main}%
 {\def\@multi@glossentry@donext{\@provide@multi@glossaryentry@noop}}%
 {%
   \ifnum\@glsxtr@docdefval=1\relax
    \def\@multi@glossentry@donext{\@def@multi@glossaryentry}%
   \else
    \def\@multi@glossentry@donext{\@multi@glossaryentry}%
   \fi
 }%
 \@multi@glossentry@donext
}
\newcommand*{\@provide@multi@glossaryentry@noop}[2][]{}
\newcommand*{\@multi@glossaryentry@list}{}
\newcommand*{\@multi@glossary@entry}[2]{%
  \protected@edef\@gls@combined@current@main{#1}%
  \protected@edef\@gls@combined@currentlist{#2}%
  \mglselementindex=0\relax
  \@for\@gls@tmp:=\@gls@combined@currentlist\do{%
    \advance\mglselementindex by 1\relax
    \@multi@glossary@doifexists{\@gls@tmp}{}%
    \let\@gls@combined@finalitem\@gls@tmp
    \ifdefvoid\@gls@combined@current@main
    {}%
    {%
      \ifx\@gls@combined@current@main\@gls@tmp
        \ifmultiglossaryentryglobal
          \global\cslet{@gls@combined@\@gls@combined@current@label @main}%
                \@gls@combined@current@main
          \csxdef{@gls@combined@\@gls@combined@current@label @mainindex}%
                {\the\mglselementindex}%
        \else
```

```
        \cslet{@gls@combined@\@gls@combined@current@label @main}%
               \@gls@combined@current@main
        \csedef{@gls@combined@\@gls@combined@current@label @mainindex}%
               {\the\mglselementindex}%
      \fi
    \else
      \ifmultiglossaryentryglobal
        \csxdef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
               {\the\mglselementindex}%
      \else
        \csedef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
               {\the\mglselementindex}%
      \fi
    \fi
  }%
}%
\ifmultiglossaryentryglobal
  \csxdef{@gls@combined@\@gls@combined@current@label @total}%
         {\the\mglselementindex}%
\else
  \csedef{@gls@combined@\@gls@combined@current@label @total}%
         {\the\mglselementindex}%
\fi
\ifnum\mglselementindex<2\relax
   \PackageError{glossaries-extra}{At least 2 labels required in
     multi-entry element list (\number\mglselementindex\space found)}{}%
\else
  \ifdefvoid\@gls@combined@current@main
  {}%
  {%
    \ifcsundef{@gls@combined@\@gls@combined@current@label @main}%
    {\PackageError{glossaries-extra}%
     {Main element '\@gls@combined@current@main' not found in list}%
     {The final element '\@gls@combined@finalitem' will be used instead}
     \let\@gls@combined@current@main\@empty
    }%
    {}%
  }%
  \ifdefvoid\@gls@combined@current@main
  {%
    \ifmultiglossaryentryglobal
      \global\cslet{@gls@combined@\@gls@combined@current@label @main}%
             \@gls@combined@finalitem
      \global\csletcs{@gls@combined@\@gls@combined@current@label @mainindex}%
        {@gls@combined@\@gls@combined@current@label @total}%
      \csxdef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
             {\the\numexpr\mglselementindex-1 }%
    \else
      \cslet{@gls@combined@\@gls@combined@current@label @main}%
             \@gls@combined@finalitem
```

```
          \csletcs{@gls@combined@\@gls@combined@current@label @mainindex}%
            {@gls@combined@\@gls@combined@current@label @total}%
          \csedef{@gls@combined@\@gls@combined@current@label @lastotherindex}%
                  {\the\numexpr\mglselementindex-1 }%
        \fi
      }%
      {}%
      \ifmultiglossaryentryglobal
        \global\cslet{@gls@combined@\@gls@combined@current@label @list}%
                \@gls@combined@currentlist
        \protected@csxdef{@gls@combined@\@gls@combined@current@label @options}%
         {\@gls@combined@current@opts}%
        \expandafter\@ifdefinable
         \csname if@gls@combined@\@gls@combined@current@label @flag\endcsname
         {\expandafter\global\expandafter
           \newif\csname if@gls@combined@\@gls@combined@current@label @flag\endcsname}%
        \expandafter\global
         \csname @gls@combined@\@gls@combined@current@label @flagfalse\endcsname
      \else
        \cslet{@gls@combined@\@gls@combined@current@label @list}%
                \@gls@combined@currentlist
        \protected@csedef{@gls@combined@\@gls@combined@current@label @options}%
         {\@gls@combined@current@opts}%
        \newboolean{@gls@combined@\@gls@combined@current@label @flag}%
        \csname @gls@combined@\@gls@combined@current@label @flagfalse\endcsname
      \fi
    \fi
    \writemultiglossentry
     {\@gls@combined@current@opts}{\@gls@combined@current@label}%
     {\csuse{@gls@combined@\@gls@combined@current@label @main}}{#2}%
    \ifmultiglossaryentryglobal
      \ifdefempty\@multi@glossaryentry@list
      {\let\@multi@glossaryentry@list\@gls@combined@current@label}%
      {%
        \eappto\@multi@glossaryentry@list{,\expandonce\@gls@combined@current@label}%
      }%
    \else
      \ifdefempty\@multi@glossaryentry@list
      {\global\let\@multi@glossaryentry@list\@gls@combined@current@label}%
      {%
        \xappto\@multi@glossaryentry@list{,\expandonce\@gls@combined@current@label}%
      }%
    \fi
}
\newcommand*{\@glsxtr@multientry}[4]{%
 \ifnum\@glsxtr@docdefval=1\relax
  \bgroup
    \def\@gls@combined@current@opts{#1}%
    \def\@gls@combined@current@label{#2}%
    \let\@multi@glossary@doifexists\@secondoftwo
```

1094

```
    \let\writemultiglossentry\@gobblefour
    \multiglossaryentryglobaltrue
    \@multi@glossary@entry{#3}{#4}%
  \egroup
 \fi
}
\newcommand*{\writemultiglossentry}[4]{%
  \protected@write\@auxout{}{\string\@glsxtr@multientry{#1}{#2}{#3}{#4}}%
}
\newcommand*{\ifmglsused}[3]{%
 \ifbool{@gls@combined@#1@flag}{#2}{#3}%
}
\newcommand*{\mglsunset}[1]{%
 \gls@ifnotmeasuring
 {%
    \glsxtrifmulti{#1}{\@mglsunset{#1}}%
    {%
      \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
      {You need to define '#1' with \string\multiglossaryentry}%
    }%
 }%
}
\newcommand*{\@mglsunset}[1]{%
 \expandafter\global\csname @gls@combined@#1@flagtrue\endcsname
}
\newcommand*{\mglsreset}[1]{%
 \gls@ifnotmeasuring
 {%
    \glsxtrifmulti{#1}{\@mglsreset{#1}}%
    {%
      \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
      {You need to define '#1' with \string\multiglossaryentry}%
    }%
 }%
}
\newcommand*{\@mglsreset}[1]{%
 \expandafter\global\csname @gls@combined@#1@flagfalse\endcsname
}
\newcommand*{\mglslocalunset}[1]{%
 \gls@ifnotmeasuring
 {%
    \glsxtrifmulti{#1}{\@mglslocalunset{#1}}%
    {%
      \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
      {You need to define '#1' with \string\multiglossaryentry}%
    }%
 }%
}
\newcommand*{\@mglslocalunset}[1]{%
 \csname @gls@combined@#1@flagtrue\endcsname
```

```
}
\newcommand*{\mglslocalreset}[1]{%
 \gls@ifnotmeasuring
 {%
    \glsxtrifmulti{#1}{\@mglslocalreset{#1}}%
    {%
      \glsxtrundefaction{Multi entry '#1' hasn't been defined}%
      {You need to define '#1' with \string\multiglossaryentry}%
    }%
 }%
}
\newcommand*{\@mglslocalreset}[1]{%
 \csname @gls@combined@#1@flagfalse\endcsname
}
\newcommand*{\mglsunsetall}{%
  \@for\@mgls@thislabel:=\@multi@glossaryentry@list\do{\mglsunset\@mgls@thislabel}%
}%
\newcommand*{\mglsresetall}{%
  \@for\@mgls@thislabel:=\@multi@glossaryentry@list\do{\mglsreset\@mgls@thislabel}%
}%
\newrobustcmd{\mglsSetMain}[2]{%
  \ifcsundef{@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label '#1' not defined}{}}%
  {%
    \protected@edef\@gls@combined@current@main{#2}%
    \letcs\@gls@combined@currentlist{@gls@combined@#1@list}%
    \mglselementindex=0\relax
    \count@=0\relax
    \@for\@gls@tmp:=\@gls@combined@currentlist\do{%
      \advance\mglselementindex by 1\relax
      \ifx\@gls@combined@current@main\@gls@tmp
        \count@=\mglselementindex\relax
        \let\@gls@combined@finalitem\@gls@tmp
        \ifmultiglossaryentryglobal
          \global\cslet{@gls@combined@#1@main}\@gls@combined@current@main
          \csxdef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \else
          \cslet{@gls@combined@#1@main}\@gls@combined@current@main
          \csedef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \fi
      \else
        \ifmultiglossaryentryglobal
          \csxdef{@gls@combined@#1@lastotherindex}{\the\mglselementindex}%
        \else
          \csedef{@gls@combined@#1@lastotherindex}{\the\mglselementindex}%
        \fi
      \fi
    }%
    \ifnum\count@=0\relax
      \PackageError{glossaries-extra}{Label '#2' is not in '#1' set
```

```
        (\@gls@combined@currentlist)}{}%
      \ifmultiglossaryentryglobal
        \global\cslet{@gls@combined@#1@main}\@gls@combined@finalitem
        \csxdef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \csxdef{@gls@combined@#1@lastotherindex}{%
          \number\numexpr\mglselementindex-1 }%
      \else
        \cslet{@gls@combined@#1@main}\@gls@combined@finalitem
        \csedef{@gls@combined@#1@mainindex}{\the\mglselementindex}%
        \csedef{@gls@combined@#1@lastotherindex}{%
          \number\numexpr\mglselementindex-1 }%
      \fi
    \fi
  }%
}
\newrobustcmd{\mglsSetOptions}[2]{%
  \ifcsundef{@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label '#1' not defined}{}}%
  {%
    \csdef{@gls@combined@#1@options}{#2}%
  }%
}
\newrobustcmd{\mglsAddOptions}[2]{%
  \ifcsundef{@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label '#1' not defined}{}}%
  {%
    \ifcsempty{@gls@combined@#1@options}%
    {\csdef{@gls@combined@#1@options}{#2}}%
    {\csappto{@gls@combined@#1@options}{,#2}}%
  }%
}
\newcommand*{\@mgls@all}{}
\define@key{mgls}{all}{\renewcommand*{\@mgls@all}{#1}}
\newcommand*{\@mgls@main}{}
\define@key{mgls}{main}{\renewcommand*{\@mgls@main}{#1}}
\newcommand*{\@mgls@others}{}
\define@key{mgls}{others}{\renewcommand*{\@mgls@others}{#1}}
\newcommand*{\@mgls@setup}{}
\define@key{mgls}{setup}{%
 \@mgls@setup@do{\renewcommand*{\@mgls@setup}{#1}}%
}
\newcommand*{\@mgls@setup@do}[1]{#1}
\newcommand*{\@mgls@setup@do@not}[1]{%
  \PackageError{glossaries-extra}{'setup' key not permitted inside
    'mglsopts' value}{}%
}
\newcommand*{\mgls@disable@setup}{%
  \let\@mgls@setup@do\@mgls@setup@do@not
}
\newcommand*{\mgls@enable@setup}{%
```

```
    \let\@mgls@setup@do\@firstofone
}
\newcommand\@mgls@unsetaction{0}
\define@choicekey{mgls}{multiunset}[\@mgls@unsetaction@val\@mgls@unsetaction]%
 {global,local,none}{}
\define@boolkey{mgls}{presetlocal}[true]{}
\KV@mgls@presetlocalfalse
\newcommand*{\@mgls@hyper}{}
\define@choicekey{mgls}{hyper}[\@mgls@hyper@val\@mgls@hyper@nr]{true,false}[true]%
{%
  \renewcommand*{\@mgls@hyper}{hyper=#1}%
  \ifnum\@mgls@hyper@nr=1\relax
   \let\@mgls@hyperlink\@secondoftwo
  \else
   \let\@mgls@hyperlink\@@mgls@hyperlink
  \fi
}
\newcommand*{\@@mgls@hyperlink}[2]{%
  \ifx\@glslink\glsdonohyperlink
   #2%
  \else
   \glsxtr@org@dohyperlink{\glolinkprefix#1}{#2}%
  \fi
}
\let\@mgls@hyperlink\@@mgls@hyperlink
\newcommand*{\mglsforelements}[3]{%
  \expandafter\@for\expandafter#2\expandafter:\expandafter
    =\csname @gls@combined@#1@list\endcsname\do{#3}%
}
\newcommand*{\mglsforotherelements}[3]{%
  \expandafter\@for\expandafter#2\expandafter:\expandafter
    =\csname @gls@combined@#1@list\endcsname\do
  {\expandafter\ifdefequal\csname @gls@combined@#1@main\endcsname{#2}{}{#3}}%
}
\newcommand*{\mglsunsetothers}[1]{%
  \mglsforotherelements{#1}{\@gls@tmp}{\glsunset{\@gls@tmp}}%
}
\newcommand*{\mglslocalunsetothers}[1]{%
  \mglsforotherelements{#1}{\@gls@tmp}{\glslocalunset{\@gls@tmp}}%
}
\newcommand*{\mglselementreset}[1]{%
 \ifKV@mgls@presetlocal
  \glslocalreset{#1}%
 \else
  \glsreset{#1}%
 \fi
}
\newcommand*{\mglselementunset}[1]{%
 \ifKV@mgls@presetlocal
  \glslocalunset{#1}%
```

```
 \else
  \glsunset{#1}%
 \fi
}
\newcommand*{\@mgls@resetall}{}
\define@choicekey{mgls}{resetall}%
[\@mgls@resetall@val\@mgls@resetall@nr]{false,true}[true]%
{%
 \ifcase\@mgls@resetall@nr\relax
    \renewcommand*{\@mgls@resetall}{}%
 \or
    \renewcommand*{\@mgls@resetall}{%
      \@for\@gls@resetlabel:=\mglscurrentlist\do{\mglselementreset\@gls@resetlabel}}%
    \renewcommand*{\@mgls@unsetall}{}%
 \fi
}
\newcommand*{\@mgls@resetmain}{}
\define@choicekey{mgls}{resetmain}
[\@mgls@resetmain@val\@mgls@resetmain@nr]{false,true}[true]%
{%
 \ifcase\@mgls@resetmain@nr\relax
    \renewcommand*{\@mgls@resetmain}{}%
 \or
    \renewcommand*{\@mgls@resetmain}{\mglselementreset\mglscurrentmainlabel}%
    \renewcommand*{\@mgls@unsetmain}{}%
 \fi
}
\newcommand*{\@mgls@resetothers}{}
\define@choicekey{mgls}{resetothers}
[\@mgls@resetothers@val\@mgls@resetothers@nr]{false,true}[true]%
{%
 \ifcase\@mgls@resetothers@nr\relax
    \renewcommand*{\@mgls@resetothers}{}%
 \or
    \renewcommand*{\@mgls@resetothers}{%
      \@for\@gls@resetlabel:=\mglscurrentlist\do{%
        \ifx\@gls@resetlabel\mglscurrentmainlabel
        \else
          \mglselementreset\@gls@resetlabel
        \fi
      }%
    }%
    \renewcommand*{\@mgls@unsetothers}{}%
 \fi
}
\newcommand*{\@mgls@unsetall}{}
\define@choicekey{mgls}{unsetall}%
[\@mgls@unsetall@val\@mgls@unsetall@nr]{false,true}[true]%
{%
 \ifcase\@mgls@unsetall@nr\relax
```

```
    \renewcommand*{\@mgls@unsetall}{}%
 \or
    \renewcommand*{\@mgls@unsetall}{%
      \@for\@gls@unsetlabel:=\mglscurrentlist\do{\mglselementunset\@gls@unsetlabel}}%
    \renewcommand*{\@mgls@resetall}{}%
 \fi
}
\newcommand*{\@mgls@unsetmain}{}
\define@choicekey{mgls}{unsetmain}
[\@mgls@unsetmain@val\@mgls@unsetmain@nr]{false,true}[true]%
{%
 \ifcase\@mgls@unsetmain@nr\relax
    \renewcommand*{\@mgls@unsetmain}{}%
 \or
    \renewcommand*{\@mgls@unsetmain}{\mglselementunset\mglscurrentmainlabel}%
    \renewcommand*{\@mgls@resetmain}{}%
 \fi
}
\newcommand*{\@mgls@unsetothers}{}
\define@choicekey{mgls}{unsetothers}
[\@mgls@unsetothers@val\@mgls@unsetothers@nr]{false,true}[true]%
{%
 \ifcase\@mgls@unsetothers@nr\relax
    \renewcommand*{\@mgls@unsetothers}{}%
 \or
    \renewcommand*{\@mgls@unsetothers}{%
      \@for\@gls@unsetlabel:=\mglscurrentlist\do{%
        \ifx\@gls@unsetlabel\mglscurrentmainlabel
        \else
          \mglselementunset\@gls@unsetlabel
        \fi
      }%
    }%
    \renewcommand*{\@mgls@resetothers}{}%
 \fi
}
\newcommand{\glsxtr@setup@docurrent}{%
 \ifx\mglscurrentlabel\mglscurrentmainlabel
    \mglsisfirstuse
    {%
      \ifKV@glsxtrcombined@firstskipmain
        \let\@mgls@do@current@element\@gobble
      \else
        \let\@mgls@do@current@element\@firstofone
      \fi
    }%
    {%
      \ifKV@glsxtrcombined@usedskipmain
        \let\@mgls@do@current@element\@gobble
      \else
```

1100

```
        \let\@mgls@do@current@element\@firstofone
      \fi
    }%
 \else
   \mglsisfirstuse
   {%
     \ifKV@glsxtrcombined@firstskipothers
       \let\@mgls@do@current@element\@gobble
     \else
       \let\@mgls@do@current@element\@firstofone
     \fi
   }%
   {%
     \ifKV@glsxtrcombined@usedskipothers
       \let\@mgls@do@current@element\@gobble
     \else
       \let\@mgls@do@current@element\@firstofone
     \fi
   }%
 \fi
}
\newcommand*{\glsxtr@mgls@checklastelement}[2]{%
  \ifbool{KV@glsxtrcombined@#1skipmain}%
  {%
    \ifbool{KV@glsxtrcombined@#1skipothers}%
    {%
    }%
    {%
      \ifnum\mglselementindex=\glsxtrmultilastotherindex{#2}\relax
        \let\mglsiflast\@firstoftwo
      \else
        \let\mglsiflast\@secondoftwo
      \fi
    }%
  }%
  {%
    \ifbool{KV@glsxtrcombined@#1skipothers}%
    {%
      \ifnum\mglselementindex=\glsxtrmultimainindex{#2}\relax
        \let\mglsiflast\@firstoftwo
      \else
        \let\mglsiflast\@secondoftwo
      \fi
    }%
    {%
      \let\mglsiflast\@secondoftwo
    }%
  }%
}
\newcommand{\glsxtrmglsWarnAllSkipped}[3]{%
```

```
    \GlossariesExtraWarning{#1}%
    #3{#2}%
}
\newcommand*{\glsxtr@mgls@applyopts}[1]{%
  \edef\@mgls@dooptions{\noexpand\setkeys*{mgls}{\expandonce#1}}%
  \@mgls@dooptions
  \ifdefvoid\XKV@rm{}{\eappto\@mgls@all{,\expandonce\XKV@rm}}%
  \ifdefvoid\@mgls@setup
  {}%
  {%
    \edef\@mgls@dooptions{%
      \noexpand\setkeys*{glsxtrcombinedpreopts}{\expandonce\@mgls@setup}}%
    \mgls@disable@mglsopts
    \@mgls@dooptions
    \mgls@enable@mglsopts
    \ifx\@mgls@setupoptions\@empty
      \let\@mgls@setupoptions\XKV@rm
    \else
      \eappto\@mgls@setupoptions{,\expandonce\XKV@rm}%
    \fi
  }%
  \@mgls@resetall
  \@mgls@unsetall
  \@mgls@resetmain
  \@mgls@unsetmain
  \@mgls@resetothers
  \@mgls@unsetothers
  \let\@mgls@resetall\@empty
  \let\@mgls@resetmain\@empty
  \let\@mgls@resetothers\@empty
  \let\@mgls@unsetall\@empty
  \let\@mgls@unsetmain\@empty
  \let\@mgls@unsetothers\@empty
  \ifmglsused\mglscurrentmultilabel
  {\let\mglsisfirstuse\@secondoftwo}%
  {\let\mglsisfirstuse\@firstoftwo}%
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\providecommand{\@thirdofthree}[3]{#3}
\newcommand*{\glsxtr@mgls@inner}[7]{%
 \let\mglslastmainlabel\@empty
 \let\mglsiflastmainwasfirstuse\@firstoftwo
 \let\mglsiflastmainwasplural\@secondoftwo
 \let\mglsiflastmaincapscase\@firstofthree
 \let\mglsiflastmainskipped\@firstoftwo
 \bgroup
 \ifcsundef{@gls@combined@#2@main}%
 {%
   \glsxtrundefaction{Multi entry '#2' hasn't been defined}%
```

```
    {You need to define '#2' with \string\multiglossaryentry}%
    \gdef\@mgls@post@hookdefs{%
      \protected@edef\mglslastmultilabel{#2}%
      \let\mglswasfirstuse\@firstoftwo
      \let\mglslastcategory\@empty
      \let\mglsiflastelementskipped\@firstoftwo
      \let\mglsiflastelementwasfirstuse\@firstoftwo
      \let\mglsiflastelementwasplural\@secondoftwo
      \let\mglsiflastelementcapscase\@firstofthree
      \let\mglslastelementlabel\@empty
      \let\mgls@do@postlinkhook\relax
    }%
}%
{%
    \protected@edef\mglscurrentmultilabel{#2}%
    \letcs\mglscurrentmainlabel{@gls@combined@#2@main}%
    \letcs\mglscurrentlist{@gls@combined@#2@list}%
    \letcs\mglscurrentoptions{@gls@combined@#2@options}%
    \ifmglsused\mglscurrentmultilabel
    {\let\mglsisfirstuse\@secondoftwo}%
    {\let\mglsisfirstuse\@firstoftwo}%
    \edef\@mgls@dooptions{%
        \noexpand\setkeys*{glsxtrcombinedpreopts}{\expandonce\mglscurrentoptions}}%
    \@mgls@dooptions
    \let\@mgls@setupoptions\XKV@rm
    \mgls@disable@setup
    \ifdefvoid\@gls@combined@mglsopts
    {}%
    {\glsxtr@mgls@applyopts\@gls@combined@mglsopts}%
    \mgls@enable@setup
    \ifstrempty{#1}{}{\def\@mgls@options{#1}\glsxtr@mgls@applyopts\@mgls@options}%
    \ifx\@gls@combined@category\empty
    \else
      \glshascategoryattribute{\@gls@combined@category}{multioptions}%
      {%
        \letcs\@mgls@attroptions{@glsxtr@categoryattr@@\@gls@combined@category
            @multioptions}%
        \let\@gls@combined@mglsopts\@empty
        \edef\@mgls@dooptions{%
          \noexpand\setkeys*{glsxtrcombinedpreopts}{\expandonce\@mgls@attroptions}}%
        \@mgls@dooptions
        \eappto\@mgls@setupoptions{,\expandonce\XKV@rm}%
        \ifx\@gls@combined@mglsopts\@empty
        \else
          \let\@mgls@setup\@empty
          \mgls@disable@setup
          \glsxtr@mgls@applyopts\@gls@combined@mglsopts
          \mgls@enable@setup
        \fi
      }%
```

```
    {}%
\fi
\edef\@mgls@dooptions{%
   \noexpand\setkeys{glsxtrcombined}{\expandonce\@mgls@setupoptions}}%
\@mgls@dooptions
\let\mglscurrentcategory\@gls@combined@category
\ifnum\@gls@combined@hyper=1\relax
  \def\@mgls@combinedlink{\@mgls@hyperlink{\mglscurrentmainlabel}}%
\else
  \def\@mgls@combinedlink{\@firstofone}%
\fi
\def\@gls@combined@encapsulator##1{%
   \@mgls@combinedlink{\csuse{\@gls@combined@textformat}{##1}}}%
\let\@mgls@do@current@element\@firstofone
\mglsisfirstuse
{%
  \ifKV@glsxtrcombined@firstskipmain
    \ifKV@glsxtrcombined@firstskipothers
      \let\@gls@org@combined@encapsulator\@gls@combined@encapsulator
      \def\@gls@combined@encapsulator##1{%
        \glsxtrmglsWarnAllSkipped{All elements skipped for
          first use of multi-entry '#2'}{#3}%
          {\@gls@org@combined@encapsulator}%
      }%
      \let\@mgls@do@current@element\@gobble
    \fi
  \fi
}%
{%
  \ifKV@glsxtrcombined@usedskipmain
    \ifKV@glsxtrcombined@usedskipothers
      \let\@gls@org@combined@encapsulator\@gls@combined@encapsulator
      \def\@gls@combined@encapsulator##1{%
        \glsxtrmglsWarnAllSkipped{All elements skipped for
          subsequent use of multi-entry '#2'}{#3}%
          {\@gls@org@combined@encapsulator}%
      }%
      \let\@mgls@do@current@element\@gobble
    \fi
  \fi
}%
\mglsisfirstuse
{%
  \let\mglscurrentprefix\@gls@combined@firstprefix
  \let\mglscurrentsuffix\@gls@combined@firstsuffix
}%
{%
  \let\mglscurrentprefix\@gls@combined@usedprefix
  \let\mglscurrentsuffix\@gls@combined@usedsuffix
}%
```

```
\xdef\@mgls@post@hookdefs{%
 \noexpand\def\noexpand\mglslastmultilabel{\expandonce\mglscurrentmultilabel}%
 \noexpand\def\noexpand\mglslastcategory{\mglscurrentcategory}%
}%
\ifx\@mgls@do@current@element\@gobble
  \gappto\@mgls@post@hookdefs{%
    \let\mglsiflastelementskipped\@firstoftwo
    \let\mglslastelementlabel\@empty
    \let\mglsiflastelementwasfirstuse\@firstoftwo
    \let\mglsiflastelementwasplural\@secondoftwo
    \let\mglsiflastelementcapscase\@firstofthree
  }%
\fi
\mglsisfirstuse
{%
  \gappto\@mgls@post@hookdefs{\let\mglswasfirstuse\@firstoftwo}%
  \ifcase\@gls@combined@mpostlink@nr\relax
    \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
  \or
    \ifcase\@gls@combined@mpostlinkelement@nr\relax
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
    \fi
  \or
    \ifcase\@gls@combined@mpostlinkelement@nr\relax
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
    \fi
  \or
    \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
  \fi
}%
{%
  \gappto\@mgls@post@hookdefs{\let\mglswasfirstuse\@secondoftwo}%
  \ifcase\@gls@combined@mpostlink@nr\relax
    \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
  \or
    \ifcase\@gls@combined@mpostlinkelement@nr\relax
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
    \fi
```

```
  \or
    \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\relax}%
  \or
    \ifcase\@gls@combined@mpostlinkelement@nr\relax
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastelementpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglslastmainpostlinkhook}%
    \or
      \gappto\@mgls@post@hookdefs{\let\mgls@do@postlinkhook\mglscustompostlinkhook}%
    \fi
  \fi
}%
\let\mgls@org@postlinkhook\glspostlinkhook
\mglsprefix
\let\mglslastelementlabel\@empty
\@gls@combined@encapsulator
{%
 \def\@mgls@previouslabel{}%
 \mglselementindex=0\relax
 \@for\mglscurrentlabel:=\mglscurrentlist\do{%
  \advance\mglselementindex by 1\relax
  \glsxtr@setup@docurrent
  \ifx\@xfor@nextelement\@nnil
    \let\mglsiflast\@firstoftwo
  \else
    \let\mglsiflast\@secondoftwo
    \mglsisfirstuse
    {%
     \glsxtr@mgls@checklastelement{first}{#2}%
    }%
    {%
     \glsxtr@mgls@checklastelement{used}{#2}%
    }%
  \fi
  \ifcase\@gls@combined@postlinks@nr\relax
    \let\glspostlinkhook\relax
  \or
    \let\glspostlinkhook\mgls@org@postlinkhook
  \or
    \mglsiflast
    {%
     \let\glspostlinkhook\relax
    }%
    {%
     \let\glspostlinkhook\mgls@org@postlinkhook
    }%
  \or
    \ifx\mglscurrentlabel\mglscurrentmainlabel
      \mglsiflast
      {%
```

```
        \let\glspostlinkhook\relax
      }%
      {%
        \let\glspostlinkhook\mgls@org@postlinkhook
      }%
    \else
      \let\glspostlinkhook\relax
    \fi
\or
  \ifx\mglscurrentlabel\mglscurrentmainlabel
    \let\glspostlinkhook\mgls@org@postlinkhook
  \else
    \let\glspostlinkhook\relax
  \fi
\or
  \ifx\mglscurrentlabel\mglscurrentmainlabel
    \let\glspostlinkhook\relax
  \else
    \mglsiflast
    {%
      \let\glspostlinkhook\relax
    }%
    {%
      \let\glspostlinkhook\mgls@org@postlinkhook
    }%
  \fi
\or
  \ifx\mglscurrentlabel\mglscurrentmainlabel
    \let\glspostlinkhook\relax
  \else
    \let\glspostlinkhook\mgls@org@postlinkhook
  \fi
\fi
\mglsiflast
{%
  \xappto\@mgls@post@hookdefs{%
    \noexpand\def\noexpand\mglslastelementlabel
      {\expandonce\mglscurrentlabel}}%
}%
{}%
\@mgls@do@current@element
{%
  \mglselementprehook
  \GlsXtrIfUnusedOrUndefined{\mglscurrentlabel}%
  {\let\@mgls@current@iffirstuse\@firstoftwo}%
  {\let\@mgls@current@iffirstuse\@secondoftwo}%
  \ifx\mglscurrentlabel\mglscurrentmainlabel
    \edef\@mgls@current@options{format=\@gls@combined@encapmain}%
    \ifcase\@gls@combined@indexmain
      \appto\@mgls@current@options{,noindex}%
```

```
\or
 \appto\@mgls@current@options{,noindex=false}%
\or
 \@mgls@current@iffirstuse
 {\appto\@mgls@current@options{,noindex=false}}%
 {\appto\@mgls@current@options{,noindex}}%
\fi
\ifcase\@gls@combined@hyper\relax
  \appto\@mgls@current@options{,hyper=false}% none
\or
  \appto\@mgls@current@options{,hyper=false}% allmain
\or
  \eappto\@mgls@current@options{,\@mgls@hyper}% mainonly
\or
  \eappto\@mgls@current@options{,\@mgls@hyper}% individual
\or
  \appto\@mgls@current@options{,hyper=false}% otheronly
\or
 \mglsisfirstuse
 {%
  \appto\@mgls@current@options{,hyper=false}% notmainfirst
 }%
 {%
  \eappto\@mgls@current@options{,\@mgls@hyper}% notmainfirst
 }%
\or
  \eappto\@mgls@current@options{,\@mgls@hyper}% nototherfirst
\or
 \mglsisfirstuse
 {%
  \appto\@mgls@current@options{,hyper=false}% notfirst
 }%
 {%
  \eappto\@mgls@current@options{,\@mgls@hyper}% notfirst
 }%
 \fi
 \eappto\@mgls@current@options{,\@mgls@all,\@mgls@main}%
\else
 \edef\@mgls@current@options{format=\@gls@combined@encapothers}%
 \ifcase\@gls@combined@indexothers\relax
 \appto\@mgls@current@options{,noindex}%
\or
 \appto\@mgls@current@options{,noindex=false}%
\or
 \@mgls@current@iffirstuse
 {\appto\@mgls@current@options{,noindex=false}}%
 {\appto\@mgls@current@options{,noindex}}%
\fi
 \ifcase\@gls@combined@hyper\relax
  \appto\@mgls@current@options{,hyper=false}% none
```

```
  \or
    \appto\@mgls@current@options{,hyper=false}% allmain
  \or
    \appto\@mgls@current@options{,hyper=false}% mainonly
  \or
    \eappto\@mgls@current@options{,\@mgls@hyper}% individual
  \or
    \eappto\@mgls@current@options{,\@mgls@hyper}% otheronly
  \or
    \eappto\@mgls@current@options{,\@mgls@hyper}% notmainfirst
  \or
   \mglsisfirstuse
   {%
    \appto\@mgls@current@options{,hyper=false}% nototherfirst
   }%
   {%
    \eappto\@mgls@current@options{,\@mgls@hyper}% nototherfirst
   }%
  \or
   \mglsisfirstuse
   {%
    \appto\@mgls@current@options{,hyper=false}% notfirst
   }%
   {%
    \eappto\@mgls@current@options{,\@mgls@hyper}% notfirst
   }%
  \fi
  \eappto\@mgls@current@options{,\@mgls@all,\@mgls@others}%
\fi
\ifx\@mgls@previouslabel\empty
 \ifx\mglscurrentlabel\mglscurrentmainlabel
   \let\@mgls@cs#6\relax
 \else
   \let\@mgls@cs#4\relax
 \fi
\else
  \@mgls@previous@iffirstuse
  {%
    \@mgls@current@iffirstuse
    {\glscombinedfirstsepfirst{\@mgls@previouslabel}{\mglscurrentlabel}}%
    {\glscombinedfirstsep{\@mgls@previouslabel}{\mglscurrentlabel}}%
  }%
  {%
    \@mgls@current@iffirstuse
    {\glscombinedsepfirst{\@mgls@previouslabel}{\mglscurrentlabel}}%
    {\glscombinedsep{\@mgls@previouslabel}{\mglscurrentlabel}}%
  }%
 \ifx\mglscurrentlabel\mglscurrentmainlabel
   \let\@mgls@cs#7\relax
 \else
```

```
      \let\@mgls@cs#5\relax
    \fi
  \fi
  \mglsiflast
   {\expandafter\@mgls@cs\expandafter{\@mgls@current@options}{\mglscurrentlabel}[#3]}%
   {\expandafter\@mgls@cs\expandafter{\@mgls@current@options}{\mglscurrentlabel}[]}%
  \ifx\mglscurrentlabel\mglscurrentmainlabel
   \xappto\@mgls@post@hookdefs{%
     \noexpand\def\noexpand\mglslastmainlabel
       {\expandonce\mglscurrentmainlabel}%
   }%
   \glsxtrifwasfirstuse
   {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasfirstuse\@firstoftwo}%
   }%
   {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasfirstuse\@secondoftwo}%
   }%
   \glsifplural
   {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasplural\@firstoftwo}%
   }%
   {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastmainwasplural\@secondoftwo}%
   }%
   \glscapscase
   {%
     \gappto\@mgls@post@hookdefs{%
       \let\mglsiflastmaincapscase\@firstofthree
     }%
   }%
   {%
     \gappto\@mgls@post@hookdefs{%
       \let\mglsiflastmaincapscase\@secondofthree
     }%
   }%
   {%
     \gappto\@mgls@post@hookdefs{%
       \let\mglsiflastmaincapscase\@thirdofthree
     }%
   }%
  \fi
  \let\@mgls@previouslabel\mglscurrentlabel
  \let\@mgls@previous@iffirstuse\@mgls@current@iffirstuse
 }%
  \mglselementposthook
}%
\ifx\mglslastmainlabel\@empty
 \gappto\@mgls@post@hookdefs{\let\mglsiflastmainskipped\@firstoftwo}%
\else
```

```
    \gappto\@mgls@post@hookdefs{\let\mglsiflastmainskipped\@secondoftwo}%
  \fi
  \ifx\@mgls@do@current@element\@gobble
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementskipped\@firstoftwo}%
  \else
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementskipped\@secondoftwo}%
  \fi
  \glsxtrifwasfirstuse
  {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@firstoftwo}%
  }%
  {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@secondoftwo}%
  }%
  \glsifplural
  {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasplural\@firstoftwo}%
  }%
  {%
    \gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasplural\@secondoftwo}%
  }%
  \glscapscase
  {%
     \gappto\@mgls@post@hookdefs{%
       \let\mglsiflastelementcapscase\@firstofthree
     }%
  }%
  {%
     \gappto\@mgls@post@hookdefs{%
       \let\mglsiflastelementcapscase\@secondofthree
     }%
  }%
  {%
     \gappto\@mgls@post@hookdefs{%
       \let\mglsiflastelementcapscase\@thirdofthree
     }%
   }%
  }%
  \@mgls@post@hookdefs
  \mglssuffix
  \ifcase\@mgls@unsetaction\relax
    \xappto\@mgls@post@hookdefs{%
       \noexpand\mglsunset{\expandonce\mglscurrentmultilabel}}%
  \or
    \xappto\@mgls@post@hookdefs{%
       \noexpand\mglslocalunset{\expandonce\mglscurrentmultilabel}}%
  \fi
}%
\glsxtrmglswrite{#2}%
\egroup
```

```
 \@mgls@post@hookdefs
 \mgls@do@postlinkhook
}
\newcommand*{\mglscustompostlinkhook}{}
\newcommand*{\mglslastelementpostlinkhook}{%
 \let\glsxtrifwasfirstuse\mglsiflastelementwasfirstuse
 \let\glsifplural\mglsiflastelementwasplural
 \let\glscapscase\mglsiflastelementcapscase
 \let\glslabel\mglslastelementlabel
 \glspostlinkhook
}
\newcommand*{\mglslastmainpostlinkhook}{%
 \let\glsxtrifwasfirstuse\mglsiflastmainwasfirstuse
 \let\glsifplural\mglsiflastmainwasplural
 \let\glscapscase\mglsiflastmaincapscase
 \let\glslabel\mglslastmainlabel
 \glspostlinkhook
}
\newcommand*{\mglsdefcategoryprefix}[2]{%
 \csdef{mglsprefix@#1}{#2}%
}
\newcommand*{\mglshascategoryprefix}[3]{%
 \ifcsdef{mglsprefix@#1}{#2}{#3}%
}
\newcommand*{\mglsusecategoryprefix}[1]{%
 \csuse{mglsprefix@#1}%
}
\newcommand*{\mglsprefix}{%
 \ifdefempty\mglscurrentcategory
 {\mglscurrentprefix}%
 {%
   \mglshascategoryprefix{\mglscurrentcategory}%
   {\mglsusecategoryprefix{\mglscurrentcategory}}%
   {\mglscurrentprefix}%
 }%
}
\newcommand*{\mglsdefcategorysuffix}[2]{%
 \csdef{mglssuffix@#1}{#2}%
}
\newcommand*{\mglshascategorysuffix}[3]{%
 \ifcsdef{mglssuffix@#1}{#2}{#3}%
}
\newcommand*{\mglsusecategorysuffix}[1]{%
 \csuse{mglssuffix@#1}%
}
\newcommand*{\mglssuffix}{%
 \ifdefempty\mglscurrentcategory
 {\ifdefempty{\mglscurrentsuffix}{}{\space(\mglscurrentsuffix)}}%
 {%
   \mglshascategorysuffix{\mglscurrentcategory}%
```

```
    {\mglsusecategorysuffix{\mglscurrentcategory}}%
    {\ifdefempty{\mglscurrentsuffix}{}{\space(\mglscurrentsuffix)}}}%
 }%
}
\newcommand*{\mglselementprehook}{}
\newcommand*{\mglselementposthook}{}
\newcommand*{\glscombinedsep}[2]{%
  \glshasattribute{#1}{combinedsep}%
  {\glsgetattribute{#1}{combinedsep}}%
  { }%
}
\newcommand*{\glscombinedfirstsepfirst}[2]{%
  \glshasattribute{#1}{combinedfirstsepfirst}%
  {\glsgetattribute{#1}{combinedfirstsepfirst}}%
  {\glscombinedsep{#1}{#2}}%
}
\newcommand*{\glscombinedfirstsep}[2]{%
  \glshasattribute{#1}{combinedfirstsep}%
  {\glsgetattribute{#1}{combinedfirstsep}}%
  {\glscombinedsep{#1}{#2}}%
}
\newcommand*{\glscombinedsepfirst}[2]{%
  \glshasattribute{#1}{combinedsepfirst}%
  {\glsgetattribute{#1}{combinedsepfirst}}%
  {\glscombinedsep{#1}{#2}}%
}
\newcommand*{\glssetcombinedsepabbrvnbsp}{%
 \renewcommand*{\glscombinedsep}[2]{%
  \glshasattribute{##1}{combinedsep}%
  {\glsgetattribute{##1}{combinedsep}}%
  {\ifhasshort{##1}{~}{ }}%
 }%
 \renewcommand*{\glscombinedsepfirst}[2]{%
  \glshasattribute{##1}{combinedsepfirst}%
  {\glsgetattribute{##1}{combinedsepfirst}}%
  {\ifhasshort{##1}{~}{ }}%
 }%
 \renewcommand*{\glscombinedfirstsep}[2]{%
  \glshasattribute{##1}{combinedfirstsep}%
  {\glsgetattribute{##1}{combinedfirstsep}}%
  { }%
 }%
 \renewcommand*{\glscombinedfirstsepfirst}[2]{%
  \glshasattribute{##1}{combinedfirstsepfirst}%
  {\glsgetattribute{##1}{combinedfirstsepfirst}}%
  { }%
 }%
}
\newcommand*{\glssetcombinedsepabbrvnone}{%
 \renewcommand*{\glscombinedsep}[2]{%
```

```
   \glshasattribute{##1}{combinedsep}%
   {\glsgetattribute{##1}{combinedsep}}%
   {\ifhasshort{##1}{}{\ifhasshort{##2}{}{ }}}%
 }%
 \renewcommand*{\glscombinedsepfirst}[2]{%
   \glshasattribute{##1}{combinedsepfirst}%
   {\glsgetattribute{##1}{combinedsepfirst}}%
   {\ifhasshort{##1}{}{ }}%
 }%
 \renewcommand*{\glscombinedfirstsep}[2]{%
  \glshasattribute{##1}{combinedfirstsep}%
  {\glsgetattribute{##1}{combinedfirstsep}}%
  {\ifhasshort{##2}{}{ }}%
 }%
 \renewcommand*{\glscombinedfirstsepfirst}[2]{%
  \glshasattribute{##1}{combinedfirstsepfirst}%
  {\glsgetattribute{##1}{combinedfirstsepfirst}}%
  { }%
 }%
}
\newcommand*{\glssetcombinedsepnarrow}[2]{%
 \renewcommand*{\glscombinedsep}[2]{%
 \glshasattribute{##1}{combinedsep}%
 {\glsgetattribute{##1}{combinedsep}}%
 {%
  \ifhasshort{##1}%
  {\settowidth{\dimen@}{\glsentryshort{##1}}}%
  {\settowidth{\dimen@}{\glsentrytext{##1}}}%
  \ifdim\dimen@<#1\relax
   #2%
  \else
    \ifhasshort{##2}%
    {\settowidth{\dimen@}{\glsentryshort{##2}}}%
    {\settowidth{\dimen@}{\glsentrytext{##2}}}%
    \ifdim\dimen@<#1\relax
     #2%
    \else
     \space
    \fi
  \fi
 }%
}%
 \renewcommand*{\glscombinedsepfirst}[2]{%
 \glshasattribute{##1}{combinedsepfirst}%
 {\glsgetattribute{##1}{combinedsepfirst}}%
 {%
  \ifhasshort{##1}%
  {\settowidth{\dimen@}{\glsentryshort{##1}}}%
  {\settowidth{\dimen@}{\glsentrytext{##1}}}%
  \ifdim\dimen@<#1\relax
```

```
   #2%
  \else
    \ifhaslong{##2}%
    {\settowidth{\dimen@}{\glsentrylong{##2}}}%
    {\settowidth{\dimen@}{\glsentryfirst{##2}}}%
    \ifdim\dimen@<#1\relax
     #2%
    \else
     \space
    \fi
  \fi
 }%
}%
\renewcommand*{\glscombinedfirstsep}[2]{%
 \glshasattribute{##1}{combinedfirstsep}%
 {\glsgetattribute{##1}{combinedfirstsep}}%
 {%
    \ifhaslong{##1}%
    {\settowidth{\dimen@}{\glsentrylong{##1}}}%
    {\settowidth{\dimen@}{\glsentryfirst{##1}}}%
    \ifdim\dimen@<#1\relax
     #2%
    \else
      \ifhasshort{##2}%
      {\settowidth{\dimen@}{\glsentryshort{##2}}}%
      {\settowidth{\dimen@}{\glsentrytext{##2}}}%
      \ifdim\dimen@<#1\relax
       #2%
      \else
       \space
      \fi
    \fi
 }%
}%
\renewcommand*{\glscombinedfirstsepfirst}[2]{%
 \glshasattribute{##1}{combinedfirstsepfirst}%
 {\glsgetattribute{##1}{combinedfirstsepfirst}}%
 {%
    \ifhaslong{##1}%
    {\settowidth{\dimen@}{\glsentrylong{##1}}}%
    {\settowidth{\dimen@}{\glsentryfirst{##1}}}%
    \ifdim\dimen@<#1\relax
     #2%
    \else
      \ifhaslong{##2}%
      {\settowidth{\dimen@}{\glsentrylong{##2}}}%
      {\settowidth{\dimen@}{\glsentryfirst{##2}}}%
      \ifdim\dimen@<#1\relax
       #2%
      \else
```

```
        \space
      \fi
    \fi
  }%
 }%
}
\newcommand{\glsxtrmglswrite}[1]{%
 \ifx\@glsxtr@record@setting\@glsxtr@record@setting@off
 \else
  \protected@edef\@glsxtr@mglslabel{#1}%
  \ifdef\@glsxtr@mglsreflist
  {%
     \expandafter\DTLifinlist\expandafter{\@glsxtr@mglslabel}%
     {\@glsxtr@mglsreflist}{}%
     {%
       \xappto\@glsxtr@mglsreflist{,\expandonce\@glsxtr@mglslabel}%
       \if@mgls@writeseparaterefs
         \protected@write\@auxout{}{\string\@glsxtr@mglsrefs{#1}}%
       \fi
     }%
  }%
  {%
     \global\let\@glsxtr@mglsreflist\@glsxtr@mglslabel
     \if@mgls@writeseparaterefs
       \protected@write\@auxout{}{\string\@glsxtr@mglsrefs{#1}}%
     \else
       \AtEndDocument{\immediate\protected@write\@auxout{}%
         {\string\@glsxtr@mglsrefs{\@glsxtr@mglsreflist}}}%
     \fi
     \@mgls@disable@writeseparateref@cond
  }%
 \fi
}
\newcommand{\@glsxtr@mglsrefs}[1]{}
\newif\if@mgls@writeseparaterefs \@mgls@writeseparaterefsfalse
\newcommand{\mglsWriteSeparateRefsTrue}{\global\@mgls@writeseparaterefstrue}
\newcommand{\mglsWriteSeparateRefsFalse}{\global\@mgls@writeseparaterefsfalse}
\newcommand*{\@mgls@disable@writeseparateref@cond}{%
  \gdef\mglsWriteSeparateRefsTrue{\PackageError{glossaries-extra}%
   {Too late to use \string\mglsWriteSeparateRefsTrue}%
   {\string\mglsWriteSeparateRefsTrue\space can only be used before
   the first instance of any \string\mgls-like command}}%
  \gdef\mglsWriteSeparateRefsFalse{\PackageError{glossaries-extra}%
   {Too late to use \string\mglsWriteSeparateRefsFalse}%
   {\string\mglsWriteSeparateRefsFalse\space can only be used before
   the first instance of any \string\mgls-like command}}%
}
\newcommand{\glsxtr@newmgls}[5]{%
  \edef\@glsxr@newmgls@do{%
    \noexpand\newrobustcmd*{\expandonce{\csname #1\endcsname}}%
```

```
      {\noexpand\@gls@hyp@opt\expandonce{\csname ns@glsxtr@#1\endcsname}}%
     \noexpand\newcommand*{\expandonce{\csname ns@glsxtr@#1\endcsname}}[2][]{%
      \noexpand\new@ifnextchar[%
       {\expandonce{\csname glsxtr@#1\endcsname}{####1}{####2}}%
       {\expandonce{\csname glsxtr@#1\endcsname}{####1}{####2}[]}%
     }%
     \noexpand\def\expandonce{\csname glsxtr@#1\endcsname}####1####2[####3]{%
       \noexpand\def\noexpand\glsxtrcurrentmglscsname{#1}%
       \noexpand\glsxtr@mgls@inner{####1}{####2}{####3}%
        {\noexpand#2}{\noexpand#3}{\noexpand#4}{\noexpand#5}%
     }%
   }%
   \@glsxr@newmgls@do
   \ifx\@glsxtr@record@setting\@glsxtr@record@setting@off
   \else
     \ifdef\@glsxtr@mglslikelist
     {\xappto\@glsxtr@mglslikelist{,#1}}%
     {%
       \gdef\@glsxtr@mglslikelist{#1}%
       \AtEndDocument{\immediate\protected@write\@auxout{}%
         {\string\@glsxtr@mglslike{\@glsxtr@mglslikelist}}}%
     }%
   \fi
}
\newcommand*{\@glsxtr@mglslike}[1]{}
\newcommand*{\GlsXtrMglsOrGls}[2]{%
  \def\@glsxtr@mgls@or@gls@mcs{#1}%
  \def\@glsxtr@mgls@or@gls@gcs{#2}%
  \@ifstar{\s@GlsXtrMglsOrGls}%
  {%
   \@ifnextchar+{\PLUS\@firstoftwo{\p@GlsXtrMglsOrGls}}%
   {%
     \ifdefempty\@gls@alt@hyp@opt@char\@GlsXtrMglsOrGls\alt@GlsXtrMglsOrGls
   }%
  }%
}
\newcommand*{\alt@GlsXtrMglsOrGls}{
 \expandafter\@ifnextchar\@gls@alt@hyp@opt@char
 {\@firstoftwo{\@alt@GlsXtrMglsOrGls}}{\@GlsXtrMglsOrGls}%
}
\newcommand*{\@GlsXtrMglsOrGls}[2][]{%
  \glsxtrifmulti{#2}%
  {\@glsxtr@mgls@or@gls@mcs[#1]{#2}}%
  {\@glsxtr@mgls@or@gls@gcs[#1]{#2}}%
}
\newcommand*{\s@GlsXtrMglsOrGls}[2][]{%
  \glsxtrifmulti{#2}%
  {\@glsxtr@mgls@or@gls@mcs*[#1]{#2}}%
  {\@glsxtr@mgls@or@gls@gcs*[#1]{#2}}%
}
```

```
\newcommand*{\p@GlsXtrMglsOrGls}[2][]{%
  \glsxtrifmulti{#2}%
  {\@glsxtr@mgls@or@gls@mcs+[#1]{#2}}%
  {\@glsxtr@mgls@or@gls@gcs+[#1]{#2}}%
}
\newcommand*{\@alt@GlsXtrMglsOrGls}[2][]{%
  \glsxtrifmulti{#2}%
  {\expandafter\@glsxtr@mgls@or@gls@mcs\@gls@alt@hyp@opt@char[#1]{#2}}%
  {\expandafter\@glsxtr@mgls@or@gls@gcs\@gls@alt@hyp@opt@char[#1]{#2}}%
}
\glsxtr@newmgls{mgls}{\@gls@}{\@gls@}{\@gls@}{\@gls@}%
\glsxtr@newmgls{mglspl}{\@glspl@}{\@glspl@}{\@glspl@}{\@glspl@}%
\glsxtr@newmgls{mglsmainpl}{\@gls@}{\@gls@}{\@glspl@}{\@glspl@}%
\glsxtr@newmgls{Mgls}{\@Gls@}{\@gls@}{\@Gls@}{\@gls@}%
\glsxtr@newmgls{Mglspl}{\@Glspl@}{\@glspl@}{\@Glspl@}{\@glspl@}%
\glsxtr@newmgls{Mglsmainpl}{\@Gls@}{\@gls@}{\@Glspl@}{\@glspl@}%
\glsxtr@newmgls{MGls}{\@Gls@}{\@Gls@}{\@Gls@}{\@Gls@}%
\glsxtr@newmgls{MGlspl}{\@Glspl@}{\@Glspl@}{\@Glspl@}{\@Glspl@}%
\glsxtr@newmgls{MGlsmainpl}{\@Gls@}{\@Gls@}{\@Glspl@}{\@Glspl@}%
\glsxtr@newmgls{MGLS}{\@GLS@}{\@GLS@}{\@GLS@}{\@GLS@}%
\glsxtr@newmgls{MGLSpl}{\@GLSpl@}{\@GLSpl@}{\@GLSpl@}{\@GLSpl@}%
\glsxtr@newmgls{MGLSmainpl}{\@GLS@}{\@GLS@}{\@GLSpl@}{\@GLSpl@}%
\def\@glslongortext#1#2[#3]{%
  \ifglshaslong{#2}{\@glsxtrlong{#1}{#2}[#3]}{\@glstext@{#1}{#2}[#3]}%
}
\def\@glsshortortext#1#2[#3]{%
  \ifglshasshort{#2}{\@glsxtrshort{#1}{#2}[#3]}{\@glstext@{#1}{#2}[#3]}%
}
\def\@glsfullorfirst#1#2[#3]{%
  \ifglshasshort{#2}{\@glsxtr@full{#1}{#2}[#3]}{\@glsfirst@{#1}{#2}[#3]}%
}
\def\@Glslongortext#1#2[#3]{%
  \ifglshaslong{#2}{\@Glsxtrlong{#1}{#2}[#3]}{\@Glstext@{#1}{#2}[#3]}%
}
\def\@Glsshortortext#1#2[#3]{%
  \ifglshasshort{#2}{\@Glsxtrshort{#1}{#2}[#3]}{\@Glstext@{#1}{#2}[#3]}%
}
\def\@Glsfullorfirst#1#2[#3]{%
  \ifglshasshort{#2}{\@Glsxtr@full{#1}{#2}[#3]}{\@Glsfirst@{#1}{#2}[#3]}%
}
\glsxtr@newmgls{mglsshort}%
{\@glsshortortext}{\@glsshortortext}{\@glsshortortext}{\@glsshortortext}%
\glsxtr@newmgls{mglslong}%
{\@glslongortext}{\@glslongortext}{\@glslongortext}{\@glslongortext}%
\glsxtr@newmgls{mglsfull}%
{\@glsfullorfirst}{\@glsfullorfirst}{\@glsfullorfirst}{\@glsfullorfirst}%
\glsxtr@newmgls{Mglsshort}%
{\@Glsshortortext}{\@glsshortortext}{\@Glsshortortext}{\@glsshortortext}%
\glsxtr@newmgls{Mglslong}%
{\@Glslongortext}{\@glslongortext}{\@Glslongortext}{\@glslongortext}%
```

```
\glsxtr@newmgls{Mglsfull}%
{\@Glsfullorfirst}{\@glsfullorfirst}{\@Glsfullorfirst}{\@glsfullorfirst}%
\glsxtr@newmgls{mglsname}%
 {\@glsname@}{\@glsname@}{\@glsname@}{\@glsname@}%
\glsxtr@newmgls{Mglsname}%
 {\@Glsname@}{\@glsname@}{\@Glsname@}{\@glsname@}%
\glsxtr@newmgls{MGlsname}%
 {\@Glsname@}{\@Glsname@}{\@Glsname@}{\@Glsname@}%
\def\@glssymbolorgls#1#2[#3]{%
  \ifglshassymbol{#2}{\@glssymbol@{#1}{#2}[#3]}{\@gls@{#1}{#2}[#3]}%
}
\def\@glssymbolorGls#1#2[#3]{%
  \ifglshassymbol{#2}{\@glssymbol@{#1}{#2}[#3]}{\@Gls@{#1}{#2}[#3]}%
}
\glsxtr@newmgls{mglssymbol}%
 {\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
\glsxtr@newmgls{Mglssymbol}%
 {\@glssymbolorGls}{\@glssymbolorgls}{\@glssymbolorGls}{\@glssymbolorgls}%
\glsxtr@newmgls{MGlssymbol}%
 {\@glssymbolorGls}{\@glssymbolorGls}{\@glssymbolorGls}{\@glssymbolorGls}%
\newcommand{\mglsfield}{useri}
\def\@glsfieldorgls#1#2[#3]{%
  \glsxtrifhasfield{\mglsfield}{#2}%
  {\@glsdisp[#1]{#2}{\glscurrentfieldvalue#3}}%
  {\@gls@{#1}{#2}[#3]}%
}
\def\@Glsfieldorgls#1#2[#3]{%
  \glsxtrifhasfield{\mglsfield}{#2}%
  {\@glsdisp[#1]{#2}{\xmakefirstuc\glscurrentfieldvalue#3}}%
  {\@Gls@{#1}{#2}[#3]}%
}
\glsxtr@newmgls{mglsusefield}%
 {\@glsfieldorgls}{\@glsfieldorgls}{\@glsfieldorgls}{\@glsfieldorgls}%
\glsxtr@newmgls{Mglsusefield}%
 {\@Glsfieldorgls}{\@glsfieldorgls}{\@Glsfieldorgls}{\@glsfieldorgls}%
\glsxtr@newmgls{MGlsusefield}%
 {\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}%
\newcommand*{\mpglsWarning}{%
  \GlossariesExtraWarning{glossaries-prefix.sty is required for
   \string\mpgls\space family of commands (either load after
   glossaries-extra.sty or use the 'prefix' package option)}%
}
\def\@pglsorgls#1#2[#3]{%
  \ifdef\@pgls@{\@pgls@{#1}{#2}[#3]}{\mpglsWarning\@gls@{#1}{#2}[#3]}%
}
\def\@pglsorglspl#1#2[#3]{%
  \ifdef\@pglspl@{\@pglspl@{#1}{#2}[#3]}{\mpglsWarning\@glspl@{#1}{#2}[#3]}%
}
\def\@Pglsorgls#1#2[#3]{%
  \ifdef\@Pgls@{\@Pgls@{#1}{#2}[#3]}{\mpglsWarning\@Gls@{#1}{#2}[#3]}%
```

```
}
\def\@pglsorglspl#1#2[#3]{%
  \ifdef\@pglspl@{\@pglspl@{#1}{#2}[#3]}{\mpglsWarning\@glspl@{#1}{#2}[#3]}%
}
\def\@Pglsorglspl#1#2[#3]{%
  \ifdef\@Pglspl@{\@Pglspl@{#1}{#2}[#3]}{\mpglsWarning\@Glspl@{#1}{#2}[#3]}%
}
\def\@PGLSorgls#1#2[#3]{%
  \ifdef\@PGLS@{\@PGLS@{#1}{#2}[#3]}{\mpglsWarning\@GLS@{#1}{#2}[#3]}%
}
\def\@PGLSorglspl#1#2[#3]{%
  \ifdef\@PGLSpl@{\@PGLSpl@{#1}{#2}[#3]}{\mpglsWarning\@GLSpl@{#1}{#2}[#3]}%
}
\glsxtr@newmgls{mpgls}{\@pglsorgls@}{\@gls@}{\@pglsorgls@}{\@gls@}%
\glsxtr@newmgls{mpglspl}{\@pglsorglspl@}{\@glspl@}{\@pglsorglspl@}{\@glspl@}%
\glsxtr@newmgls{mpglsmainpl}{\@pglsorgls@}{\@gls@}{\@pglsorglspl@}{\@glspl@}%
\glsxtr@newmgls{Mpgls}{\@Pglsorgls@}{\@gls@}{\@Pglsorgls@}{\@gls@}%
\glsxtr@newmgls{Mpglspl}{\@Pglsorglspl@}{\@glspl@}{\@Pglsorglspl@}{\@glspl@}%
\glsxtr@newmgls{Mpglsmainpl}{\@Pglsorgls@}{\@gls@}{\@Pglsorglspl@}{\@glspl@}%
\glsxtr@newmgls{MPGls}{\@Pglsorgls@}{\@Gls@}{\@Pglsorgls@}{\@Gls@}%
\glsxtr@newmgls{MPGlspl}{\@Pglsorglspl@}{\@Glspl@}{\@Pglsorglspl@}{\@Glspl@}%
\glsxtr@newmgls{MPGlsmainpl}{\@Pglsorgls@}{\@Gls@}{\@Pglsorglspl@}{\@Glspl@}%
\glsxtr@newmgls{MPGLS}{\@PGLSorgls@}{\@GLS@}{\@PGLSorgls@}{\@GLS@}%
\glsxtr@newmgls{MPGLSpl}{\@PGLSorglspl@}{\@GLSpl@}{\@PGLSorglspl@}{\@GLSpl@}%
\glsxtr@newmgls{MPGLSmainpl}{\@PGLSorgls@}{\@GLS@}{\@PGLSorglspl@}{\@GLSpl@}%
\newcommand*{\RequireGlossariesExtraLang}[1]{%
  \@ifundefined{ver@glossariesxtr-#1.ldf}{\input{glossariesxtr-#1.ldf}}{}%
}
\newcommand*{\ProvidesGlossariesExtraLang}[1]{%
  \ProvidesFile{glossariesxtr-#1.ldf}%
}
\newcommand{\glsxtr@loaddialect}{%
  \IfTrackedLanguageFileExists{\this@dialect}%
  {glossariesxtr-}% prefix
  {.ldf}%
  {%
    \RequireGlossariesExtraLang{\CurrentTrackedTag}%
  }%
  {}% not found
  \@glsxtrdialecthook
}
\@ifpackageloaded{tracklang} {%
  \AnyTrackedLanguages
  {%
    \ForEachTrackedDialect{\this@dialect}{\glsxtr@loaddialect}%
  }%
  {}%
} {}
\@@glsxtr@redefstyles
\@@glsxtr@do@style
```

9.2 Rollback v1.48 (glossaries-extra-bib2gls-2021-11-22.sty)

Version 1.48 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-extra-bib2gls}[2021/11/22 v1.48 (NLCT)]
\ifglsacronym
  \providecommand*{\printunsrtacronyms}[1][]{%
    \printunsrtglossary[type=\acronymtype,#1]}%
\fi
\ifglossaryexists{index}
{
  \providecommand*{\printunsrtindex}[1][]{%
    \printunsrtglossary[type=index,#1]}%
}{}
\ifglossaryexists{symbols}
{
  \providecommand*{\printunsrtsymbols}[1][]{%
    \printunsrtglossary[type=symbols,#1]}%
}{}
\ifglossaryexists{numbers}
{
  \providecommand*{\printunsrtnumbers}[1][]{%
    \printunsrtglossary[type=numbers,#1]}%
}{}
\ifglossaryexists{abbreviations}
{
  \providecommand*{\printunsrtabbreviations}[1][]{%
    \printunsrtglossary[type=abbreviations,#1]}%
}{}
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    {\let\bibglsdelimN\glsnumlistsep
     \let\bibglslastDelimN\glsnumlistlastsep
     \glsxtrusefield{#1}{location}%
    }%
  }%
}
\robustify\glsdisplaynumberlist
\renewcommand*{\glsentrynumberlist}[1]{\glsxtrusefield{#1}{location}}
\newcommand*{\glshex}{\string\u}
\newcommand*{\glscapturedgroup}{\string\$}
\newcommand*{\GlsXtrIfHasNonZeroChildCount}{%
  \@ifstar\s@GlsXtrIfHasNonZeroChildCount\@GlsXtrIfHasNonZeroChildCount
}
\newcommand*{\@GlsXtrIfHasNonZeroChildCount}[3]{%
  \@GlsXtrIfFieldNonZero{childcount}{#1}{#2}{#3}%
}
\newcommand*{\s@GlsXtrIfHasNonZeroChildCount}[3]{%
  \s@GlsXtrIfFieldNonZero{childcount}{#1}{#2}{#3}%
```

```
}
\newcommand*{\glsxtrprovidecommand}{\providecommand}
\newcommand*{\glsrenewcommand}{\@star@or@long\glsxtr@renewcommand}
\newcommand*{\glsxtr@renewcommand}[1]{%
 \begingroup \escapechar\m@ne\xdef\@gtempa{{\string#1}}\endgroup
 \expandafter\@ifundefined\@gtempa
   {%
     \GlossariesExtraWarning{can't redefine \noexpand#1(not already defined)}%
   }%
   \relax
  \relax
 \let\@ifdefinable\@rc@ifdefinable
 \new@command#1%
}
\newcommand*{\glsxtr@wrglossarylocation}[2]{#1}
\ifdef\hyperref
{%
  \newcommand*{\GlsXtrIndexCounterLink}[2]{%
    \glsxtrifhasfield{indexcounter}{#2}%
    {\hyperref[wrglossary.\glscurrentfieldvalue]{#1}}%
    {#1}%
  }
}
{
  \newcommand*{\GlsXtrIndexCounterLink}[2]{#1}
}
\newcommand*{\GlsXtrDualField}{dual}
\newcommand*{\GlsXtrDualBackLink}[2]{%
  \glsxtrifhasfield{\GlsXtrDualField}{#2}%
  {\glshyperlink[#1]{\glscurrentfieldvalue}}%
  {#2}%
}
\newcommand*{\GlsXtrBibTeXEntryAliases}{%
  article=bibtexentry,
  book=bibtexentry,
  booklet=bibtexentry,
  conference=bibtexentry,
  inbook=bibtexentry,
  incollection=bibtexentry,
  inproceedings=bibtexentry,
  manual=bibtexentry,
  mastersthesis=bibtexentry,
  misc=bibtexentry,
  phdthesis=bibtexentry,
  proceedings=bibtexentry,
  techreport=bibtexentry,
  unpublished=bibtexentry
}
\newcommand*{\GlsXtrProvideBibTeXFields}{%
  \glsaddstoragekey{address}{}{\glsxtrbibaddress}%
```

```
    \glsaddstoragekey{author}{}{\glsxtrbibauthor}%
    \glsaddstoragekey{booktitle}{}{\glsxtrbibbooktitle}%
    \glsaddstoragekey{chapter}{}{\glsxtrbibchapter}%
    \glsaddstoragekey{edition}{}{\glsxtrbibedition}%
    \glsaddstoragekey{howpublished}{}{\glsxtrbibhowpublished}%
    \glsaddstoragekey{institution}{}{\glsxtrbibinstitution}%
    \glsaddstoragekey{journal}{}{\glsxtrbibjournal}%
    \glsaddstoragekey{month}{}{\glsxtrbibmonth}%
    \glsaddstoragekey{note}{}{\glsxtrbibnote}%
    \glsaddstoragekey{number}{}{\glsxtrbibnumber}%
    \glsaddstoragekey{organization}{}{\glsxtrbiborganization}%
    \glsaddstoragekey{pages}{}{\glsxtrbibpages}%
    \glsaddstoragekey{publisher}{}{\glsxtrbibpublisher}%
    \glsaddstoragekey{school}{}{\glsxtrbibschool}%
    \glsaddstoragekey{series}{}{\glsxtrbibseries}%
    \glsaddstoragekey{title}{}{\glsxtrbibtitle}%
    \glsaddstoragekey{bibtextype}{}{\glsxtrbibtype}%
    \glsaddstoragekey{volume}{}{\glsxtrbibvolume}%
}
\newcommand*{\glsxtrmultisupplocation}[3]{%
 {%
    \def\glsxtrsupplocationurl{#2}%
    \glshypernumber{#1}%
 }%
}
\newcommand*\glsxtrdisplaysupploc[5]{%
  \setentrycounter[#1]{#2}%
  \glsxtrmultisupplocation{#5}{#4}{#3}%
}
\ifundef\hyperlink
{
  \newcommand*{\glsxtrdisplaylocnameref}[8]{%
    \glsnoidxdisplayloc{#1}{#2}{#3}{#4}%
  }
}
{
  \newcommand*{\glsxtrdisplaylocnameref}[8]{%
    \ifcsdef{glsxtr#2locfmt}%
    {\glsxtrnamereflink{#3}{\csuse{glsxtr#2locfmt}{#4}{#5}}{#2.#7}{#8}}%
    {%
      \ifstrempty{#5}%
      {%
        \glsxtrnamereflink{#3}{#4}{#2.#7}{#8}%
      }%
      {%
        \ifstrequal{#2}{page}%
        {\glsxtrnamereflink{#3}{#4}{#2.#7}{#8}}%
        {\glsxtrnamereflink{#3}{#5}{#2.#7}{#8}}%
      }%
    }%
```

```
  }
}
\newcommand*{\glsxtrequationlocfmt}[2]{(#1)}
\newcommand*{\glsxtrnamereflink}[4]{%
  \begingroup
    \let\glshypernumber\@firstofone
    \ifstrempty{#4}%
    {\glsxtrfmtinternalnameref{#3}{#1}{#2}}%
    {\glsxtrfmtexternalnameref{#3}{#1}{#2}{#4}}%
  \endgroup
}
\newcommand{\glsxtrnameloclink}[6]{%
 \begingroup
  \setentrycounter[#1]{#2}%
  \def\glsxtr@locationhypertext{#5}%
  \let\glshypernumber\@firstofone
  \def\@glsnumberformat{#3}%
  \def\glsxtrsupplocationurl{#6}%
  \toks@={}%
  \@glsxtr@bibgls@removespaces#4 \@nil
 \endgroup
}
\def\@glsxtr@bibgls@removespaces#1 #2\@nil{%
 \toks@=\expandafter{\the\toks@#1}%
 \ifx\\#2\\%
   \edef\@glo@tmp{\the\toks@}%
   \ifx\@glo@tmp\empty
   \else
     \protected@edef\@glo@tmp{\glsentrycounter\@glo@counterprefix\the\toks@}%
     \ifdefvoid\glsxtrsupplocationurl
     {%
       \expandafter\glsxtrfmtinternalnameref\expandafter{\@glo@tmp}%
       {\@glsnumberformat}{\glsxtr@locationhypertext}%
     }%
     {%
       \expandafter\glsxtrfmtexternalnameref\expandafter{\@glo@tmp}%
       {\@glsnumberformat}{\glsxtr@locationhypertext}{\glsxtrsupplocationurl}%
     }%
   \fi
 \else
   \@gls@ReturnAfterFi{%
     \@glsxtr@bibgls@removespaces#2\@nil
   }%
 \fi
}
\newcommand*{\glsxtrfmtinternalnameref}[3]{%
  \csuse{#2}{\glsdohyperlink{#1}{#3}}%
}
\newcommand*{\glsxtrfmtexternalnameref}[4]{%
  \csuse{#2}{\hyperref{#4}{}{#1}{#3}}%
```

```
}
\newcommand*{\glsxtrSetWidest}[3]{%
  \ifdef\glsupdatewidest
  {%
    \ifdef\glslongextraUpdateWidest
    {%
      \ifstrempty{#1}
      {%
        \glsupdatewidest[#2]{#3}%
        \ifnum#2=0\relax
          \glslongextraUpdateWidest{#3}%
        \else
          \glslongextraUpdateWidestChild{#2}{#3}%
        \fi
      }%
      {%
        \apptoglossarypreamble[#1]{\glsupdatewidest[#2]{#3}}%
        \ifnum#2=0\relax
          \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
        \else
          \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
        \fi
      }%
    }%
    {%
      \ifstrempty{#1}
      {%
        \glsupdatewidest[#2]{#3}%
      }%
      {%
        \apptoglossarypreamble[#1]{\glsupdatewidest[#2]{#3}}%
      }%
    }%
  }%
  {%
    \ifdef\glssetwidest
    {%
      \ifdef\glslongextraUpdateWidest
      {%
        \ifstrempty{#1}
        {%
          \glssetwidest[#2]{#3}%
          \ifnum#2=0\relax
            \glslongextraUpdateWidest{#3}%
          \else
            \glslongextraUpdateWidestChild{#2}{#3}%
          \fi
        }%
        {%
          \apptoglossarypreamble[#1]{\glssetwidest[#2]{#3}}%
```

```
      \ifnum#2=0\relax
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
      \else
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
      \fi
    }%
  }%
  {%
    \ifstrempty{#1}
    {%
      \glssetwidest[#2]{#3}%
    }%
    {%
      \apptoglossarypreamble[#1]{\glssetwidest[#2]{#3}}%
    }%
  }%
  }%
  }%
  {%
    \ifdef\glslongextraUpdateWidest
    {%
      \ifstrempty{#1}
      {%
        \ifnum#2=0\relax
          \glslongextraUpdateWidest{#3}%
        \else
          \glslongextraUpdateWidestChild{#2}{#3}%
        \fi
      }%
      {%
        \ifnum#2=0\relax
          \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
        \else
          \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
        \fi
      }%
    }%
    {}%
  }%
  }%
}
\newcommand*{\glsxtrSetWidestFallback}[2]{%
  \ifnum#1=0\relax
   \ifdef\glsFindWidestTopLevelName
   {%
     \glsFindWidestTopLevelName[#2]%
   }%
   {%
     \GlossariesExtraWarning{You need stylemods={tree} to
       provide a fallback for set-widest}%
   }%
```

1126

```
    \else
     \ifdef\glsFindWidestLevelTwo
     {%
       \glsFindWidestLevelTwo[#2]%
       \ifdef\glslongextraUpdateWidestChild
       {%
        \glslongextraUpdateWidestChild{#1}{\csuse{@glswidestnamei}}%
        \glslongextraUpdateWidestChild{#1}{\csuse{@glswidestnameii}}%
       }%
       {}%
     }%
     {%
       \GlossariesExtraWarning{You need stylemods={tree} to
         provide a fallback for set-widest}%
     }%
    \fi
}
\newcommand*{\@glsxtr@labelprefixes}{}
\newcommand*{\glsxtrclearlabelprefixes}{%
  \renewcommand*{\@glsxtr@labelprefixes}{}%
}
\newcommand*{\glsxtraddlabelprefix}[1]{%
  \ifstrempty{#1}%
  {\glsxtraddlabelprefix{\empty}}%
  {%
    \ifdefempty\@glsxtr@labelprefixes
    {\def\@glsxtr@labelprefixes{#1}}%
    {\appto\@glsxtr@labelprefixes{,#1}}%
  }%
}
\newcommand*{\glsxtrprependlabelprefix}[1]{%
  \ifstrempty{#1}%
  {\glsxtrprependlabelprefix{\empty}}%
  {%
    \ifdefempty\@glsxtr@labelprefixes
    {\def\@glsxtr@labelprefixes{#1}}%
    {\preto\@glsxtr@labelprefixes{#1,}}%
  }%
}
\newcommand*{\glsxtrifinlabelprefixlist}[3]{%
  \ifstrempty{#1}%
  {\glsxtrifinlabelprefixlist{\empty}{#2}{#3}}%
  {%
    \DTLifinlist{#1}{\@glsxtr@labelprefixes}{#2}{#3}%
  }%
}
\AtBeginDocument{%
 \protected@write\@auxout{}{\string\providecommand{\string\@glsxtr@prefixlabellist}[1]{}}%
 \protected@write\@auxout{}{\string\@glsxtr@prefixlabellist{\@glsxtr@labelprefixes}}%
}
```

```
\newcommand*{\@glsxtr@get@prefixedlabel}[1]{%
 \begingroup
 \protected@edef\@gls@thislabel{#1}%
 \@for\@glsxtr@prefix:=\@glsxtr@labelprefixes\do
 {%
    \protected@edef\@gls@thislabel{\@glsxtr@prefix#1}%
    \ifglsentryexists{\@gls@thislabel}{\@endfortrue}{}%
 }%
 \edef\@glo@tmp{\endgroup\noexpand\def\noexpand\@gls@thislabel{\@gls@thislabel}}\@glo@tmp
}
\newrobustcmd*{\dgls}{\@gls@hyp@opt\@dgls}
\newcommand*{\@dgls}[2][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \new@ifnextchar[{\@gls@{#1}{\@gls@thislabel}}{\@gls@{#1}{\@gls@thislabel}[]}%
}
\newrobustcmd*{\dglspl}{\@gls@hyp@opt\@dglspl}
\newcommand*{\@dglspl}[2][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \new@ifnextchar[{\@glspl@{#1}{\@gls@thislabel}}{\@glspl@{#1}{\@gls@thislabel}[]}%
}
\newrobustcmd*{\dGls}{\@gls@hyp@opt\@dGls}
\newcommand*{\@dGls}[2][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \new@ifnextchar[{\@Gls@{#1}{\@gls@thislabel}}{\@Gls@{#1}{\@gls@thislabel}[]}%
}
\newrobustcmd*{\dGlspl}{\@gls@hyp@opt\@dGlspl}
\newcommand*{\@dGlspl}[2][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \new@ifnextchar[{\@Glspl@{#1}{\@gls@thislabel}}{\@Glspl@{#1}{\@gls@thislabel}[]}%
}
\newrobustcmd*{\dGLS}{\@gls@hyp@opt\@dGLS}
\newcommand*{\@dGLS}[2][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \new@ifnextchar[{\@GLS@{#1}{\@gls@thislabel}}{\@GLS@{#1}{\@gls@thislabel}[]}%
}
\newrobustcmd*{\dGLSpl}{\@gls@hyp@opt\@dGLSpl}
\newcommand*{\@dGLSpl}[2][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \new@ifnextchar[{\@GLSpl@{#1}{\@gls@thislabel}}{\@GLSpl@{#1}{\@gls@thislabel}[]}%
}
\newrobustcmd*{\dglslink}[3][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \glslink[#1]{\@gls@thislabel}{#3}%
}
\newrobustcmd*{\dglsdisp}[3][]{%
  \@glsxtr@get@prefixedlabel{#2}%
  \glsdisp[#1]{\@gls@thislabel}{#3}%
}
\newrobustcmd*{\glsxtrmultientryadjustedname}[4]{%
  \bgroup
```

```
    \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
    \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
    \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
    \let\@glsxtrmultientryadjustednameother\glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefmt\glsxtrmultientryadjustednamefmt
    \let\@glsxtrmultientryadjustednamefirstother\glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefirstfmt\glsxtrmultientryadjustednamefmt
    \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
\newrobustcmd*{\Glsxtrmultientryadjustedname}[4]{%
  \bgroup
    \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
    \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
    \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
    \let\@glsxtrmultientryadjustednameother\glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefmt\glsxtrmultientryadjustednamefmt
    \let\@glsxtrmultientryadjustednamefirstother\Glsxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefirstfmt\Glsxtrmultientryadjustednamefmt
    \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
\newrobustcmd*{\GlsXtrmultientryadjustedname}[4]{%
  \bgroup
    \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
    \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
    \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
    \let\@glsxtrmultientryadjustednameother\GlsXtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefmt\GlsXtrmultientryadjustednamefmt
    \let\@glsxtrmultientryadjustednamefirstother\GlsXtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefirstfmt\GlsXtrmultientryadjustednamefmt
    \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
\newrobustcmd*{\GLSxtrmultientryadjustedname}[4]{%
  \bgroup
    \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
    \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
    \let\@glsxtrmultientryadjustednamepostsep\glsxtrmultientryadjustednamepostsep
    \let\@glsxtrmultientryadjustednameother\GLSxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefmt\GLSxtrmultientryadjustednamefmt
    \let\@glsxtrmultientryadjustednamefirstother\GLSxtrmultientryadjustednameother
    \let\@glsxtrmultientryadjustednamefirstfmt\GLSxtrmultientryadjustednamefmt
    \@glsxtrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
\newcommand*{\@glsxtrmultientryadjustedname}[4]{%
  \letcs\mglscurrentmainlabel{@gls@combined@#4@main}%
  \letcs\mglscurrentmainlist{@gls@combined@#4@list}%
  \letcs\mglscurrentmainoptions{@gls@combined@#4@options}%
```

```
        \ifblank{#1}%
        {%
          \@glsxtrmultientryadjustednamefirstfmt{#2}%
        }%
        {%
          \def\@mgls@previouslabel{}%
          \let\@gls@xtradjustedother\@glsxtrmultientryadjustednamefirstother
          \@for\mglscurrentlabel:=#1\do{%
              \ifx\@mgls@previouslabel\empty
              \else
                \@glsxtrmultientryadjustednamesep{\@mgls@previouslabel}{\mglscurrentlabel}%
              \fi
              \@gls@xtradjustedother{\mglscurrentlabel}%
              \let\@mgls@previouslabel\mglscurrentlabel
              \let\@gls@xtradjustedother\@glsxtrmultientryadjustednameother
          }%
          \@glsxtrmultientryadjustednamepresep{\@mgls@previouslabel}{\mglscurrentmainlabel}%
          \@glsxtrmultientryadjustednamefmt{#2}%
        }%
        \ifblank{#3}%
        {}%
        {%
          \let\@mgls@previouslabel\mglscurrentmainlabel
          \let\@gls@xtrmultientryadjustednamesep\@glsxtrmultientryadjustednamepostsep
          \@for\mglscurrentlabel:=#3\do{%
              \@gls@xtrmultientryadjustednamesep{\@mgls@previouslabel}{\mglscurrentlabel}%
              \@glsxtrmultientryadjustednameother{\mglscurrentlabel}%
              \let\@mgls@previouslabel\mglscurrentlabel
              \let\@gls@xtrmultientryadjustednamesep\@glsxtrmultientryadjustednamesep
          }%
        }%
}
\newcommand*{\glsxtrmultientryadjustednamesep}{\glscombinedfirstsepfirst}
\newcommand*{\glsxtrmultientryadjustednamepresep}{\glsxtrmultientryadjustednamesep}
\newcommand*{\glsxtrmultientryadjustednamepostsep}{\glsxtrmultientryadjustednamesep}
\newcommand*{\glsxtrmultientryadjustednamefmt}[1]{#1}
\newcommand*{\glsxtrmultientryadjustednameother}[1]{\glsentryname{#1}}
\newcommand*{\Glsxtrmultientryadjustednamefmt}[1]{\makefirstuc{#1}}
\newcommand*{\Glsxtrmultientryadjustednameother}[1]{\Glsentryname{#1}}
\newcommand*{\GlsXtrmultientryadjustednameother}[1]{%
 \glsentrytitlecase{#1}{name}}
\ifdef\glscapitalisewords
{%
  \newcommand*{\GlsXtrmultientryadjustednamefmt}[1]{\glscapitalisewords{#1}}
}
{
  \newcommand*{\GlsXtrmultientryadjustednamefmt}[1]{\capitalisewords{#1}}
}
\newcommand*{\GLSxtrmultientryadjustednameother}[1]{%
 \mfirstucMakeUppercase{\glsentryname{#1}}}
```

1130

```
\newcommand*{\GLSxtrmultientryadjustednamefmt}[1]{\mfirstucMakeUppercase{#1}}
\providecommand*{\Alpha}{\mathrm{A}}
\providecommand*{\Beta}{\mathrm{B}}
\providecommand*{\Epsilon}{\mathrm{E}}
\providecommand*{\Zeta}{\mathrm{Z}}
\providecommand*{\Eta}{\mathrm{H}}
\providecommand*{\Iota}{\mathrm{I}}
\providecommand*{\Kappa}{\mathrm{K}}
\providecommand*{\Mu}{\mathrm{M}}
\providecommand*{\Nu}{\mathrm{N}}
\providecommand*{\Omicron}{\mathrm{O}}
\providecommand*{\Rho}{\mathrm{P}}
\providecommand*{\Tau}{\mathrm{T}}
\providecommand*{\Chi}{\mathrm{X}}
\providecommand*{\Digamma}{\mathrm{F}}
\providecommand*{\omicron}{\mathit{o}}
\@ifpackageloaded{upgreek}%
{
  \providecommand*{\Upalpha}{\mathrm{A}}
  \providecommand*{\Upbeta}{\mathrm{B}}
  \providecommand*{\Upepsilon}{\mathrm{E}}
  \providecommand*{\Upzeta}{\mathrm{Z}}
  \providecommand*{\Upeta}{\mathrm{H}}
  \providecommand*{\Upiota}{\mathrm{I}}
  \providecommand*{\Upkappa}{\mathrm{K}}
  \providecommand*{\Upmu}{\mathrm{M}}
  \providecommand*{\Upnu}{\mathrm{N}}
  \providecommand*{\Upomicron}{\mathrm{O}}
  \providecommand*{\Uprho}{\mathrm{P}}
  \providecommand*{\Uptau}{\mathrm{T}}
  \providecommand*{\Upchi}{\mathrm{X}}
  \providecommand*{\upomicron}{\mathrm{o}}
}%
{}% upgreek.sty not loaded
\newcommand*{\glsxtrcontrolrules}{%
 \string'\glshex 200B\string'\string=\glshex 200C\string=\glshex 200D
 \string=\glshex 200E\string=\glshex 200F\string=\glshex 0000\string=\glshex 0001
 \string=\glshex 0002\string=\glshex 0003\string=\glshex 0004\string=\glshex 0005
 \string=\glshex 0006\string=\glshex 0007\string=\glshex 0008
 \string=\string'\glshex 0009\string'\string=\string'\glshex 000B\string'
 \string=\glshex 000E\string=\glshex 000F\string=\string'\glshex
0010\string'\string=\glshex 0011
 \string=\glshex 0012\string=\glshex 0013\string=\glshex 0014\string=\glshex 0015
 \string=\glshex 0016\string=\glshex 0017\string=\glshex 0018\string=\glshex 0019
 \string=\glshex 001A\string=\glshex 001B\string=\glshex 001C\string=\glshex 001D
 \string=\glshex 001E\string=\glshex 001F\string=\glshex 007F\string=\glshex 0080
 \string=\glshex 0081\string=\glshex 0082\string=\glshex 0083\string=\glshex 0084
 \string=\glshex 0085\string=\glshex 0086\string=\glshex 0087\string=\glshex 0088
 \string=\glshex 0089\string=\glshex 008A\string=\glshex 008B\string=\glshex 008C
 \string=\glshex 008D\string=\glshex 008E\string=\glshex 008F\string=\glshex 0090
```

```
\string=\glshex 0091\string=\glshex 0092\string=\glshex 0093\string=\glshex 0094
\string=\glshex 0095\string=\glshex 0096\string=\glshex 0097\string=\glshex 0098
\string=\glshex 0099\string=\glshex 009A\string=\glshex 009B\string=\glshex 009C
\string=\glshex 009D\string=\glshex 009E\string=\glshex 009F
}
\newcommand*{\glsxtrspacerules}{%
\string' \string'\string;
\string'\glshex 00A0\string'\string;
\string'\glshex 2000\string'\string;
\string'\glshex 2001\string'\string;
\string'\glshex 2002\string'\string;
\string'\glshex 2003\string'\string;
\string'\glshex 2004\string'\string;
\string'\glshex 2005\string'\string;
\string'\glshex 2006\string'\string;
\string'\glshex 2007\string'\string;
\string'\glshex 2008\string'\string;
\string'\glshex 2009\string'\string;
\string'\glshex 200A\string'\string;
\string'\glshex 3000\string'
}
\newcommand*{\glsxtrnonprintablerules}{%
\string'\glshex FEFF\string'\string;
\string'\glshex 000A\string'\string;
\string'\glshex 0009\string'\string;
\string'\glshex 000C\string'\string;
\string'\glshex 000B\string'
}


\newcommand*{\glsxtrcombiningdiacriticrules}{%
\glsxtrcombiningdiacriticIrules\string;
\glsxtrcombiningdiacriticIIrules\string;
\glsxtrcombiningdiacriticIIIrules\string;
\glsxtrcombiningdiacriticIVrules
}
\newcommand*{\glsxtrcombiningdiacriticIrules}{%
\glshex 0301\string;% combining acute
\glshex 0300\string;% combining grave
\glshex 0306\string;% combining breve
\glshex 0302\string;% combining circumflex
\glshex 030C\string;% combining caron
\glshex 030A\string;% combining ring
\glshex 030D\string;% combining vertical line above
\glshex 0308\string;% combining diaeresis
\glshex 030B\string;% combining double acute
\glshex 0303\string;% combining tilde
\glshex 0307\string;% combining dot above
\glshex 0304% combining macron
}
\newcommand*{\glsxtrcombiningdiacriticIIrules}{%
```

```
 \glshex 0337\string;% combining short solidus overlay
 \glshex 0327\string;% combining cedilla
 \glshex 0328\string;% combining ogonek
 \glshex 0323\string;% combining dot below
 \glshex 0332\string;% combining low line
 \glshex 0305\string;% combining overline
 \glshex 0309\string;% combining hook above
 \glshex 030E\string;% combining double vertical line above
 \glshex 030F\string;% combining double grave accent
 \glshex 0310\string;% combining candrabindu
 \glshex 0311\string;% combining inverted breve
 \glshex 0312\string;% combining turned comma above
 \glshex 0313\string;% combining comma above
 \glshex 0314\string;% combining reversed comma above
 \glshex 0315\string;% combining comma above right
 \glshex 0316\string;% combining grave accent below
 \glshex 0317% combining acute accent below
}
\newcommand*{\glsxtrcombiningdiacriticIIIrules}{%
 \glshex 0318\string;% combining left tack below
 \glshex 0319\string;% combining right tack below
 \glshex 031A\string;% combining left angle above
 \glshex 031B\string;% combining horn
 \glshex 031C\string;% combining left half ring below
 \glshex 031D\string;% combining up tack below
 \glshex 031E\string;% combining down tack below
 \glshex 031F\string;% combining plus sign below
 \glshex 0320\string;% combining minus sign below
 \glshex 0321\string;% combining palatalized hook below
 \glshex 0322\string;% combining retroflex hook below
 \glshex 0324\string;% combining diaresis below
 \glshex 0325\string;% combining ring below
 \glshex 0326\string;% combining comma below
 \glshex 0329\string;% combining vertical line below
 \glshex 032A\string;% combining bridge below
 \glshex 032B\string;% combining inverted double arch below
 \glshex 032C\string;% combining caron below
 \glshex 032D\string;% combining circumflex accent below
 \glshex 032E\string;% combining breve below
 \glshex 032F\string;% combining inverted breve below
 \glshex 0330\string;% combining tilde below
 \glshex 0331\string;% combining macron below
 \glshex 0333\string;% combining double low line
 \glshex 0334\string;% combining tilde overlay
 \glshex 0335\string;% combining short stroke overlay
 \glshex 0336\string;% combining long stroke overlay
 \glshex 0338\string;% combining long solidus overlay
 \glshex 0339\string;% combining combining right half ring below
 \glshex 033A\string;% combining inverted bridge below
 \glshex 033B\string;% combining square below
```

```
 \glshex 033C\string;% combining seagull below
 \glshex 033D\string;% combining x above
 \glshex 033E\string;% combining vertical tilde
 \glshex 033F\string;% combining double overline
 \glshex 0342\string;% combining Greek perispomeni
 \glshex 0344\string;% combining Greek dialytika tonos
 \glshex 0345\string;% combining Greek ypogegrammeni
 \glshex 0360\string;% combining double tilde
 \glshex 0361\string;% combining double inverted breve
 \glshex 0483\string;% combining Cyrillic titlo
 \glshex 0484\string;% combining Cyrillic palatalization
 \glshex 0485\string;% combining Cyrillic dasia pneumata
 \glshex 0486% combining Cyrillic psili pneumata
}
\newcommand*{\glsxtrcombiningdiacriticIVrules}{%
 \glshex 20D0\string;% combining left harpoon above
 \glshex 20D1\string;% combining right harpoon above
 \glshex 20D2\string;% combining long vertical line overlay
 \glshex 20D3\string;% combining short vertical line overlay
 \glshex 20D4\string;% combining anticlockwise arrow above
 \glshex 20D5\string;% combining clockwise arrow above
 \glshex 20D6\string;% combining left arrow above
 \glshex 20D7\string;% combining right arrow above
 \glshex 20D8\string;% combining ring overlay
 \glshex 20D9\string;% combining clockwise ring overlay
 \glshex 20DA\string;% combining anticlockwise ring overlay
 \glshex 20DB\string;% combining three dots above
 \glshex 20DC\string;% combining four dots above
 \glshex 20DD\string;% combining enclosing circle
 \glshex 20DE\string;% combining enclosing square
 \glshex 20DF\string;% combining enclosing diamond
 \glshex 20E0\string;% combining enclosing circle backslash
 \glshex 20E1%  combining left right arrow above
}
\newcommand*{\glsxtrhyphenrules}{%
 \string'\string-\string'\string;% ASCII hyphen
 \glshex 00AD\string;% soft hyphen
 \glshex 2010\string;% hyphen
 \glshex 2011\string;% non-breaking hyphen
 \glshex 2012\string;% figure dash
 \glshex 2013\string;% en dash
 \glshex 2014\string;% em dash
 \glshex 2015\string;% horizontal bar
 \glshex 2212\string=\glshex 207B\string=\glshex 208B% minus sign
}
\newcommand*{\glsxtrgeneralpuncrules}{%
  \glsxtrgeneralpuncIrules
  \string<\glsxtrcurrencyrules
  \string<\glsxtrgeneralpuncIIrules
}
```

```
\newcommand*{\glsxtrgeneralpuncIrules}{%
 \string'\glshex 005F\string'% underscore
 \string<\glshex 00AF% macron
 \string<\string'\glshex 002C\string'% comma
 \string<\string'\glshex 003B\string'% semi-colon
 \string<\string'\glshex 003A\string'% colon
 \string<\string'\glshex 0021\string'% exclamation mark
 \string<\string'\glshex 00A1% inverted exclamation mark
 \string<\string'\glshex 003F\string'% question mark
 \string<\string'\glshex 00BF% inverted question mark
 \string<\string'\glshex 002F\string'% solidus
 \string<\string'\glshex 002E\string'% full stop
 \string<\glshex 00B4% acute accent
 \string<\string'\glshex 0060\string'% grave accent
 \string<\string'\glshex 005E\string'% circumflex accent
 \string<\glshex 00A8% diaersis
 \string<\string'\glshex 007E\string'% tilde
 \string<\glshex 00B7% middle dot
 \string<\glshex 00B8% cedilla
 \string<\string'\glshex 0027\string'% straight apostrophe
 \string<\string'\glshex 0022\string'% straight double quote
 \string<\glshex 00AB% left guillemet
 \string<\glshex 00BB% right guillemet
 \string<\string'\glshex 0028\string'% left parenthesis
  \string=\glshex 207D\string=\glshex 208D% super/subscript left parenthesis
 \string<\string'\glshex 0029\string'% right parenthesis
  \string=\glshex 207E\string=\glshex 208E% super/subscript right parenthesis
 \string<\string'\glshex 005B\string'% left square bracket
 \string<\string'\glshex 005D\string'% right square bracket
 \string<\string'\glshex 007B\string'% left curly bracket
 \string<\string'\glshex 007D\string'% right curly bracket
 \string<\glshex 00A7% section sign
 \string<\glshex 00B6% pilcrow sign
 \string<\glshex 00A9% copyright sign
 \string<\glshex 00AE% registered sign
 \string<\string'\glshex 0040\string'% at sign
}
\newcommand*{\glsxtrcurrencyrules}{%
 \glshex 00A4% currency sign
 \string<\glshex 0E3F% Thai currency symbol baht
 \string<\glshex 00A2% cent sign
 \string<\glshex 20A1% colon sign
 \string<\glshex 20A2% cruzeiro sign
 \string<\string'\glshex 0024\string'% dollar sign
 \string<\glshex 20AB% dong sign
 \string<\glshex 20AC% euro sign
 \string<\glshex 20A3% French franc sign
 \string<\glshex 20A4% lira sign
 \string<\glshex 20A5% mill sign
 \string<\glshex 20A6% naira sign
```

```
 \string<\glshex 20A7% peseta sign
 \string<\glshex 00A3% pound sign
 \string<\glshex 20A8% rupee sign
 \string<\glshex 20AA% new sheqel sign
 \string<\glshex 20A9% won sign
 \string<\glshex 00A5% yen sign
}
\newcommand*{\glsxtrgeneralpuncIIrules}{%
 \string'\glshex 002A\string'% asterisk
 \string<\string'\glshex 005C\string'% backslash
 \string<\string'\glshex 0026\string'% ampersand
 \string<\string'\glshex 0023\string'% hash sign
 \string<\string'\glshex 0025\string'% percent sign
 \string<\string'\glshex 002B\string'% plus sign
  \string=\glshex 207A\string=\glshex 208A% super/subscript plus sign
 \string<\glshex 00B1% plus-minus sign
 \string<\glshex 00F7% division sign
 \string<\glshex 00D7% multiplication sign
 \string<\string'\glshex 003C\string'% less-than sign
 \string<\string'\glshex 003D\string'% equals sign
 \string<\string'\glshex 003E\string'% greater-than sign
 \string<\glshex 00AC% not sign
 \string<\string'\glshex 007C\string'% vertical bar (pipe)
 \string<\glshex 00A6% broken bar
 \string<\glshex 00B0% degree sign
 \string<\glshex 00B5% micron sign
}
\newcommand*{\glsxtrGeneralLatinIrules}{%
 \glsxtrLatinA
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinE
 \string<f,F%
 \string<g,G%
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string<\glsxtrLatinP
 \string<q,Q%
 \string<r,R%
 \string<\glsxtrLatinS
 \string<\glsxtrLatinT
 \string<u,U%
 \string<v,V%
```

```
 \string<w,W%
 \string<\glsxtrLatinX
 \string<y,Y%
 \string<z,Z
}
\newcommand*{\glsxtrGeneralLatinIIrules}{%
 \glsxtrLatinA
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinEth
 \string<\glsxtrLatinE
 \string<f,F%
 \string<g,G%
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string<\glsxtrLatinP
 \string<q,Q%
 \string<r,R%
 \string<\glsxtrLatinS
 \string& SS \string, \glsxtrLatinEszettSs
 \string<\glsxtrLatinT
 \string<u,U%
 \string<v,V%
 \string<w,W%
 \string<\glsxtrLatinX
 \string<y,Y%
 \string<z,Z%
}
\newcommand*{\glsxtrGeneralLatinIIIrules}{%
 \glsxtrLatinA
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinEth
 \string<\glsxtrLatinE
 \string<f,F%
 \string<g,G%
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
```

```
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string<\glsxtrLatinP
 \string<q,Q%
 \string<r,R%
 \string<\glsxtrLatinS
 \string& SZ, \glsxtrLatinEszettSz
 \string<\glsxtrLatinT
 \string<u,U%
 \string<v,V%
 \string<w,W%
 \string<\glsxtrLatinX
 \string<y,Y%
 \string<z,Z%
}
\newcommand*{\glsxtrGeneralLatinIVrules}{%
 \glsxtrLatinA
 \string& AE , \glsxtrLatinAELigature
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinEth
 \string<\glsxtrLatinE
 \string<f,F%
 \string<g,G%
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string& OE , \glsxtrLatinOELigature
 \string<\glsxtrLatinP
 \string<q,Q%
 \string<r,R%
 \string<\glsxtrLatinS
 \string& SS , \glsxtrLatinEszettSs
 \string<\glsxtrLatinT
 \string& th =\glshex 00DE
 \string& TH =\glshex 00FE
 \string<u,U%
 \string<v,V%
 \string<w,W%
 \string<\glsxtrLatinX
 \string<y,Y%
 \string<z,Z%
}
\newcommand*{\glsxtrGeneralLatinVrules}{%
```

```
 \glsxtrLatinA
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinEth
 \string<\glsxtrLatinE
 \string<f,F%
 \string<g,G%
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string<\glsxtrLatinP
 \string<q,Q%
 \string<r,R%
 \string<\glsxtrLatinS
 \string& SS , \glsxtrLatinEszettSs
 \string<\glsxtrLatinT
 \string& th =\glshex 00DE
 \string& TH =\glshex 00FE
 \string<u,U%
 \string<v,V%
 \string<w,W%
 \string<\glsxtrLatinX
 \string<y,Y%
 \string<z,Z%
}
\newcommand*{\glsxtrGeneralLatinVIrules}{%
 \glsxtrLatinA
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinEth
 \string<\glsxtrLatinE
 \string<f,F%
 \string<g,G%
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string<\glsxtrLatinP
 \string<q,Q%
```

```
 \string<r,R%
 \string<\glsxtrLatinS
 \string& SZ , \glsxtrLatinEszettSz
 \string<\glsxtrLatinT
 \string& th =\glshex 00DE
 \string& TH =\glshex 00FE
 \string<u,U%
 \string<v,V%
 \string<w,W%
 \string<\glsxtrLatinX
 \string<y,Y%
 \string<z,Z%
}
\newcommand*{\glsxtrGeneralLatinVIIrules}{%
 \glsxtrLatinA
 \string<\glsxtrLatinAELigature
 \string<b,B%
 \string<c,C%
 \string<d,D%
 \string<\glsxtrLatinEth
 \string<\glsxtrLatinE
 \string<f,F%
 \string<\glsxtrLatinInsularG
 \string<\glsxtrLatinH
 \string<\glsxtrLatinI
 \string<j,J%
 \string<\glsxtrLatinK
 \string<\glsxtrLatinL
 \string<\glsxtrLatinM
 \string<\glsxtrLatinN
 \string<\glsxtrLatinO
 \string<\glsxtrLatinOELigature
 \string<\glsxtrLatinP
 \string<q,Q%
 \string<r,R%
 \string<\glshex 017F=\glsxtrLatinS % s and long s
 \string<\glsxtrLatinT
 \string<\glsxtrLatinThorn
 \string<u,U%
 \string<v,V%
 \string< w\string=\glshex 01BF, W\string=\glshex 01F7
 \string<\glsxtrLatinX
 \string<y,Y%
 \string<z,Z%
}
\newcommand*{\glsxtrGeneralLatinVIIIrules}{%
 \glsxtrLatinA
 \string& AE , \glsxtrLatinAELigature
 \string<b,B%
 \string<c,C%
```

```
\string<\glshex 00F0\string;d,\glshex 00D0\string;D% D and eth
\string<\glsxtrLatinE
\string<f,F%
\string<g,G%
\string<\glsxtrLatinH
\string<\glsxtrLatinI
\string<j,J%
\string<\glsxtrLatinK
\string<\glshex 0142\string=\glsxtrLatinL\string=\glshex 0141% L and \L
\string<\glsxtrLatinM
\string<\glsxtrLatinN
\string<\glshex 00F8\string=\glsxtrLatinO\string=\glshex 00D8% O and \O
\string& OE , \glsxtrLatinOELigature
\string<\glsxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glsxtrLatinS
\string& SS , \glsxtrLatinEszettSs
\string<\glsxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glsxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glsxtrLatinA}{%
  a\string=\glshex 00AA\string=\glshex 2090,A
}
\newcommand*{\glsxtrLatinE}{%
  e\string=\glshex 2091,E
}
\newcommand*{\glsxtrLatinH}{%
  h\string=\glshex 2095,H
}
\newcommand*{\glsxtrLatinI}{%
  i\string=\glshex 2071,I
}
\newcommand*{\glsxtrLatinK}{%
  k\string=\glshex 2096,K
}
\newcommand*{\glsxtrLatinL}{%
  l\string=\glshex 2097,L
}
\newcommand*{\glsxtrLatinM}{%
  m\string=\glshex 2098,M
}
\newcommand*{\glsxtrLatinN}{%
```

```
   n\string=\glshex 207F\string=\glshex 2099,N
}
\newcommand*{\glsxtrLatinO}{%
  o\string=\glshex 00BA\string=\glshex 2092,O
}
\newcommand*{\glsxtrLatinP}{%
  p\string=\glshex 209A,P
}
\newcommand*{\glsxtrLatinS}{%
  s\string=\glshex 209B,S
}
\newcommand*{\glsxtrLatinT}{%
  t\string=\glshex 209C,T
}
\newcommand*{\glsxtrLatinX}{%
  x\string=\glshex 2093,X
}
\newcommand*{\glsxtrLatinSchwa}{%
  \glshex 0259\string=\glshex 2094,\glshex 018F
}
\newcommand*{\glsxtrLatinEszettSs}{%
 \glshex 00DF% eszett
 \string=\glshex 017Fs % long S s
}
\newcommand*{\glsxtrLatinEszettSz}{%
 \glshex 00DF% eszett
 \string= \glshex 017Fz % long S z
}
\newcommand*{\glsxtrLatinEth}{%
 \glshex 00F0,\glshex 00D0% eth
}
\newcommand*{\glsxtrLatinThorn}{%
 \glshex 00FE,\glshex 00DE% thorn
}
\newcommand*{\glsxtrLatinAELigature}{%
 \glshex 00E6,\glshex 00C6% AE-ligature
}
\newcommand*{\glsxtrLatinOELigature}{%
 \glshex 0153,\glshex 0152% OE-ligature
}
\newcommand*{\glsxtrLatinAA}{%
 \glshex 00E5=a\glshex 030A,% \aa
 \glshex 00C5=A\glshex 030A% \AA
}
\newcommand*{\glsxtrLatinWynn}{%
 \glshex 01BF,\glshex 01F7% wynn
}
\newcommand*{\glsxtrLatinInsularG}{%
 \glshex 1D79,\glshex A77D% insular G
 \string; g, G
```

```
}
\newcommand*{\glsxtrLatinOslash}{%
 \glshex 00F8,\glshex 00D8% \o, \O
}
\newcommand*{\glsxtrLatinLslash}{%
 \glshex 0142,\glshex 0141% \l, \L
}
\newcommand*{\glsxtrMathUpGreekIrules}{%
 \glsxtrUpAlpha
 \string<\glsxtrUpBeta
 \string<\glsxtrUpGamma
 \string<\glsxtrUpDelta
 \string<\glsxtrUpEpsilon
 \string<\glsxtrUpDigamma
 \string<\glsxtrUpZeta
 \string<\glsxtrUpEta
 \string<\glsxtrUpTheta
 \string<\glsxtrUpIota
 \string<\glsxtrUpKappa
 \string<\glsxtrUpLambda
 \string<\glsxtrUpMu
 \string<\glsxtrUpNu
 \string<\glsxtrUpXi
 \string<\glsxtrUpOmicron
 \string<\glsxtrUpPi
 \string<\glsxtrUpRho
 \string<\glsxtrUpSigma
 \string<\glsxtrUpTau
 \string<\glsxtrUpUpsilon
 \string<\glsxtrUpPhi
 \string<\glsxtrUpChi
 \string<\glsxtrUpPsi
 \string<\glsxtrUpOmega
}
\newcommand*{\glsxtrMathUpGreekIIrules}{%
 \glsxtrUpAlpha
 \string<\glsxtrUpBeta
 \string<\glsxtrUpGamma
 \string<\glsxtrUpDelta
 \string<\glsxtrUpEpsilon
 \string<\glsxtrUpZeta
 \string<\glsxtrUpEta
 \string<\glsxtrUpTheta
 \string<\glsxtrUpIota
 \string<\glsxtrUpKappa
 \string<\glsxtrUpLambda
 \string<\glsxtrUpMu
 \string<\glsxtrUpNu
 \string<\glsxtrUpXi
 \string<\glsxtrUpOmicron
```

```
 \string<\glsxtrUpPi
 \string<\glsxtrUpRho
 \string<\glsxtrUpSigma
 \string<\glsxtrUpTau
 \string<\glsxtrUpUpsilon
 \string<\glsxtrUpPhi
 \string<\glsxtrUpChi
 \string<\glsxtrUpPsi
 \string<\glsxtrUpOmega
}
\newcommand*{\glsxtrMathItalicGreekIrules}{%
 \glsxtrMathItalicAlpha
 \string<\glsxtrMathItalicBeta
 \string<\glsxtrMathItalicGamma
 \string<\glsxtrMathItalicDelta
 \string<\glsxtrMathItalicEpsilon
 \string<\glsxtrUpDigamma
 \string<\glsxtrMathItalicZeta
 \string<\glsxtrMathItalicEta
 \string<\glsxtrMathItalicTheta
 \string<\glsxtrMathItalicIota
 \string<\glsxtrMathItalicKappa
 \string<\glsxtrMathItalicLambda
 \string<\glsxtrMathItalicMu
 \string<\glsxtrMathItalicNu
 \string<\glsxtrMathItalicXi
 \string<\glsxtrMathItalicOmicron
 \string<\glsxtrMathItalicPi
 \string<\glsxtrMathItalicRho
 \string<\glsxtrMathItalicSigma
 \string<\glsxtrMathItalicTau
 \string<\glsxtrMathItalicUpsilon
 \string<\glsxtrMathItalicPhi
 \string<\glsxtrMathItalicChi
 \string<\glsxtrMathItalicPsi
 \string<\glsxtrMathItalicOmega
}
\newcommand*{\glsxtrMathItalicGreekIIrules}{%
 \glsxtrMathItalicAlpha
 \string<\glsxtrMathItalicBeta
 \string<\glsxtrMathItalicGamma
 \string<\glsxtrMathItalicDelta
 \string<\glsxtrMathItalicEpsilon
 \string<\glsxtrMathItalicZeta
 \string<\glsxtrMathItalicEta
 \string<\glsxtrMathItalicTheta
 \string<\glsxtrMathItalicIota
 \string<\glsxtrMathItalicKappa
 \string<\glsxtrMathItalicLambda
 \string<\glsxtrMathItalicMu
```

```
 \string<\glsxtrMathItalicNu
 \string<\glsxtrMathItalicXi
 \string<\glsxtrMathItalicOmicron
 \string<\glsxtrMathItalicPi
 \string<\glsxtrMathItalicRho
 \string<\glsxtrMathItalicSigma
 \string<\glsxtrMathItalicTau
 \string<\glsxtrMathItalicUpsilon
 \string<\glsxtrMathItalicPhi
 \string<\glsxtrMathItalicChi
 \string<\glsxtrMathItalicPsi
 \string<\glsxtrMathItalicOmega
}
\newcommand*{\glsxtrMathItalicUpperGreekIrules}{%
 \glshex 1D6E2% upper case alpha (maths italic)
 \string<\glshex 1D6E3% upper case beta (maths italic)
 \string<\glshex 1D6E4% upper case gamma (maths italic)
 \string<\glshex 1D6E5% upper case delta (maths italic)
 \string<\glshex 1D6E6% upper case epsilon (maths italic)
 \string<\glshex 03DC% upper case digamma
 \string<\glshex 1D6E7% upper case zeta (maths italic)
 \string<\glshex 1D6E8% upper case eta (maths italic)
 \string<\glshex 1D6E9% upper case theta (maths italic)
 \string=\glshex 1D6F3% upper case theta variant (maths italic)
 \string<\glshex 1D6EA% upper case iota (maths italic)
 \string<\glshex 1D6EB% upper case kappa (maths italic)
 \string<\glshex 1D6EC% upper case lambda (maths italic)
 \string<\glshex 1D6ED% upper case mu (maths italic)
 \string<\glshex 1D6EE% upper case nu (maths italic)
 \string<\glshex 1D6EF% upper case xi (maths italic)
 \string<\glshex 1D6F0% upper case omicron (maths italic)
 \string<\glshex 1D6F1% upper case pi (maths italic)
 \string<\glshex 1D6F2% upper case rho (maths italic)
 \string<\glshex 1D6F4% upper case sigma (maths italic)
 \string<\glshex 1D6F5% upper case tau (maths italic)
 \string<\glshex 1D6F6% upper case upsilon (maths italic)
 \string<\glshex 1D6F7% upper case phi (maths italic)
 \string<\glshex 1D6F8% upper case chi (maths italic)
 \string<\glshex 1D6F9% upper case psi (maths italic)
 \string<\glshex 1D6FA% upper case omega (maths italic)
}
\newcommand*{\glsxtrMathItalicUpperGreekIIrules}{%
 \glshex 1D6E2% upper case alpha (maths italic)
 \string<\glshex 1D6E3% upper case beta (maths italic)
 \string<\glshex 1D6E4% upper case gamma (maths italic)
 \string<\glshex 1D6E5% upper case delta (maths italic)
 \string<\glshex 1D6E6% upper case epsilon (maths italic)
 \string<\glshex 1D6E7% upper case zeta (maths italic)
 \string<\glshex 1D6E8% upper case eta (maths italic)
 \string<\glshex 1D6E9% upper case theta (maths italic)
```

```
 \string=\glshex 1D6F3% upper case theta variant (maths italic)
 \string<\glshex 1D6EA% upper case iota (maths italic)
 \string<\glshex 1D6EB% upper case kappa (maths italic)
 \string<\glshex 1D6EC% upper case lambda (maths italic)
 \string<\glshex 1D6ED% upper case mu (maths italic)
 \string<\glshex 1D6EE% upper case nu (maths italic)
 \string<\glshex 1D6EF% upper case xi (maths italic)
 \string<\glshex 1D6F0% upper case omicron (maths italic)
 \string<\glshex 1D6F1% upper case pi (maths italic)
 \string<\glshex 1D6F2% upper case rho (maths italic)
 \string<\glshex 1D6F4% upper case sigma (maths italic)
 \string<\glshex 1D6F5% upper case tau (maths italic)
 \string<\glshex 1D6F6% upper case upsilon (maths italic)
 \string<\glshex 1D6F7% upper case phi (maths italic)
 \string<\glshex 1D6F8% upper case chi (maths italic)
 \string<\glshex 1D6F9% upper case psi (maths italic)
 \string<\glshex 1D6FA% upper case omega (maths italic)
}
\newcommand*{\glsxtrMathItalicLowerGreekIrules}{%
 \glshex 1D6FC% lower case alpha (maths italic)
 \string<\glshex 1D6FD% lower case beta (maths italic)
 \string<\glshex 1D6FE% lower case gamma (maths italic)
 \string<\glshex 1D6FF% lower case delta (maths italic)
 \string<\glshex 1D700% lower case epsilon (maths italic)
 \string=\glshex 1D716% lower case epsilon variant (maths italic)
 \string<\glshex 03DD% lower case digamma
 \string<\glshex 1D701% lower case zeta (maths italic)
 \string<\glshex 1D702% lower case eta (maths italic)
 \string<\glshex 1D703% lower case theta (maths italic)
 \string=\glshex 1D717% lower case theta variant (maths italic)
 \string<\glshex 1D704% lower case iota (maths italic)
 \string<\glshex 1D705% lower case kappa (maths italic)
 \string=\glshex 1D718% lower case kappa variant (maths italic)
 \string<\glshex 1D706% lower case lambda (maths italic)
 \string<\glshex 1D707% lower case mu (maths italic)
 \string<\glshex 1D708% lower case nu (maths italic)
 \string<\glshex 1D709% lower case xi (maths italic)
 \string<\glshex 1D70A% lower case omicron (maths italic)
 \string<\glshex 1D70B% lower case pi (maths italic)
 \string=\glshex 1D71B% lower case pi variant (maths italic)
 \string<\glshex 1D70C% lower case rho (maths italic)
 \string=\glshex 1D71A% lower case rho variant (maths italic)
 \string<\glshex 1D70D% lower case final sigma (maths italic)
 \string=\glshex 1D70E% lower case sigma (maths italic)
 \string<\glshex 1D70F% lower case tau (maths italic)
 \string<\glshex 1D710% lower case upsilon (maths italic)
 \string<\glshex 1D711% lower case phi (maths italic)
 \string=\glshex 1D719% lower case phi variant (maths italic)
 \string<\glshex 1D712% lower case chi (maths italic)
 \string<\glshex 1D713% lower case psi (maths italic)
```

```
    \string<\glshex 1D714% lower case omega (maths italic)
}
\newcommand*{\glsxtrMathItalicLowerGreekIIrules}{%
 \glshex 1D6FC% lower case alpha (maths italic)
 \string<\glshex 1D6FD% lower case beta (maths italic)
 \string<\glshex 1D6FE% lower case gamma (maths italic)
 \string<\glshex 1D6FF% lower case delta (maths italic)
 \string<\glshex 1D700% lower case epsilon (maths italic)
 \string=\glshex 1D716% lower case epsilon variant (maths italic)
 \string<\glshex 1D701% lower case zeta (maths italic)
 \string<\glshex 1D702% lower case eta (maths italic)
 \string<\glshex 1D703% lower case theta (maths italic)
 \string=\glshex 1D717% lower case theta variant (maths italic)
 \string<\glshex 1D704% lower case iota (maths italic)
 \string<\glshex 1D705% lower case kappa (maths italic)
 \string=\glshex 1D718% lower case kappa variant (maths italic)
 \string<\glshex 1D706% lower case lambda (maths italic)
 \string<\glshex 1D707% lower case mu (maths italic)
 \string<\glshex 1D708% lower case nu (maths italic)
 \string<\glshex 1D709% lower case xi (maths italic)
 \string<\glshex 1D70A% lower case omicron (maths italic)
 \string<\glshex 1D70B% lower case pi (maths italic)
 \string=\glshex 1D71B% lower case pi variant (maths italic)
 \string<\glshex 1D70C% lower case rho (maths italic)
 \string=\glshex 1D71A% lower case rho variant (maths italic)
 \string<\glshex 1D70D% lower case final sigma (maths italic)
 \string=\glshex 1D70E% lower case sigma (maths italic)
 \string<\glshex 1D70F% lower case tau (maths italic)
 \string<\glshex 1D710% lower case upsilon (maths italic)
 \string<\glshex 1D711% lower case phi (maths italic)
 \string=\glshex 1D719% lower case phi variant (maths italic)
 \string<\glshex 1D712% lower case chi (maths italic)
 \string<\glshex 1D713% lower case psi (maths italic)
 \string<\glshex 1D714% lower case omega (maths italic)
}
\newcommand*{\glsxtrMathGreekIrules}{%
 \glsxtrMathItalicAlpha
 \string;\glsxtrUpAlpha
 \string<\glsxtrMathItalicBeta
 \string;\glsxtrUpBeta
 \string<\glsxtrMathItalicGamma
 \string;\glsxtrUpGamma
 \string<\glsxtrMathItalicDelta
 \string;\glsxtrUpDelta
 \string<\glsxtrMathItalicEpsilon
 \string;\glsxtrUpEpsilon
 \string<\glsxtrUpDigamma
 \string<\glsxtrMathItalicZeta
 \string;\glsxtrUpZeta
 \string<\glsxtrMathItalicEta
```

```
 \string;\glsxtrUpEta
 \string<\glsxtrMathItalicTheta
 \string;\glsxtrUpTheta
 \string<\glsxtrMathItalicIota
 \string;\glsxtrUpIota
 \string<\glsxtrMathItalicKappa
 \string;\glsxtrUpKappa
 \string<\glsxtrMathItalicLambda
 \string;\glsxtrUpLambda
 \string<\glsxtrMathItalicMu
 \string;\glsxtrUpMu
 \string<\glsxtrMathItalicNu
 \string;\glsxtrUpNu
 \string<\glsxtrMathItalicXi
 \string;\glsxtrUpXi
 \string<\glsxtrMathItalicOmicron
 \string;\glsxtrUpOmicron
 \string<\glsxtrMathItalicPi
 \string;\glsxtrUpPi
 \string<\glsxtrMathItalicRho
 \string;\glsxtrUpRho
 \string<\glsxtrMathItalicSigma
 \string;\glsxtrUpSigma
 \string<\glsxtrMathItalicTau
 \string;\glsxtrUpTau
 \string<\glsxtrMathItalicUpsilon
 \string;\glsxtrUpUpsilon
 \string<\glsxtrMathItalicPhi
 \string;\glsxtrUpPhi
 \string<\glsxtrMathItalicChi
 \string;\glsxtrUpChi
 \string<\glsxtrMathItalicPsi
 \string;\glsxtrUpPsi
 \string<\glsxtrMathItalicOmega
 \string;\glsxtrUpOmega
}
\newcommand*{\glsxtrMathGreekIIrules}{%
 \glsxtrMathItalicAlpha
 \string;\glsxtrUpAlpha
 \string<\glsxtrMathItalicBeta
 \string;\glsxtrUpBeta
 \string<\glsxtrMathItalicGamma
 \string;\glsxtrUpGamma
 \string<\glsxtrMathItalicDelta
 \string;\glsxtrUpDelta
 \string<\glsxtrMathItalicEpsilon
 \string;\glsxtrUpEpsilon
 \string<\glsxtrMathItalicZeta
 \string;\glsxtrUpZeta
 \string<\glsxtrMathItalicEta
```

```
 \string;\glsxtrUpEta
 \string<\glsxtrMathItalicTheta
 \string;\glsxtrUpTheta
 \string<\glsxtrMathItalicIota
 \string;\glsxtrUpIota
 \string<\glsxtrMathItalicKappa
 \string;\glsxtrUpKappa
 \string<\glsxtrMathItalicLambda
 \string;\glsxtrUpLambda
 \string<\glsxtrMathItalicMu
 \string;\glsxtrUpMu
 \string<\glsxtrMathItalicNu
 \string;\glsxtrUpNu
 \string<\glsxtrMathItalicXi
 \string;\glsxtrUpXi
 \string<\glsxtrMathItalicOmicron
 \string;\glsxtrUpOmicron
 \string<\glsxtrMathItalicPi
 \string;\glsxtrUpPi
 \string<\glsxtrMathItalicRho
 \string;\glsxtrUpRho
 \string<\glsxtrMathItalicSigma
 \string;\glsxtrUpSigma
 \string<\glsxtrMathItalicTau
 \string;\glsxtrUpTau
 \string<\glsxtrMathItalicUpsilon
 \string;\glsxtrUpUpsilon
 \string<\glsxtrMathItalicPhi
 \string;\glsxtrUpPhi
 \string<\glsxtrMathItalicChi
 \string;\glsxtrUpChi
 \string<\glsxtrMathItalicPsi
 \string;\glsxtrUpPsi
 \string<\glsxtrMathItalicOmega
 \string;\glsxtrUpOmega
}
\newcommand*{\glsxtrUpAlpha}{%
 \glshex 03B1,% lower case alpha
 \glshex 0391% upper case alpha
}
\newcommand*{\glsxtrUpBeta}{%
 \glshex 03B2,% lower case beta
 \glshex 0392% upper case beta
}
\newcommand*{\glsxtrUpGamma}{%
 \glshex 03B3,% lower case gamma
 \glshex 0393% upper case gamma
}
\newcommand*{\glsxtrUpDelta}{%
 \glshex 03B4,% lower case delta
```

```
\glshex 0394% upper case delta
}
\newcommand*{\glsxtrUpEpsilon}{%
 \glshex 03B5% lower case epsilon
 \string=\glshex 03F5,% lower case epsilon variant
 \glshex 0395% upper case epsilon
}
\newcommand*{\glsxtrUpDigamma}{%
 \glshex 03DD,% lower case digamma
 \glshex 03DC% upper case digamma
}
\newcommand*{\glsxtrUpZeta}{%
 \glshex 03B6,% lower case zeta
 \glshex 0396% upper case zeta
}
\newcommand*{\glsxtrUpEta}{%
 \glshex 03B7,% lower case eta
 \glshex 0397% upper case eta
}
\newcommand*{\glsxtrUpTheta}{%
 \glshex 03B8% lower case theta
 \string=\glshex 03D1,% lower case theta variant
 \glshex 0398% upper case theta
}
\newcommand*{\glsxtrUpIota}{%
 \glshex 03B9,% lower case iota
 \glshex 0399% upper case iota
}
\newcommand*{\glsxtrUpKappa}{%
 \glshex 03BA% lower case kappa
 \string=\glshex 03F0,% lower case kappa variant
 \glshex 039A% upper case kappa
}
\newcommand*{\glsxtrUpLambda}{%
 \glshex 03BB,% lower lambda
 \glshex 039B% upper case lambda
}
\newcommand*{\glsxtrUpMu}{%
 \glshex 03BC,% lower case mu
 \glshex 039C% upper case mu
}
\newcommand*{\glsxtrUpNu}{%
 \glshex 03BD,% lower case nu
 \glshex 039D% upper case nu
}
\newcommand*{\glsxtrUpXi}{%
 \glshex 03BE,% lower case xi
 \glshex 039E% upper case xi
}
\newcommand*{\glsxtrUpOmicron}{%
```

```
\glshex 03BF,% lower case omicron
\glshex 039F% upper case omicron
}
\newcommand*{\glsxtrUpPi}{%
\glshex 03C0% lower case pi
\string=\glshex 03D6,% lower case pi variant
\glshex 03A0% upper case pi
}
\newcommand*{\glsxtrUpRho}{%
\glshex 03C1% lower case rho
\string=\glshex 03F1,% lower case rho variant
\glshex 03A1% upper case rho
}
\newcommand*{\glsxtrUpSigma}{%
\glshex 03C2% lower case sigma
\string=\glshex 03C3,% lower case sigma
\glshex 03A3% upper case sigma
}
\newcommand*{\glsxtrUpTau}{%
\glshex 03C4,% lower case tau
\glshex 03A4% upper case tau
}
\newcommand*{\glsxtrUpUpsilon}{%
\glshex 03C5,% lower case upsilon
\glshex 03A5% upper case upsilon
}
\newcommand*{\glsxtrUpPhi}{%
\glshex 03C6% lower case phi
\string=\glshex 03D5,% lower case phi variant
\glshex 03A6% upper case phi
}
\newcommand*{\glsxtrUpChi}{%
\glshex 03C7,% lower case chi
\glshex 03A7% upper case chi
}
\newcommand*{\glsxtrUpPsi}{%
\glshex 03C8,% lower case psi
\glshex 03A8% upper case psi
}
\newcommand*{\glsxtrUpOmega}{%
\glshex 03C9,% lower case omega
\glshex 03A9% upper case omega
}
\newcommand*{\glsxtrMathItalicAlpha}{%
\glshex 1D6FC,% lower case alpha (maths italic)
\glshex 1D6E2% upper case alpha (maths italic)
}
\newcommand*{\glsxtrMathItalicBeta}{%
\glshex 1D6FD,% lower case beta (maths italic)
\glshex 1D6E3% upper case beta (maths italic)
```

```
}
\newcommand*{\glsxtrMathItalicGamma}{%
 \glshex 1D6FE,% lower case gamma (maths italic)
 \glshex 1D6E4% upper case gamma (maths italic)
}
\newcommand*{\glsxtrMathItalicDelta}{%
 \glshex 1D6FF,% lower case delta (maths italic)
 \glshex 1D6E5% upper case delta (maths italic)
}
\newcommand*{\glsxtrMathItalicEpsilon}{%
 \glshex 1D700% lower case epsilon (maths italic)
 \string=\glshex 1D716,% lower case epsilon variant (maths italic)
 \glshex 1D6E6% upper case epsilon (maths italic)
}
\newcommand*{\glsxtrMathItalicZeta}{%
 \glshex 1D701,% lower case zeta (maths italic)
 \glshex 1D6E7% upper case zeta (maths italic)
}
\newcommand*{\glsxtrMathItalicEta}{%
 \glshex 1D702,% lower case eta (maths italic)
 \glshex 1D6E8% upper case eta (maths italic)
}
\newcommand*{\glsxtrMathItalicTheta}{%
 \glshex 1D703% lower case theta (maths italic)
 \string=\glshex 1D717,% lower case theta variant (maths italic)
 \glshex 1D6E9% upper case theta (maths italic)
 \string=\glshex 1D6F3% upper case theta variant (maths italic)
}
\newcommand*{\glsxtrMathItalicIota}{%
 \glshex 1D704,% lower case iota (maths italic)
 \glshex 1D6EA% upper case iota (maths italic)
}
\newcommand*{\glsxtrMathItalicKappa}{%
 \glshex 1D705% lower case kappa (maths italic)
 \string=\glshex 1D718,% lower case kappa variant (maths italic)
 \glshex 1D6EB% upper case kappa (maths italic)
}
\newcommand*{\glsxtrMathItalicLambda}{%
 \glshex 1D706,% lower case lambda (maths italic)
 \glshex 1D6EC% upper case lambda (maths italic)
}
\newcommand*{\glsxtrMathItalicMu}{%
 \glshex 1D707,% lower case mu (maths italic)
 \glshex 1D6ED% upper case mu (maths italic)
}
\newcommand*{\glsxtrMathItalicNu}{%
 \glshex 1D708,% lower case nu (maths italic)
 \glshex 1D6EE% upper case nu (maths italic)
}
\newcommand*{\glsxtrMathItalicXi}{%
```

```
 \glshex 1D709,% lower case xi (maths italic)
 \glshex 1D6EF% upper case xi (maths italic)
}
\newcommand*{\glsxtrMathItalicOmicron}{%
 \glshex 1D70A,% lower case omicron (maths italic)
 \glshex 1D6F0% upper case omicron (maths italic)
}
\newcommand*{\glsxtrMathItalicPi}{%
 \glshex 1D70B% lower case pi (maths italic)
 \string=\glshex 1D71B,% lower case pi variant (maths italic)
 \glshex 1D6F1% upper case pi (maths italic)
}
\newcommand*{\glsxtrMathItalicRho}{%
 \glshex 1D70C% lower case rho (maths italic)
 \string=\glshex 1D71A,% lower case rho variant (maths italic)
 \glshex 1D6F2% upper case rho (maths italic)
}
\newcommand*{\glsxtrMathItalicSigma}{%
 \glshex 1D70D% lower case final sigma (maths italic)
 \string=\glshex 1D70E,% lower case sigma (maths italic)
 \glshex 1D6F4% upper case sigma (maths italic)
}
\newcommand*{\glsxtrMathItalicTau}{%
 \glshex 1D70F,% lower case tau (maths italic)
 \glshex 1D6F5% upper case tau (maths italic)
}
\newcommand*{\glsxtrMathItalicUpsilon}{%
 \glshex 1D710,% lower case upsilon (maths italic)
 \glshex 1D6F6% upper case upsilon (maths italic)
}
\newcommand*{\glsxtrMathItalicPhi}{%
 \glshex 1D711% lower case phi (maths italic)
 \string=\glshex 1D719,% lower case phi variant (maths italic)
 \glshex 1D6F7% upper case phi (maths italic)
}
\newcommand*{\glsxtrMathItalicChi}{%
 \glshex 1D712,% lower case chi (maths italic)
 \glshex 1D6F8% upper case chi (maths italic)
}
\newcommand*{\glsxtrMathItalicPsi}{%
 \glshex 1D713,% lower case psi (maths italic)
 \glshex 1D6F9% upper case psi (maths italic)
}
\newcommand*{\glsxtrMathItalicOmega}{%
 \glshex 1D714,% lower case omega (maths italic)
 \glshex 1D6FA% upper case omega (maths italic)
}
\newcommand*{\glsxtrMathItalicPartial}{%
 \glshex 1D715% partial differential (maths italic)
}
```

```
\newcommand*{\glsxtrMathItalicNabla}{%
 \glshex 1D6FB% nabla (maths italic)
}
\newcommand*{\glsxtrdigitrules}{%
 0\string=\glshex 2080\string=\glshex 2070
 \string<1\string=\glshex 2081\string=\glshex 00B9
 \string<2\string=\glshex 2082\string=\glshex 00B2
 \string<3\string=\glshex 2083\string=\glshex 00B3
 \string<4\string=\glshex 2084\string=\glshex 2074
 \string<5\string=\glshex 2085\string=\glshex 2075
 \string<6\string=\glshex 2086\string=\glshex 2076
 \string<7\string=\glshex 2087\string=\glshex 2077
 \string<8\string=\glshex 2088\string=\glshex 2078
 \string<9\string=\glshex 2089\string=\glshex 2079
}
\newcommand*{\glsxtrBasicDigitrules}{%
 0\string<1\string<2\string<3\string<4%
 \string<5\string<6\string<7\string<8\string<9%
}
\newcommand*{\glsxtrSubScriptDigitrules}{%
 \glshex 2080% subscript 0
 \string<\glshex 2081% subscript 1
 \string<\glshex 2082% subscript 2
 \string<\glshex 2083% subscript 3
 \string<\glshex 2084% subscript 4
 \string<\glshex 2085% subscript 5
 \string<\glshex 2086% subscript 6
 \string<\glshex 2087% subscript 7
 \string<\glshex 2088% subscript 8
 \string<\glshex 2089% subscript 9
}
\newcommand*{\glsxtrSuperScriptDigitrules}{%
 \glshex 2070% superscript 0
 \string<\glshex 00B9% superscript 1
 \string<\glshex 00B2% superscript 2
 \string<\glshex 00B3% superscript 3
 \string<\glshex 2074% superscript 4
 \string<\glshex 2075% superscript 5
 \string<\glshex 2076% superscript 6
 \string<\glshex 2077% superscript 7
 \string<\glshex 2078% superscript 8
 \string<\glshex 2079% superscript 9
}
\newcommand*{\glsxtrfractionrules}{%
 \glshex 215F% fraction numerator one (1/)
 \string<\glshex 2189% zero thirds (0/3 = 0)
 \string<\glshex 2152% one tenth (1/10 = 0.1)
 \string<\glshex 2151% one ninth (1/9 ~ 0.111)
 \string<\glshex 215B% one eighth (1/8 = 0.125)
 \string<\glshex 2150% one seventh (1/7 ~ 0.143)
```

```
 \string<\glshex 2159% one sixth (1/6 ~ 0.167)
 \string<\glshex 2155% one fifth (1/5 = 0.2)
 \string<\glshex 00BC% one quarter (1/4 = 0.25)
 \string<\glshex 2153% one third (1/3 ~ 0.333)
 \string<\glshex 215C% three eighths (3/8 = 0.375)
 \string<\glshex 2156% two fifths (2/5 = 0.4)
 \string<\glshex 00BD% one half (1/2 = 0.5)
 \string<\glshex 2157% three fifths (3/5 = 0.6)
 \string<\glshex 215D% five eighths (5/8 = 0.625)
 \string<\glshex 2154% two thirds (2/3 ~ 0.667)
 \string<\glshex 00BE% three quarters (3/4 = 0.75)
 \string<\glshex 2158% four fifths (4/5 = 0.8)
 \string<\glshex 215A% five sixths (5/6 ~ 0.833)
 \string<\glshex 215E% seven eighths (7/8 = 0.875)
}
\renewcommand{\@glsxtrdialecthook}{%
  \ifundef\CurrentTrackedScript
  {%
    \TrackLangIfHasDefaultScript{\CurrentTrackedLanguage}%
    {%
     \edef\CurrentTrackedScript{%
        \TrackLangGetDefaultScript\CurrentTrackedLanguage}%
    }%
    {}%
  }%
  {}%
  \ifdef\CurrentTrackedScript
  {%
    \let\gls@orgTrackLangRequireDialectPrefix\TrackLangRequireDialectPrefix
    \def\TrackLangRequireDialectPrefix{glossariesxtr-}%
    \let\CurrentTrackedTag\CurrentTrackedScript
    \IfFileExists{\TrackLangRequireDialectPrefix\CurrentTrackedTag.ldf}
    {\RequireGlossariesExtraLang{\CurrentTrackedTag}}%
    {}%
    \let\TrackLangRequireDialectPrefix\gls@orgTrackLangRequireDialectPrefix
  }%
  {}%
}
\ifdef\glsxtr@loaddialect
{%
  \@ifpackageloaded{tracklang}
  {%
    \AnyTrackedLanguages
    {%
      \ForEachTrackedDialect{\this@dialect}{\glsxtr@loaddialect}%
    }%
    {}%
  }
  {}
}
```

```
  {}
```

9.3   Rollback v1.48 (glossaries-extra-stylemods-2021-11-22.sty)

Version 1.48 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-extra-stylemods}[2021/11/22 v1.48 (NLCT)]
\newcommand*{\@glsxtr@loadstyles}{}
\DeclareOption{all}{%
  \appto\@glsxtr@loadstyles{%
    \RequirePackage{glossary-inline}%
    \RequirePackage{glossary-list}%
    \RequirePackage{glossary-tree}%
    \RequirePackage{glossary-mcols}%
    \RequirePackage{glossary-long}%
    \RequirePackage{glossary-longragged}%
    \RequirePackage{glossary-longbooktabs}%
    \RequirePackage{glossary-super}%
    \RequirePackage{glossary-superragged}%
    \RequirePackage{glossary-bookindex}[=v1.48]%
    \RequirePackage{glossary-longextra}[=v1.48]%
    \RequirePackage{glossary-topic}[=v1.48]%
  }
}
\DeclareOption*{%
  \IfFileExists{glossary-\CurrentOption.sty}
   {\eappto\@glsxtr@loadstyles{%
      \noexpand\RequirePackage{glossary-\CurrentOption}}%
   }%
   {%
     \PackageError{glossaries-extra-styles}%
     {Unknown option '\CurrentOption'}{}%
   }%
}
\ProcessOptions
\@glsxtr@loadstyles
\providecommand*{\glsxtrprelocation}{\space}
\providecommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries-extra}{Glossary style '#1' isn't already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\ifdef{\@glsstyle@listdotted}
{%
 \renewglossarystyle{listdotted}{%
```

1156

```
      \setglossarystyle{list}%
      \renewcommand*{\glossentry}[2]{%
       \item[]\makebox[\glslistdottedwidth][l]{%
          \glsentryitem{##1}%
          \glstarget{##1}{\glossentryname{##1}}%
          \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}%
          \glossentrydesc{##1}\glspostdescription}%
      \renewcommand*{\subglossentry}[3]{%
       \item[]\makebox[\glslistdottedwidth][l]{%
       \glssubentryitem{##2}%
       \glstarget{##2}{\glossentryname{##2}}%
       \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}%
       \glossentrydesc{##2}\glspostdescription}%
 }
}
{%
}
\ifdef{\@glsstyle@list}
{%
  \newcommand{\glslistprelocation}{\glsxtrprelocation}
  \newcommand{\glslistchildprelocation}{\glslistprelocation}
  \newcommand{\glslistchildpostlocation}{.}
  \newcommand{\glslistdesc}[1]{\glossentrydesc{#1}\glspostdescription}
  \newcommand{\glslistgroupskip}{\nobreak\indexspace\nobreak}
  \newcommand{\glslistitem}[1]{%
    \item[\glsentryitem{#1}%
          \glstarget{#1}{\glossentryname{#1}}]%
  }
  \providecommand{\glslistinit}{%
    \ifdef\GetTitleStringDisableCommands
    {%
      \GetTitleStringSetup{expand}%
      \GetTitleStringDisableCommands{%
        \let\glsentryitem\@gobble
        \let\glstarget\@secondoftwo
        \let\glossentryname\glslistexpandedname
        \let\glslistgroupheaderfmt\@firstofone
        \let\glsgetgrouptitle\@firstofone
        \let\glsnavhypertarget\@secondoftwo
        \let\glsnavigation\relax
      }%
    }%
    {}%
  }
  \providecommand{\glslistexpandedname}[1]{%
    \ifcsname glo@\glsdetoklabel{#1}@name\endcsname
     \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\expandafter\endcsname
    \fi
  }
  \renewglossarystyle{list}{%
```

1157

```
    \renewenvironment{theglossary}%
      {\glslistinit\begin{description}}{\end{description}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand*{\glossentry}[2]{%
      \glslistitem{##1}\glslistdesc{##1}\glslistprelocation ##2}%
    \renewcommand*{\subglossentry}[3]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\space
      \glslistdesc{##2}%
      \glslistchildprelocation ##3\glslistchildpostlocation}%
    \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glslistgroupskip\fi}%
  }
}
{}
\ifdef{\@glsstyle@altlist}
{%
  \newcommand{\glsaltlistitem}[1]{%
    \glslistitem{#1}%
    \mbox{}\par\nobreak\@afterheading
  }
  \renewglossarystyle{altlist}{%
    \setglossarystyle{list}%
    \renewcommand*{\glossentry}[2]{%
      \glsaltlistitem{##1}%
      \glslistdesc{##1}\glslistprelocation ##2}%
    \renewcommand{\subglossentry}[3]{%
      \par
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glslistdesc{##2}%
      \glslistchildprelocation ##3}%
  }
}
{}
\ifdef{\@glsstyle@listgroup}
{%
  \newcommand{\glslistgroupheaderitem}[2]{\item[{#2}]}
  \newcommand{\glslistgroupafterheader}{%
      \mbox{}\par\nobreak\@afterheading
  }
  \renewglossarystyle{listgroup}{%
    \setglossarystyle{list}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glslistgroupheaderitem{##1}{\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}}%
      \glslistgroupafterheader
    }%
  }
}
{}
\ifdef{\@glsstyle@listhypergroup}
```

```
{%
  \renewglossarystyle{listhypergroup}{%
    \setglossarystyle{list}%
    \renewcommand*{\glossaryheader}{%
      \glslistnavigationitem{\glsnavigation}}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glslistgroupheaderitem{##1}{\glslistgroupheaderfmt
        {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}}%
      \glslistgroupafterheader
    }%
  }
}
{}
\ifdef{\@glsstyle@altlistgroup}
{%
  \renewglossarystyle{altlistgroup}{%
    \setglossarystyle{altlist}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glslistgroupheaderitem{##1}%
        {\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}}%
      \glslistgroupafterheader
    }%
  }
}
{}
\ifdef{\@glsstyle@altlisthypergroup}
{%
  \renewglossarystyle{altlisthypergroup}{%
    \setglossarystyle{altlist}%
    \renewcommand*{\glossaryheader}{%
      \glslistnavigationitem{\glsnavigation}}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glslistgroupheaderitem{##1}{\glslistgroupheaderfmt
        {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}}%
      \glslistgroupafterheader
    }%
  }
}
{}
\ifcsdef{@glsstyle@long}
{%
  \renewglossarystyle{long}{%
    \renewenvironment{theglossary}%
        {\begin{longtable}{lp{\glsdescwidth}}}%
        {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription
```

```
        \glsxtrprelocation ##2\tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
         &
         \glssubentryitem{##2}%
         \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
         \glsxtrprelocation ##3\tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
      \fi
   }
}
{}
\ifcsdef{@glsstyle@long3col}
{%
   \renewglossarystyle{long3col}{%
      \renewenvironment{theglossary}%
        {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{longtable}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
      \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
         &
         \glssubentryitem{##2}%
         \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
         ##3\tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{& &\tabularnewline}%
      \fi
   }
}
{}
\ifcsdef{@glsstyle@long4col}
{%
   \renewglossarystyle{long4col}{%
      \renewenvironment{theglossary}%
        {\begin{longtable}{llll}}%
        {\end{longtable}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
```

```
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription &
      \glossentrysymbol{##1} &
      ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      \glossentrysymbol{##2} & ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
    \fi
  }
}
{}
\ifcsdef{@glsstyle@longragged}
{%
  \renewglossarystyle{longragged}{%
    \renewenvironment{theglossary}%
       {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
       {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription\glsxtrprelocation ##2%
      \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}%
      \glspostdescription\glsxtrprelocation ##3%
      \tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
    \fi
  }
}
{}
\ifcsdef{@glsstyle@longragged3col}
{%
```

```
\renewglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
        >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& &\tabularnewline}%
  \fi
 }
}
{}
\ifcsdef{@glsstyle@altlongragged4col}
{%
  \renewglossarystyle{altlongragged4col}{%
    \renewenvironment{theglossary}%
      {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
          >{\raggedright}p{\glspagelistwidth}}}%
      {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription & \glossentrysymbol{##1} &
      ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      \glossentrysymbol{##2} & ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
    \fi
```

```
  }
}
{}
\ifcsdef{@glsstyle@super}
{%
  \renewglossarystyle{super}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}%
       \begin{supertabular}{lp{\glsdescwidth}}}%
      {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription
      \glsxtrprelocation ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
      \glsxtrprelocation ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
  }
}
{}
\ifcsdef{@glsstyle@super3col}
{%
  \renewglossarystyle{super3col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}%
       \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
      {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
       &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      ##3\tabularnewline
    }%
```

```
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{ & &\tabularnewline}%
      \fi
    }
  }
  {}
  \ifcsdef{@glsstyle@super4col}
  {%
    \renewglossarystyle{super4col}{%
      \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}%
          \begin{supertabular}{llll}}{%
          \end{supertabular}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
      \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription &
        \glossentrysymbol{##1} & ##2\tabularnewline
      }%
      \renewcommand{\subglossentry}[3]{%
          &
          \glssubentryitem{##2}%
          \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
          \glossentrysymbol{##2} & ##3\tabularnewline
      }%
      \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
      \else
        \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
      \fi
    }
  }
  {}
  \ifcsdef{@glsstyle@superragged}
  {%
    \renewglossarystyle{superragged}{%
      \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}%
          \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
          {\end{supertabular}}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
      \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription\glsxtrprelocation ##2%
        \tabularnewline
      }%
```

1164

```
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
        \glsxtrprelocation ##3%
        \tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
  }
}
{}
\ifcsdef{@glsstyle@superragged3col}
{%
  \renewglossarystyle{superragged3col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}%
       \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
          >{\raggedright}p{\glspagelistwidth}}}%
      {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription &
      ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
        ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{ & &\tabularnewline}%
    \fi
  }
}
{}
\ifcsdef{@glsstyle@altsuperragged4col}
{%
  \renewglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}%
       \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
```

```
          >{\raggedright}p{\glspagelistwidth}}}%
        {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription &
      \glossentrysymbol{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      \glossentrysymbol{##2} & ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & &\tabularnewline}%
    \fi
  }
}
{}
\ifdef{\@glsstyle@inline}
{%
   \renewcommand*{\glspostinline}{.\spacefactor\sfcode`\.}
   \renewcommand*{\glsinlinedescformat}[3]{%
     \space#1\glsxtrpostdescription}
   \renewcommand*{\glsinlinesubdescformat}[3]{%
     #1\glsxtrpostdescription}
}
{}
\ifdef\glstreenamefmt
{
  \newcommand{\glstreedefaultnamefmt}[1]{\textbf{#1}}
  \renewcommand{\glstreenamefmt}[1]{\glstreedefaultnamefmt{#1}}
  \def\glstreegroupheaderfmt#1{\glstreedefaultnamefmt{#1}}
  \def\glstreenavigationfmt#1{\glstreedefaultnamefmt{#1}}
  \newcommand{\glstreePreHeader}[2]{}
}
{}
\ifdef{\@glsstyle@index}
{
  \newcommand*{\glstreeprelocation}{\glsxtrprelocation}
  \newcommand*{\glstreechildprelocation}{\glstreeprelocation}
  \newcommand{\glstreegroupskip}{\indexspace}
  \newcommand{\glstreegroupheaderskip}{\nopagebreak\glstreegroupskip\nobreak}
  \renewglossarystyle{index}{%
    \renewenvironment{theglossary}%
      {\setlength{\parindent}{0pt}%
       \setlength{\parskip}{0pt plus 0.3pt}%
```

```
        \let\item\glstreeitem
        \let\subitem\glstreesubitem
        \let\subsubitem\glstreesubsubitem
      }%
    {\par}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand*{\glossentry}[2]{%
        \item\glsentryitem{##1}%
        \glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \glstreesymbol{##1}%
        \glstreeDescLoc{##1}{##2}%
    }%
    \renewcommand{\subglossentry}[3]{%
      \ifcase##1\relax
        \item
      \or
        \subitem
        \glssubentryitem{##2}%
      \else
        \subsubitem
      \fi
      \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
      \glstreechildsymbol{##2}%
      \glstreeChildDescLoc{##2}{##3}%
    }%
    \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
  }
}
{}
\ifdef{\@glsstyle@indexgroup}
{%
  \renewglossarystyle{indexgroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \item\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}
{}
\ifdef{\@glsstyle@indexhypergroup}
{%
  \renewglossarystyle{indexhypergroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glossaryheader}{%
      \item\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip\@afterheading}%
```

```
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \item\glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading}%
   }%
}
{}
\ifdef{\@glsstyle@tree}
{%
\newcommand{\glsxtrtreepredesc}{\glstreepredesc}
\newcommand{\glsxtrtreechildpredesc}{\glstreechildpredesc}
  \newcommand{\glstreedesc}[1]{%
    \glsxtrtreepredesc\glossentrydesc{#1}\glspostdescription
  }
  \newcommand{\glstreeDescLoc}[2]{%
    \ifglshasdesc{#1}%
    {\glstreedesc{#1}\glstreeprelocation}%
    {\ifglshassymbol{#1}{\glstreeprelocation}{\glstreeNoDescSymbolPreLocation}}%
    #2%
  }
  \newcommand{\glstreeNoDescSymbolPreLocation}{\space}
  \newcommand{\glstreesymbol}[1]{%
    \ifglshassymbol{#1}{\space(\glossentrysymbol{#1})}{}%
  }%
  \newcommand{\glstreechilddesc}[1]{%
    \glsxtrtreechildpredesc\glossentrydesc{#1}\glspostdescription
  }%
  \newcommand{\glstreeChildDescLoc}[2]{%
    \ifglshasdesc{#1}%
    {\glstreechilddesc{#1}\glstreechildprelocation}%
    {\ifglshassymbol{#1}{\glstreechildprelocation}%
      {\glstreeNoDescSymbolPreLocation}%
    }%
    #2%
  }%
  \newcommand{\glstreechildsymbol}[1]{%
    \glstreesymbol{#1}%
  }%
  \renewglossarystyle{tree}{%
    \renewenvironment{theglossary}%
      {\setlength{\parindent}{0pt}%
       \setlength{\parskip}{0pt plus 0.3pt}}%
      {}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \hangindent0pt\relax
      \parindent0pt\relax
```

1168

```
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \glstreesymbol{##1}%
    \glstreeDescLoc{##1}{##2}\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \glstreechildsymbol{##2}%
    \glstreeChildDescLoc{##2}{##3}\par
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
  }%
}
{}
\ifdef{\@glsstyle@treegroup}
{%
  \renewglossarystyle{treegroup}{%
    \setglossarystyle{tree}%
    \renewcommand{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading}%
  }
}
{}
\ifdef{\@glsstyle@treehypergroup}
{%
  \renewglossarystyle{treehypergroup}{%
    \setglossarystyle{tree}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip\@afterheading}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading}%
  }
}
{}
\ifdef{\@glsstyle@treenoname}
{%
  \newcommand{\glstreenonamedesc}[1]{%
```

1170

```
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \glstreesymbol{##1}%
    \glstreeDescLoc{##1}{##2}\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \glstreechildsymbol{##2}%
    \glstreeChildDescLoc{##2}{##3}\par
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
  }%
}
{}
\ifdef{\@glsstyle@treegroup}
{%
  \renewglossarystyle{treegroup}{%
    \setglossarystyle{tree}%
    \renewcommand{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading}%
  }
}
{}
\ifdef{\@glsstyle@treehypergroup}
{%
  \renewglossarystyle{treehypergroup}{%
    \setglossarystyle{tree}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip\@afterheading}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading}%
  }
}
{}
\ifdef{\@glsstyle@treenoname}
{%
  \newcommand{\glstreenonamedesc}[1]{%
```

```
      \glstreepredesc\glossentrydesc{#1}\glspostdescription
    }%
    \newcommand{\glstreenonamesymbol}[1]{%
      \ifglshassymbol{#1}{\space(\glossentrysymbol{#1})}{}%
    }%
    \newcommand{\glstreenonameDescLoc}[2]{%
      \glstreenonamedesc{#1}\glstreeprelocation#2%
    }
    \newcommand{\glstreenonamechilddesc}[1]{%
      \glossentrydesc{#1}\glspostdescription
    }%
    \newcommand{\glstreenonameChildDescLoc}[2]{%
      \glstreenonamechilddesc{#1}\glstreechildprelocation#2%
    }
    \renewglossarystyle{treenoname}{%
      \renewenvironment{theglossary}%
        {\setlength{\parindent}{0pt}%
         \setlength{\parskip}{0pt plus 0.3pt}}%
        {}%
      \renewcommand*{\glossaryheader}{}%
      \renewcommand*{\glsgroupheading}[1]{}%
      \renewcommand{\glossentry}[2]{%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \glstreenonamesymbol{##1}%
        \glstreenonameDescLoc{##1}{##2}\par
      }%
      \renewcommand{\subglossentry}[3]{%
        \hangindent##1\glstreeindent\relax
        \parindent##1\glstreeindent\relax
        \ifnum##1=1\relax
          \glssubentryitem{##2}%
        \fi
        \glstarget{##2}{\strut}%
        \glstreenonameChildDescLoc{##2}{##3}\par
      }%
      \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
    }
  }
}
{}
\ifdef{\@glsstyle@treenonamegroup}
{%
  \renewglossarystyle{treenonamegroup}{%
    \setglossarystyle{treenoname}%
    \renewcommand{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading
```

```
      }%
    }
  }
  {}
  \ifdef{\@glsstyle@treenonamehypergroup}
  {%
    \renewglossarystyle{treenonamehypergroup}{%
      \setglossarystyle{treenoname}%
      \renewcommand*{\glossaryheader}{%
        \par\noindent\glstreenavigationfmt{\glsnavigation}%
        \glstreegroupheaderskip\@afterheading}%
      \renewcommand*{\glsgroupheading}[1]{%
        \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \par\noindent
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
        \glstreegroupheaderskip\@afterheading}%
    }
  }
  {}
  \ifdef{\@glsstyle@alttree}
  {%
  \newcommand{\glsalttreepredesc}{}
  \newcommand{\glsalttreechildpredesc}{\glsalttreepredesc}
    \newcommand{\glsxtralttreeSymbolDescLocation}[2]{%
      {%
        \let\par\glsxtrAltTreePar
        \let\glsxtrtreepredesc\glsalttreepredesc
        \let\glsxtrtreechildpredesc\glsalttreechildpredesc
        \ifglshassymbol{#1}{(\glossentrysymbol{#1})\space}{}%
        \glstreeDescLoc{#1}{#2}\par
      }%
    }
    \newlength\glsxtrAltTreeIndent
    \newcommand{\glsxtrAltTreePar}{%
      \@@par
      \glsxtrAltTreeSetHangIndent
      \setlength{\parindent}{\dimexpr\hangindent+\glsxtrAltTreeIndent}%
    }
    \newcommand{\glsxtralttreeSubSymbolDescLocation}[3]{%
      \glsxtralttreeSymbolDescLocation{#2}{#3}%
    }
    \newlength\glsxtrtreetopindent
    \newcommand*{\glsxtralttreeInit}{%
      \settowidth{\glsxtrtreetopindent}{\glstreenamefmt{\glsgetwidestname\space}}%
      \glsxtrAltTreeIndent=\parindent
    }
    \newcommand*{\gglssetwidest}[2][0]{%
      \csgdef{@glswidestname\romannumeral#1}{#2}%
    }
```

1171

```
\newcommand*{\eglssetwidest}[2][0]{%
  \protected@csedef{@glswidestname\romannumeral#1}{#2}%
}
\newcommand*{\xglssetwidest}[2][0]{%
  \protected@csxdef{@glswidestname\romannumeral#1}{#2}%
}
\newcommand*{\glsupdatewidest}[2][0]{%
  \ifcsundef{@glswidestname\romannumeral#1}%
  {\csdef{@glswidestname\romannumeral#1}{#2}}%
  {%
    \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
    \settowidth{\dimen@ii}{#2}%
    \ifdim\dimen@ii>\dimen@
     \csdef{@glswidestname\romannumeral#1}{#2}%
    \fi
  }%
}
\newcommand*{\gglsupdatewidest}[2][0]{%
  \ifcsundef{@glswidestname\romannumeral#1}%
  {\csgdef{@glswidestname\romannumeral#1}{#2}}%
  {%
    \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
    \settowidth{\dimen@ii}{#2}%
    \ifdim\dimen@ii>\dimen@
     \csgdef{@glswidestname\romannumeral#1}{#2}%
    \fi
  }%
}
\newcommand*{\eglsupdatewidest}[2][0]{%
  \ifcsundef{@glswidestname\romannumeral#1}%
  {\protected@csedef{@glswidestname\romannumeral#1}{#2}}%
  {%
    \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
    \settowidth{\dimen@ii}{#2}%
    \ifdim\dimen@ii>\dimen@
     \protected@csedef{@glswidestname\romannumeral#1}{#2}%
    \fi
  }%
}
\newcommand*{\xglsupdatewidest}[2][0]{%
  \ifcsundef{@glswidestname\romannumeral#1}%
  {\protected@csxdef{@glswidestname\romannumeral#1}{#2}}%
  {%
    \settowidth{\dimen@}{\csuse{@glswidestname\romannumeral#1}}%
    \settowidth{\dimen@ii}{#2}%
    \ifdim\dimen@ii>\dimen@
     \protected@csxdef{@glswidestname\romannumeral#1}{#2}%
    \fi
  }%
}
```

```
\newcommand*{\glsgetwidestname}{\@glswidestname}
\newcommand*{\glsgetwidestsubname}[1]{%
  \ifcsundef{@glswidestname\romannumeral#1}%
  {\@glswidestname}%
  {\csuse{@glswidestname\romannumeral#1}}%
}
\let\glsFindWidestTopLevelName\glsfindwidesttoplevelname
\newrobustcmd*{\glsFindWidestUsedTopLevelName}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \ifglshasparent{\@glo@label}%
        {}%
        {%
          \settowidth{\dimen@}%
           {\glstreenamefmt{\glsentryname{\@glo@label}}}%
          \ifdim\dimen@>\gls@tmplen
            \gls@tmplen=\dimen@
            \eglssetwidest{\glsentryname{\@glo@label}}%
          \fi
        }%
      }%
      {}%
    }%
  }%
}
\newrobustcmd*{\glsFindWidestUsedAnyName}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \settowidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
      }%
      {}%
    }%
  }%
```

```
}
\newrobustcmd*{\glsFindWidestAnyName}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \settowidth{\dimen@}%
       {\glstreenamefmt{\glsentryname{\@glo@label}}}%
      \ifdim\dimen@>\gls@tmplen
        \gls@tmplen=\dimen@
        \eglssetwidest{\glsentryname{\@glo@label}}%
      \fi
    }%
  }%
}
\newrobustcmd*{\glsFindWidestUsedLevelTwo}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \dimen@i=0pt\relax
  \dimen@ii=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \ifglshasparent{\@glo@label}%
        {%
          \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@label}@parent}}%
          \ifglshasparent{\@glo@parent}%
          {%
            \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@parent}@parent}}%
            \ifglshasparent{\@glo@parent}%
            {}%
            {%
              \settowidth{\gls@tmplen}%
                 {\glstreenamefmt{\glsentryname{\@glo@label}}}%
              \ifdim\gls@tmplen>\dimen@ii
                \dimen@ii=\gls@tmplen
                \eglssetwidest[2]{\glsentryname{\@glo@label}}%
              \fi
            }%
          }%
          {%
            \settowidth{\gls@tmplen}%
               {\glstreenamefmt{\glsentryname{\@glo@label}}}%
            \ifdim\gls@tmplen>\dimen@i
              \dimen@i=\gls@tmplen
              \eglssetwidest[1]{\glsentryname{\@glo@label}}%
```

1174

```
                \fi
              }%
            }%
            {%
              \settowidth{\gls@tmplen}%
                {\glstreenamefmt{\glsentryname{\@glo@label}}}%
              \ifdim\gls@tmplen>\dimen@
                \dimen@=\gls@tmplen
                \eglssetwidest{\glsentryname{\@glo@label}}%
              \fi
            }%
          }%
          {}%
        }%
      }%
}
\newrobustcmd*{\glsFindWidestLevelTwo}[1][\@glo@types]{%
    \dimen@=0pt\relax
    \dimen@i=0pt\relax
    \dimen@ii=0pt\relax
    \forallglossaries[#1]{\@gls@type}%
    {%
      \forglsentries[\@gls@type]{\@glo@label}%
      {%
        \ifglshasparent{\@glo@label}%
        {%
          \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@label}@parent}}%
          \ifglshasparent{\@glo@parent}%
          {%
            \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{\@glo@parent}@parent}}%
            \ifglshasparent{\@glo@parent}%
            {}%
            {%
              \settowidth{\gls@tmplen}%
                {\glstreenamefmt{\glsentryname{\@glo@label}}}%
              \ifdim\gls@tmplen>\dimen@ii
                \dimen@ii=\gls@tmplen
                \eglssetwidest[2]{\glsentryname{\@glo@label}}%
              \fi
            }%
          }%
          {%
            \settowidth{\gls@tmplen}%
                {\glstreenamefmt{\glsentryname{\@glo@label}}}%
            \ifdim\gls@tmplen>\dimen@i
              \dimen@i=\gls@tmplen
              \eglssetwidest[1]{\glsentryname{\@glo@label}}%
            \fi
          }%
        }%
```

```
      {%
        \settowidth{\gls@tmplen}%
          {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\gls@tmplen>\dimen@
          \dimen@=\gls@tmplen
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
      }%
    }%
  }%
}
\newrobustcmd*{\glsFindWidestUsedAnyNameSymbol}[2][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \settowidth{\dimen@}%
          {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
        \settowidth{\dimen@}%
          {\glsentrysymbol{\@glo@label}}%
        \ifdim\dimen@>#2\relax
          #2=\dimen@
        \fi
      }%
      {}%
    }%
  }%
}
\newrobustcmd*{\glsFindWidestAnyNameSymbol}[2][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \settowidth{\dimen@}%
        {\glstreenamefmt{\glsentryname{\@glo@label}}}%
      \ifdim\dimen@>\gls@tmplen
        \gls@tmplen=\dimen@
        \eglssetwidest{\glsentryname{\@glo@label}}%
```

```
      \fi
      \settowidth{\dimen@}%
       {\glsentrysymbol{\@glo@label}}%
      \ifdim\dimen@>#2\relax
        #2=\dimen@
      \fi
    }%
  }%
}
\newrobustcmd*{\glsFindWidestUsedAnyNameSymbolLocation}[3][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  #3=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \settowidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
        \settowidth{\dimen@}%
         {\glsentrysymbol{\@glo@label}}%
        \ifdim\dimen@>#2\relax
          #2=\dimen@
        \fi
        \settowidth{\dimen@}%
         {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
        \ifdim\dimen@>#3\relax
          #3=\dimen@
        \fi
      }%
      {}%
    }%
  }%
}
\newrobustcmd*{\glsFindWidestAnyNameSymbolLocation}[3][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  #3=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
```

```
        \settowidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
        \settowidth{\dimen@}%
         {\glsentrysymbol{\@glo@label}}%
        \ifdim\dimen@>#2\relax
          #2=\dimen@
        \fi
        \settowidth{\dimen@}%
          {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
        \ifdim\dimen@>#3\relax
          #3=\dimen@
        \fi
      }%
    }%
}
\newrobustcmd*{\glsFindWidestUsedAnyNameLocation}[2][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \settowidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
        \settowidth{\dimen@}%
         {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
        \ifdim\dimen@>#2\relax
          #2=\dimen@
        \fi
      }%
      {}%
    }%
  }%
}
\newrobustcmd*{\glsFindWidestAnyNameLocation}[2][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  #2=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
```

```
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \settowidth{\dimen@}%
       {\glstreenamefmt{\glsentryname{\@glo@label}}}%
      \ifdim\dimen@>\gls@tmplen
        \gls@tmplen=\dimen@
        \eglssetwidest{\glsentryname{\@glo@label}}%
      \fi
      \settowidth{\dimen@}%
       {\GlsXtrFormatLocationList{\glsentrynumberlist{\@glo@label}}}%
      \ifdim\dimen@>#2\relax
        #2=\dimen@
      \fi
    }%
  }%
}
\newcommand*{\glsxtrComputeTreeIndent}[1]{%
  \glstreeindent=\glsxtrtreetopindent\relax
}
\newcommand*{\glsxtrComputeTreeSubIndent}[3]{%
  \ifcsundef{@glswidestname\romannumeral#1}%
  {%
    \settowidth{#3}{\glstreenamefmt{\@glswidestname\space}}%
  }%
  {%
    \settowidth{#3}{\glstreenamefmt{%
            \csname @glswidestname\romannumeral#1\endcsname\space}}%
  }%
}
\newcommand*{\glsxtrAltTreeSetHangIndent}{\hangindent\glstreeindent}
\newcommand*{\glsxtrAltTreeSetSubHangIndent}[1]{\hangindent\glstreeindent}
\renewglossarystyle{alttree}{%
  \renewenvironment{theglossary}%
    {%
     \glsxtralttreeInit
     \def\@gls@prevlevel{-1}%
     \mbox{}\par}%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \ifnum\@gls@prevlevel=0\relax
    \else
      \glsxtrComputeTreeIndent{##1}%
    \fi
    \parindent\glstreeindent
    \glsxtrAltTreeSetHangIndent
    \makebox[0pt][r]%
    {%
```

```
            \glstreenamebox{\glstreeindent}%
          {%
              \glsentryitem{##1}%
              \glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
          }%
        }%
        \glsxtralttreeSymbolDescLocation{##1}{##2}%
        \def\@gls@prevlevel{0}%
      }
      \renewcommand{\subglossentry}[3]{%
        \ifnum##1=1\relax
          \glssubentryitem{##2}%
        \fi
        \ifnum\@gls@prevlevel=##1\relax
        \else
          \glsxtrComputeTreeSubIndent{##1}{##2}{\gls@tmplen}%
          \ifnum\@gls@prevlevel<##1\relax
            \setlength\glstreeindent\gls@tmplen
            \addtolength\glstreeindent\parindent
            \parindent\glstreeindent
          \else
            \ifnum\@gls@prevlevel=0\relax
              \glsxtrComputeTreeIndent{##2}%
            \else
              \glsxtrComputeTreeSubIndent{\@gls@prevlevel}{##2}{\glstreeindent}%
            \fi
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
          \fi
        \fi
        \glsxtrAltTreeSetSubHangIndent{##1}%
        \makebox[0pt][r]{\glstreenamebox{\gls@tmplen}{%
          \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
        \glsxtralttreeSubSymbolDescLocation{##1}{##2}{##3}%
        \def\@gls@prevlevel{##1}%
      }%
      \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\glstreegroupskip\fi}%
    }
  }%
}%
{%
}
\ifdef{\@glsstyle@alttreegroup}
{%
  \renewglossarystyle{alttreegroup}{%
    \setglossarystyle{alttree}%
    \renewcommand{\glsgroupheading}[1]{\par
      \def\@gls@prevlevel{-1}%
      \hangindent0pt\relax
      \parindent0pt\relax
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
```

1180

```
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \glstreegroupheaderfmt{\glsxtr@grptitle}%
        \glstreegroupheaderskip
      }%
    }%
}%
{%
}
\ifdef{\@glsstyle@alttreehypergroup}
{%
  \renewglossarystyle{alttreehypergroup}{%
    \setglossarystyle{alttree}%
    \renewcommand*{\glossaryheader}{%
      \par
      \def\@gls@prevlevel{-1}%
      \hangindent0pt\relax
      \parindent0pt\relax
      \glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par
      \def\@gls@prevlevel{-1}%
      \hangindent0pt\relax
      \parindent0pt\relax
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcolindexgroup}
{%
  \renewglossarystyle{mcolindexgroup}{%
    \setglossarystyle{mcolindex}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \item\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcolindexhypergroup}
{%
```

1181

```
\renewglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}%
    \glstreegroupheaderskip\@afterheading
  }%
  \renewcommand*{\glsgroupheading}[1]{%
    \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
    \glstreePreHeader{##1}{\glsxtr@grptitle}%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
    \glstreegroupheaderskip\@afterheading
  }%
}
}%
{%
}
\ifdef{\@glsstyle@mcolindexspannav}
{%
  \renewglossarystyle{mcolindexspannav}{%
    \setglossarystyle{index}%
    \renewenvironment{theglossary}%
    {%
     \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]%
     \setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
     \let\item\glstreeitem}%
    {\end{multicols}}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \item\glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcoltreegroup}
{%
  \renewglossarystyle{mcoltreegroup}{%
    \setglossarystyle{mcoltree}%
    \renewcommand{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
```

```
}%
{%
}
\ifdef{\@glsstyle@mcoltreehypergroup}
{%
  \renewglossarystyle{mcoltreehypergroup}{%
    \setglossarystyle{mcoltree}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcoltreespannav}
{%
  \renewglossarystyle{mcoltreespannav}{%
    \setglossarystyle{tree}%
    \renewenvironment{theglossary}%
    {%
       \begin{multicols}{\glsmcols}%
         [\noindent\glstreenavigationfmt{\glsnavigation}]%
       \setlength{\parindent}{0pt}%
       \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcoltreenonamegroup}
{%
  \renewglossarystyle{mcoltreenonamegroup}{%
    \setglossarystyle{mcoltreenoname}%
    \renewcommand{\glsgroupheading}[1]{%
```

```
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glsxtr@grptitle}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcoltreenonamehypergroup}
{%
  \renewglossarystyle{mcoltreenonamehypergroup}{%
    \setglossarystyle{mcoltreenoname}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading}%
  }
}%
{%
}
\ifdef{\@glsstyle@mcoltreenonamespannav}
{%
  \renewglossarystyle{mcoltreenonamespannav}{%
    \setglossarystyle{treenoname}%
    \renewenvironment{theglossary}%
    {%
       \begin{multicols}{\glsmcols}%
         [\noindent\glstreenavigationfmt{\glsnavigation}]%
       \setlength{\parindent}{0pt}%
       \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip\@afterheading}%
  }
}%
{%
}
\ifdef{\@glsstyle@mcolalttree}
```

```
{%
  \renewglossarystyle{mcolalttree}{%
    \setglossarystyle{alttree}%
    \renewenvironment{theglossary}%
    {%
        \glsxtralttreeInit
        \def\@gls@prevlevel{-1}%
        \begin{multicols}{\glsmcols}%
    }%
    {\par\end{multicols}}%
  }
}%
{%
}
\ifdef{\@glsstyle@mcolalttreegroup}
{%
  \renewglossarystyle{mcolalttreegroup}{%
    \setglossarystyle{mcolalttree}%
    \renewcommand{\glsgroupheading}[1]{%
        \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsxtr@grptitle}%
        \glstreegroupheaderskip
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcolalttreehypergroup}
{%
  \renewglossarystyle{mcolalttreehypergroup}{%
    \setglossarystyle{mcolalttree}%
    \renewcommand*{\glossaryheader}{%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreenavigationfmt{\glsnavigation}%
        \glstreegroupheaderskip
    }%
    \renewcommand*{\glsgroupheading}[1]{%
        \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
        \glstreePreHeader{##1}{\glsxtr@grptitle}%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
```

```
        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
        \glstreegroupheaderskip
      }%
    }
  }%
}%
{%
}
\ifdef{\@glsstyle@mcolalttreespannav}
{%
  \renewglossarystyle{mcolalttreespannav}{%
    \setglossarystyle{alttree}%
    \renewenvironment{theglossary}%
    {%
        \glsxtralttreeInit
        \def\@gls@prevlevel{-1}%
        \begin{multicols}{\glsmcols}%
            [\noindent\glstreenavigationfmt{\glsnavigation}]%
    }%
    {\par\end{multicols}}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glsxtrgetgrouptitle{##1}{\glsxtr@grptitle}%
      \glstreePreHeader{##1}{\glsxtr@grptitle}%
      \par
      \def\@gls@prevlevel{-1}%
      \hangindent0pt\relax
      \parindent0pt\relax
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsxtr@grptitle}}%
      \glstreegroupheaderskip
    }%
  }
}%
{%
}
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glsxtr@current@style}
\fi
```

9.4   Rollback v1.48 (glossary-bookindex-2021-11-22.sty)

Version 1.48 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-bookindex}[2021/11/22 v1.48 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}
\newcommand{\glsxtrbookindexcols}{2}
\newcommand*{\glsxtrbookindexname}[1]{\glossentryname{#1}}
\newcommand*{\glsxtrbookindexsubname}[1]{\glsxtrbookindexname{#1}}
```

```
\providecommand*{\glsxtrprelocation}{\space}

\newcommand*{\glsxtrbookindexprelocation}[1]{%
  \glsxtrifhasfield{location}{#1}%
  {,\glsxtrprelocation}%
  {\glsxtrprelocation}%
}
\newcommand*{\glsxtrbookindexsubprelocation}[1]{%
 \glsxtrbookindexprelocation{#1}%
}
\newcommand*{\glsxtrbookindexlocation}[2]{#2}
\newcommand*{\glsxtrbookindexsublocation}{\glsxtrbookindexlocation}
\newcommand{\glsxtrbookindexparentchildsep}{\nopagebreak}
\newcommand{\glsxtrbookindexparentsubchildsep}{\glsxtrbookindexparentchildsep}
\newcommand{\glsxtrbookindexbetween}[2]{}
\newcommand{\glsxtrbookindexsubbetween}[2]{}
\newcommand{\glsxtrbookindexsubsubbetween}[2]{}
\newcommand{\glsxtrbookindexatendgroup}[1]{}
\newcommand{\glsxtrbookindexsubatendgroup}[1]{}
\newcommand{\glsxtrbookindexsubsubatendgroup}[1]{}
\newcommand{\glsxtrbookindexgroupskip}{\ifglsnogroupskip\else\indexspace\fi}
\newcommand*{\glsxtrbookindexformatheader}[1]{%
  \par{\centering\glstreegroupheaderfmt{#1}\par}%
}
\ifdef\pdfbookmark
{%
  \newcommand*{\glsxtrbookindexbookmark}[2]{%
    \ifdefstring{\@@glossarysec}{chapter}%
    {\pdfbookmark[1]{#1}{#2}}%
    {\pdfbookmark[2]{#1}{#2}}%
  }
}
{%
  \newcommand*{\glsxtrbookindexbookmark}[2]{}
}
\newcommand*{\glsxtrbookindexbookmarkprefix}{\currentglossary.}
\newcommand*{\glsxtrbookindexcolspread}{}
\newcommand*{\glsxtrbookindexmulticolsenv}{multicols}
\newglossarystyle{bookindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
  {%
    \ifnum\glsxtrbookindexcols>1\relax
      \ifdefempty\glsxtrbookindexcolspread
      {%
        \edef\glsxtr@beginbookindex{%
          \noexpand\begin{\glsxtrbookindexmulticolsenv}
            {\glsxtrbookindexcols}%
        }%
      }%
```

```
    {%
      \edef\glsxtr@beginbookindex{%
        \noexpand\begin{\glsxtrbookindexmulticolsenv}%
          {\glsxtrbookindexcols}[\glsxtrbookindexcolspread]%
      }%
    }%
  \else
    \def\glsxtr@beginbookindex{}%
  \fi
  \glsxtr@beginbookindex
  \setlength{\parindent}{0pt}%
  \setlength{\parskip}{0pt plus 0.3pt}%
  \let\@glsxtr@bookindex@sep\glsxtrbookindexparentchildsep
  \let\@glsxtr@bookindex@subsep\glsxtrbookindexparentsubchildsep
  \let\@glsxtr@bookindex@between\@gobble
  \let\@glsxtr@bookindex@subbetween\@gobble
  \let\@glsxtr@bookindex@subsubbetween\@gobble
  \let\@glsxtr@bookindex@atendgroup\relax
  \let\@glsxtr@bookindex@subatendgroup\relax
  \let\@glsxtr@bookindex@subsubatendgroup\relax
  \let\@glsxtr@bookindexgroupskip\relax
}%
{%
  \@glsxtr@bookindex@subsubatendgroup
  \@glsxtr@bookindex@subatendgroup
  \@glsxtr@bookindex@atendgroup
  \ifnum\glsxtrbookindexcols>1\relax
    \edef\glsxtr@endbookindex{%
      \noexpand\end{\glsxtrbookindexmulticolsenv}%
    }%
  \else
    \def\glsxtr@endbookindex{}%
  \fi
  \glsxtr@endbookindex
}%
\renewcommand*{\glossaryheader}{\raggedright}%
\renewcommand*{\glossentry}[2]{%
  \@glsxtr@bookindex@between{##1}%
  \let\@glsxtr@bookindex@sep\glsxtrbookindexparentchildsep
  \let\@glsxtr@bookindex@subsep\glsxtrbookindexparentsubchildsep
  \let\@glsxtr@bookindex@subbetween\@gobble
  \let\@glsxtr@bookindex@subsubbetween\@gobble
  \edef\@glsxtr@bookindex@between{%
      \noexpand\glsxtrbookindexbetween{##1}%
  }%
  \edef\@glsxtr@bookindex@atendgroup{%
    \noexpand\glsxtrbookindexatendgroup{##1}%
  }%
  \let\@glsxtr@bookindex@subatendgroup\relax
  \let\@glsxtr@bookindex@subsubatendgroup\relax
```

```
    \glstreeitem
      \glsentryitem{##1}%
      \glstarget{##1}{\glsxtrbookindexname{##1}}%
    \glsxtrbookindexprelocation{##1}%
    \glsxtrbookindexlocation{##1}{##2}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifcase##1\relax
    \glstreeitem
  \or
    \@glsxtr@bookindex@sep
    \@glsxtr@bookindex@subbetween{##2}%
    \let\@glsxtr@bookindex@sep\relax
    \let\@glsxtr@bookindex@subsubbetween\@gobble
    \let\@glsxtr@bookindex@subsep\glsxtrbookindexparentsubchildsep
    \edef\@glsxtr@bookindex@subbetween{%
        \noexpand\glsxtrbookindexsubbetween{##2}%
    }%
    \edef\@glsxtr@bookindex@atsubendgroup{%
      \noexpand\glsxtrbookindexatsubendgroup{##1}%
    }%
    \glstreesubitem
    \glssubentryitem{##2}%
  \else
    \@glsxtr@bookindex@subsep
    \@glsxtr@bookindex@subsubbetween{##2}%
    \let\@glsxtr@bookindex@subsep\relax
    \edef\@glsxtr@bookindex@subsubbetween{%
        \noexpand\glsxtrbookindexsubsubbetween{##2}%
    }%
    \edef\@glsxtr@bookindex@atsubsubendgroup{%
      \noexpand\glsxtrbookindexatsubsubendgroup{##1}%
    }%
    \glstreesubsubitem
  \fi
  \glstarget{##2}{\glsxtrbookindexsubname{##2}}%
  \glsxtrbookindexsubprelocation{##2}%
  \glsxtrbookindexsublocation{##2}{##3}%
}%
\renewcommand*{\glsgroupskip}{}%
\renewcommand*{\glsgroupheading}[1]{%
  \@glsxtr@bookindex@subsubatendgroup
  \@glsxtr@bookindex@subatendgroup
  \@glsxtr@bookindex@atendgroup
  \@glsxtr@bookindexgroupskip
  \let\@glsxtr@bookindexgroupskip\glsxtrbookindexgroupskip
  \let\@glsxtr@bookindex@between\@gobble
  \let\@glsxtr@bookindex@atendgroup\relax
  \let\@glsxtr@bookindex@subatendgroup\relax
  \let\@glsxtr@bookindex@subsubatendgroup\relax
```

```
      \glsxtrgetgrouptitle{##1}{\glsxtrcurrentgrptitle}%
      \glsxtrbookindexbookmark{\glsxtrcurrentgrptitle}{\glsxtrbookindexbookmarkprefix##1}%
      \glsxtrbookindexformatheader{\glsxtrcurrentgrptitle}%
      \nopagebreak\indexspace\nopagebreak\@afterheading
   }%
}
\newcommand{\glsxtrbookindexthepage}{%
 \ifdef\currentglossary{\currentglossary.\arabic{page}}{\arabic{page}}%
}
\newcommand*{\glsxtrbookindexmarkentry}[1]{%
  \protected@write\@auxout
  {\let\glsxtrbookindexthepage\relax}%
  {\string\glsxtr@setbookindexmark{\glsxtrbookindexthepage}{#1}}%
}
\newcommand*{\glsxtr@setbookindexmark}[2]{%
 \ifcsundef{glsxtr@idxfirstmark@#1}%
 {\csgdef{glsxtr@idxfirstmark@#1}{#2}}%
 {}%
 \csgdef{glsxtr@idxlastmark@#1}{#2}%
}
\newcommand*{\glsxtrbookindexfirstmarkfmt}[1]{%
  \glsentryname{#1}%
}
\newcommand*{\glsxtrbookindexfirstmark}{%
  \letcs{\glsxtr@label}{glsxtr@idxfirstmark@\glsxtrbookindexthepage}%
  \ifdef\glsxtr@label
  {\glsxtrbookindexfirstmarkfmt{\glsxtr@label}}%
  {}%
}
\newcommand*{\glsxtrbookindexlastmarkfmt}[1]{%
  \glsentryname{#1}%
}
\newcommand*{\glsxtrbookindexlastmark}{%
  \letcs{\glsxtr@label}{glsxtr@idxlastmark@\glsxtrbookindexthepage}%
  \ifdef\glsxtr@label
  {\glsxtrbookindexlastmarkfmt{\glsxtr@label}}%
  {}%
}
```

9.5   Rollback v1.48 (glossary-longextra-2021-11-22.sty)

Version 1.48 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-longextra}[2021/11/22 v1.48 (NLCT)]
\RequirePackage{glossary-longbooktabs}
\newcommand{\glslongextraNameFmt}[1]{%
 \glsentryitem{#1}\glstarget{#1}{\glossentryname{#1}}%
}
\newcommand{\glslongextraDescFmt}[1]{%
```

```
  \glossentrydesc{#1}\glspostdescription
}
\newcommand{\glslongextraSymbolFmt}[1]{\glossentrysymbol{#1}}
\newcommand{\glslongextraLocationFmt}[2]{#2}
\newcommand{\glslongextraSubNameFmt}[2]{%
 \glssubentryitem{#2}\glstarget{#2}{\strut}%
}
\newcommand{\glslongextraSubDescFmt}[2]{%
  \glslongextraDescFmt{#2}%
}
\newcommand{\glslongextraSubSymbolFmt}[2]{%
  \glslongextraSymbolFmt{#2}%
}
\newcommand{\glslongextraSubLocationFmt}[3]{#3}
\newcommand{\glslongextraNameAlign}{l}
\newcommand{\glslongextraDescAlign}{>{\raggedright}p{\glsdescwidth}}
\newcommand{\glslongextraSymbolAlign}{c}
\newcommand{\glslongextraLocationAlign}{>{\raggedright}p{\glspagelistwidth}}
\newcommand{\glslongextraGroupHeading}[2]{}
\newcommand{\glslongextraHeaderFmt}[1]{\textbf{#1}}
\newcommand{\glslongextraNameDescHeader}{%
 \glslongextraNameDescTabularHeader\endhead
 \glslongextraNameDescTabularFooter\endfoot
}
\newcommand{\glslongextraNameDescTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\descriptionname\tabularnewline
 \midrule
}
\newcommand{\glslongextraNameDescTabularFooter}{%
 \bottomrule
}
\newcommand*{\glslongextraSetWidest}[1]{%
 \def\@glslongextrawidestname{#1}%
}
\newcommand*{\@glslongextrawidestname}{\csuse{@glswidestname}}
\newcommand*{\glslongextraUpdateWidest}[1]{%
  \ifundef\@glslongextrawidestname
  {\def\@glslongextrawidestname{#1}}%
  {%
    \settowidth{\dimen@}{\@glslongextrawidestname}%
    \settowidth{\dimen@ii}{#1}%
    \ifdim\dimen@ii>\dimen@
     \def\@glslongextrawidestname{#1}%
    \fi
  }%
}
\newcommand*{\glslongextraUpdateWidestChild}[2]{}
\newcommand{\glslongextraSetDescWidth}{%
```

```
    \settowidth{\gls@tmplen}{\glslongextraHeaderFmt\entryname}%
    \settowidth{\dimen@}{\glsnamefont{\@glslongextrawidestname}}%
    \ifdim\dimen@>\gls@tmplen
     \gls@tmplen=\dimen@
    \fi
    \setlength{\glsdescwidth}{\dimexpr\linewidth-4\tabcolsep-\gls@tmplen}%
}
\newcommand{\glslongextraSymSetDescWidth}{%
  \glslongextraSetDescWidth
  \settowidth{\gls@tmplen}{\glslongextraHeaderFmt\symbolname}%
  \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\gls@tmplen}%
}
\newcommand{\glslongextraLocSetDescWidth}{%
  \glslongextraSetDescWidth
  \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\glspagelistwidth}%
}
\newcommand{\glslongextraSymLocSetDescWidth}{%
  \glslongextraSymSetDescWidth
  \setlength{\glsdescwidth}{\dimexpr\glsdescwidth-2\tabcolsep-\glspagelistwidth}%
}
\newif\ifGlsLongExtraUseTabular
\GlsLongExtraUseTabularfalse
\newcommand*{\glslongextraTabularVAlign}{c}
\newglossarystyle{long-name-desc}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraDescAlign}}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraNameDescTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraNameDescTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraDescAlign}}%
      \@glslongextra@begintab
```

1192

```
    }%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\glslongextraNameDescHeader}%
 \fi
 \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{2}{##1}}%
 \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraDescFmt{##1}\tabularnewline
 }%
 \renewcommand{\subglossentry}[3]{%
    \glslongextraSubNameFmt{##1}{##2}
    &
    \glslongextraSubDescFmt{##1}{##2}%
    \tabularnewline
 }%
 \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
 \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
 \fi
}
\newcommand{\glslongextraNameDescLocationHeader}{%
 \glslongextraNameDescLocationTabularHeader\endhead
 \glslongextraNameDescLocationTabularFooter\endfoot
}
\newcommand{\glslongextraNameDescLocationTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\pagelistname\tabularnewline
 \midrule
}
\newcommand{\glslongextraNameDescLocationTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-name-desc-loc}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
    {%
       \glslongextraLocSetDescWidth
       \edef\@glslongextra@begintab{%
          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
             \expandonce\glslongextraNameAlign
             \expandonce\glslongextraDescAlign
             \expandonce\glslongextraLocationAlign
          }}%
       \@glslongextra@begintab
    }%
    {%
```

```
        \glslongextraNameDescLocationTabularFooter
        \end{tabular}%
      }%
    \renewcommand*{\glossaryheader}{\glslongextraNameDescLocationTabularHeader}%
  \else
    \renewenvironment{theglossary}%
     {%
        \glspatchLToutput
        \glslongextraLocSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraDescAlign
            \expandonce\glslongextraLocationAlign
        }}%
        \@glslongextra@begintab
      }%
     {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraNameDescLocationHeader}%
    \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraLocationFmt{##1}{##2}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
      \glslongextraSubNameFmt{##1}{##2}&
      \glslongextraSubDescFmt{##1}{##2}&
      \glslongextraSubLocationFmt{##1}{##2}{##3}%
      \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newcommand{\glslongextraDescNameHeader}{%
 \glslongextraDescNameTabularHeader\endhead
 \glslongextraDescNameTabularFooter\endfoot
}
\newcommand{\glslongextraDescNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\descriptionname&
 \glslongextraHeaderFmt\entryname \tabularnewline
 \midrule
}
\newcommand{\glslongextraDescNameTabularFooter}{%
 \bottomrule
```

```
}
\newglossarystyle{long-desc-name}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraNameAlign}}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraDescNameTabularFooter
      \end{tabular}%
    }%
    \renewcommand*{\glossaryheader}{\glslongextraDescNameTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraNameAlign}}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraDescNameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{2}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraDescFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newcommand{\glslongextraLocationDescNameHeader}{%
 \glslongextraLocationDescNameTabularHeader\endhead
 \glslongextraLocationDescNameTabularFooter\endfoot
```

```
}
\newcommand{\glslongextraLocationDescNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\pagelistname&
 \glslongextraHeaderFmt\descriptionname&
 \glslongextraHeaderFmt\entryname \tabularnewline
 \midrule
}
\newcommand{\glslongextraLocationDescNameTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-loc-desc-name}%
{%
  \ifGlsLongExtraUseTabular
   {%
     \glslongextraLocSetDescWidth
     \edef\@glslongextra@begintab{%
       \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
         \expandonce\glslongextraLocationAlign
         \expandonce\glslongextraDescAlign
         \expandonce\glslongextraNameAlign}}%
     \@glslongextra@begintab
   }%
   {%
     \glslongextraLocationDescNameTabularFooter
     \end{tabular}%
   }%
   \renewcommand*{\glossaryheader}{\glslongextraLocationDescNameTabularHeader}%
  \else
  \renewenvironment{theglossary}%
   {%
     \glspatchLToutput
     \glslongextraLocSetDescWidth
     \edef\@glslongextra@begintab{%
       \noexpand\begin{longtable}{%
         \expandonce\glslongextraLocationAlign
         \expandonce\glslongextraDescAlign
         \expandonce\glslongextraNameAlign}}%
     \@glslongextra@begintab
   }%
   {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraLocationDescNameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraLocationFmt{##1}{##2} &
    \glslongextraDescFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
```

```
      \glslongextraSubLocationFmt{##1}{##2}{##3} &
      \glslongextraSubDescFmt{##1}{##2} &
      \glslongextraSubNameFmt{##1}{##2}\tabularnewline
   }%
   \ifglsnogroupskip
     \renewcommand*{\glsgroupskip}{}%
   \else
     \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
   \fi
}
\newcommand{\glslongextraNameDescSymHeader}{%
 \glslongextraNameDescSymTabularHeader\endhead
 \glslongextraNameDescSymTabularFooter\endfoot
}
\newcommand{\glslongextraNameDescSymTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\symbolname\tabularnewline
 \midrule
}
\newcommand{\glslongextraNameDescSymTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-name-desc-sym}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSymSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraNameDescSymTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraNameDescSymTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
```

```
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraDescAlign
            \expandonce\glslongextraSymbolAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\glslongextraNameDescSymHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraSymbolFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
      \glslongextraSubNameFmt{##1}{##2} &
      \glslongextraSubDescFmt{##1}{##2} &
      \glslongextraSubSymbolFmt{##1}{##2}%
      \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newcommand{\glslongextraNameDescSymLocationHeader}{%
 \glslongextraNameDescSymLocationTabularHeader\endhead
 \glslongextraNameDescSymLocationTabularFooter\endfoot
}
\newcommand{\glslongextraNameDescSymLocationTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\pagelistname\tabularnewline
 \midrule
}
\newcommand{\glslongextraNameDescSymLocationTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-name-desc-sym-loc}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSymLocSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
```

1198

```
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraLocationAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraNameDescSymLocationTabularFooter
      \end{tabular}%
    }%
  \renewcommand*{\glossaryheader}{\glslongextraNameDescSymLocationTabularHeader}%
  \else
  \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymLocSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraNameAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraLocationAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\glslongextraNameDescSymLocationHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraSymbolFmt{##1}&
    \glslongextraLocationFmt{##1}{##2}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubNameFmt{##1}{##2} &
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubSymbolFmt{##1}{##2}&
    \glslongextraSubLocationFmt{##1}{##2}{##3}%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newcommand{\glslongextraNameSymDescHeader}{%
```

```
 \glslongextraNameSymDescTabularHeader\endhead
 \glslongextraNameSymDescTabularFooter\endfoot
}
\newcommand{\glslongextraNameSymDescTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\descriptionname\tabularnewline
 \midrule
}
\newcommand{\glslongextraNameSymDescTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-name-sym-desc}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
     {%
        \glslongextraSymSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraSymbolAlign
            \expandonce\glslongextraDescAlign
          }}%
        \@glslongextra@begintab
     }%
     {%
        \glslongextraNameSymDescTabularFooter
        \end{tabular}%
     }%
   \renewcommand*{\glossaryheader}{\glslongextraNameSymDescTabularHeader}%
  \else
   \renewenvironment{theglossary}%
     {%
        \glspatchLToutput
        \glslongextraSymSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraSymbolAlign
            \expandonce\glslongextraDescAlign
          }}%
        \@glslongextra@begintab
     }%
     {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraNameSymDescHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
  \renewcommand{\glossentry}[2]{%
```

```
      \glslongextraNameFmt{##1} &
      \glslongextraSymbolFmt{##1} &
      \glslongextraDescFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
      \glslongextraSubNameFmt{##1}{##2} &
      \glslongextraSubSymbolFmt{##1}{##2} &
      \glslongextraSubDescFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newcommand{\glslongextraNameSymDescLocationHeader}{%
 \glslongextraNameSymDescLocationTabularHeader\endhead
 \glslongextraNameSymDescLocationTabularFooter\endfoot
}
\newcommand{\glslongextraNameSymDescLocationTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\entryname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\pagelistname\tabularnewline
 \midrule
}
\newcommand{\glslongextraNameSymDescLocationTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-name-sym-desc-loc}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
    {%
        \glslongextraSymLocSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraSymbolAlign
            \expandonce\glslongextraDescAlign
            \expandonce\glslongextraLocationAlign
          }}%
        \@glslongextra@begintab
    }%
    {%
        \glslongextraNameSymDescLocationTabularFooter
        \end{tabular}%
    }%
    \renewcommand*{\glossaryheader}{\glslongextraNameSymDescLocationTabularHeader}%
```

```
  \else
    \renewenvironment{theglossary}%
     {%
        \glspatchLToutput
        \glslongextraSymLocSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{longtable}{%
            \expandonce\glslongextraNameAlign
            \expandonce\glslongextraSymbolAlign
            \expandonce\glslongextraDescAlign
            \expandonce\glslongextraLocationAlign
          }}%
        \@glslongextra@begintab
     }%
     {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraNameSymDescLocationHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraSymbolFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraLocationFmt{##1}{##2}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubNameFmt{##1}{##2} &
    \glslongextraSubSymbolFmt{##1}{##2} &
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubLocationFmt{##1}{##2}{##3}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newcommand{\glslongextraSymDescNameHeader}{%
 \glslongextraSymDescNameTabularHeader\endhead
 \glslongextraSymDescNameTabularFooter\endfoot
}
\newcommand{\glslongextraSymDescNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\entryname\tabularnewline
 \midrule
}
\newcommand{\glslongextraSymDescNameTabularFooter}{%
 \bottomrule
}
```

```
\newglossarystyle{long-sym-desc-name}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
     {%
       \glslongextraSymSetDescWidth
       \edef\@glslongextra@begintab{%
         \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
           \expandonce\glslongextraSymbolAlign
           \expandonce\glslongextraDescAlign
           \expandonce\glslongextraNameAlign
         }}%
       \@glslongextra@begintab
     }%
     {%
       \glslongextraSymDescNameTabularFooter
       \end{tabular}%
     }%
    \renewcommand*{\glossaryheader}{\glslongextraSymDescNameTabularHeader}%
  \else
    \renewenvironment{theglossary}%
     {%
       \glspatchLToutput
       \glslongextraSymSetDescWidth
       \edef\@glslongextra@begintab{%
         \noexpand\begin{longtable}{%
           \expandonce\glslongextraSymbolAlign
           \expandonce\glslongextraDescAlign
           \expandonce\glslongextraNameAlign
         }}%
       \@glslongextra@begintab
     }%
     {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\glslongextraSymDescNameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraSymbolFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubSymbolFmt{##1}{##2} &
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
```

```
  \fi
}
\newcommand{\glslongextraLocationSymDescNameHeader}{%
 \glslongextraLocationSymDescNameTabularHeader\endhead
 \glslongextraLocationSymDescNameTabularFooter\endfoot
}
\newcommand{\glslongextraLocationSymDescNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\pagelistname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\entryname\tabularnewline
 \midrule
}
\newcommand{\glslongextraLocationSymDescNameTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-loc-sym-desc-name}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
     {%
       \glslongextraSymLocSetDescWidth
       \edef\@glslongextra@begintab{%
         \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
           \expandonce\glslongextraLocationAlign
           \expandonce\glslongextraSymbolAlign
           \expandonce\glslongextraDescAlign
           \expandonce\glslongextraNameAlign
         }}%
       \@glslongextra@begintab
     }%
     {%
       \glslongextraLocationSymDescNameTabularFooter
       \end{tabular}%
     }%
    \renewcommand*{\glossaryheader}{\glslongextraLocationSymDescNameTabularHeader}%
  \else
   \renewenvironment{theglossary}%
     {%
       \glspatchLToutput
       \glslongextraSymLocSetDescWidth
       \edef\@glslongextra@begintab{%
         \noexpand\begin{longtable}{%
           \expandonce\glslongextraLocationAlign
           \expandonce\glslongextraSymbolAlign
           \expandonce\glslongextraDescAlign
           \expandonce\glslongextraNameAlign
         }}%
       \@glslongextra@begintab
```

```
    }%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\glslongextraLocationSymDescNameHeader}%
 \fi
 \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
 \renewcommand{\glossentry}[2]{%
    \glslongextraLocationFmt{##1}{##2} &
    \glslongextraSymbolFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
 }%
 \renewcommand{\subglossentry}[3]{%
    \glslongextraSubLocationFmt{##1}{##2}{##3} &
    \glslongextraSubSymbolFmt{##1}{##2} &
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
 }%
 \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
 \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
 \fi
}
\newcommand{\glslongextraDescSymNameHeader}{%
 \glslongextraDescSymNameTabularHeader\endhead
 \glslongextraDescSymNameTabularFooter\endfoot
}
\newcommand{\glslongextraDescSymNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\entryname\tabularnewline
 \midrule
}
\newcommand{\glslongextraDescSymNameTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-desc-sym-name}%
{%
  \ifGlsLongExtraUseTabular
    \renewenvironment{theglossary}%
      {%
        \glslongextraSymSetDescWidth
        \edef\@glslongextra@begintab{%
          \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
            \expandonce\glslongextraDescAlign
            \expandonce\glslongextraSymbolAlign
            \expandonce\glslongextraNameAlign
          }}%
        \@glslongextra@begintab
```

```
    }%
    {%
      \glslongextraDescSymNameTabularFooter
      \end{tabular}%
    }%
  \renewcommand*{\glossaryheader}{\glslongextraDescSymNameTabularHeader}%
 \else
  \renewenvironment{theglossary}%
   {%
     \glspatchLToutput
     \glslongextraSymSetDescWidth
     \edef\@glslongextra@begintab{%
       \noexpand\begin{longtable}{%
         \expandonce\glslongextraDescAlign
         \expandonce\glslongextraSymbolAlign
         \expandonce\glslongextraNameAlign
       }}%
     \@glslongextra@begintab
   }%
   {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\glslongextraDescSymNameHeader}%
 \fi
 \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{3}{##1}}%
 \renewcommand{\glossentry}[2]{%
   \glslongextraDescFmt{##1} &
   \glslongextraSymbolFmt{##1} &
   \glslongextraNameFmt{##1}\tabularnewline
 }%
 \renewcommand{\subglossentry}[3]{%
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubSymbolFmt{##1}{##2} &
    \glslongextraSubNameFmt{##1}{##2}\tabularnewline
 }%
 \ifglsnogroupskip
   \renewcommand*{\glsgroupskip}{}%
 \else
   \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
 \fi
}
\newcommand{\glslongextraLocationDescSymNameHeader}{%
 \glslongextraLocationDescSymNameTabularHeader\endhead
 \glslongextraLocationDescSymNameTabularFooter\endfoot
}
\newcommand{\glslongextraLocationDescSymNameTabularHeader}{%
 \toprule
 \glslongextraHeaderFmt\pagelistname &
 \glslongextraHeaderFmt\descriptionname &
 \glslongextraHeaderFmt\symbolname &
 \glslongextraHeaderFmt\entryname\tabularnewline
 \midrule
```

```
}
\newcommand{\glslongextraLocationDescSymNameTabularFooter}{%
 \bottomrule
}
\newglossarystyle{long-loc-desc-sym-name}%
{%
  \ifGlsLongExtraUseTabular
   \renewenvironment{theglossary}%
    {%
      \glslongextraSymLocSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{tabular}[\glslongextraTabularVAlign]{%
          \expandonce\glslongextraLocationAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {%
      \glslongextraLocationDescSymNameTabularFooter
      \end{tabular}%
    }%
   \renewcommand*{\glossaryheader}{\glslongextraLocationDescSymNameTabularHeader}%
  \else
   \renewenvironment{theglossary}%
    {%
      \glspatchLToutput
      \glslongextraSymLocSetDescWidth
      \edef\@glslongextra@begintab{%
        \noexpand\begin{longtable}{%
          \expandonce\glslongextraLocationAlign
          \expandonce\glslongextraDescAlign
          \expandonce\glslongextraSymbolAlign
          \expandonce\glslongextraNameAlign
        }}%
      \@glslongextra@begintab
    }%
    {\end{longtable}}%
   \renewcommand*{\glossaryheader}{\glslongextraLocationDescSymNameHeader}%
  \fi
  \renewcommand*{\glsgroupheading}[1]{\glslongextraGroupHeading{4}{##1}}%
  \renewcommand{\glossentry}[2]{%
    \glslongextraLocationFmt{##1}{##2} &
    \glslongextraDescFmt{##1} &
    \glslongextraSymbolFmt{##1} &
    \glslongextraNameFmt{##1}\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    \glslongextraSubLocationFmt{##1}{##2}{##3} &
```

```
      \glslongextraSubDescFmt{##1}{##2} &
      \glslongextraSubSymbolFmt{##1}{##2} &
      \glslongextraSubNameFmt{##1}{##2}\tabularnewline
   }%
   \ifglsnogroupskip
     \renewcommand*{\glsgroupskip}{}%
   \else
     \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
   \fi
 }
```

9.6 Rollback v1.48 (glossary-topic-2021-11-22.sty)

Version 1.48 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-topic}[2021/11/22 v1.48 (NLCT)]
\RequirePackage{multicol}
\newglossarystyle{topic}{%
  \renewenvironment{theglossary}%
  {%
    \glstopicInit
    \def\glstopic@prechildren{}%
    \def\glstopic@prevlevel{-1}%
  }%
  {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
    \def\glstopic@prevlevel{-1}%
    \glstopicGroupHeading{##1}%
  }%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent\glstopicParIndent\relax
    \glstopicItem{##1}{##2}%
     \ifglshasdesc{##1}%
     {%
       \def\glstopic@prechildren{}%
     }%
     {%
       \def\glstopic@prechildren{\nopagebreak}%
     }%
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifnum\glstopic@prevlevel=0\relax\glstopic@prechildren\fi
    \def\glstopic@prevlevel{##1}%
    \begingroup
    \glstopicAssignSubIndent{##1}%
    \glstopicSubItem{##1}{##2}{##3}%
    \par
```

```
    \endgroup
  }%
  \renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glstopicGroupHeading}[1]{}
\newcommand*{\glstopicItem}[2]{%
  \glspar\glstopicPreSkip\glspar\noindent
  \glstopicMarker{#1}%
  \glstopicTitleFont
  {%
    \glsentryitem{#1}\glstarget{#1}{\glstopicTitle{#1}}%
  }%
  \ifglshasdesc{#1}%
  {\glspar\nobreak\glstopicMidSkip\glspar\nobreak
   \@afterheading\glstopicDesc{#1}\glspar\glstopicPostSkip}%
  {\glspar\nobreak\glstopicPostSkip}%
  \glstopicLoc{#1}{#2}%
}
\newcommand*{\glstopicMarker}[1]{}
\newcommand*{\glstopicTitle}[1]{\Glossentryname{#1}%
  \ifglshassymbol{#1}{\space(\glossentrysymbol{#1})}{}%
}
\newcommand*{\glstopicTitleFont}[1]{\textbf{\large #1}}
\newcommand*{\glstopicDesc}[1]{\Glossentrydesc{#1}\glspostdescription}
\newcommand*{\glstopicLoc}[2]{}
\newlength\glstopicParIndent
\setlength\glstopicParIndent{20pt}
\newlength\glstopicSubIndent
\setlength\glstopicSubIndent{20pt}
\newcommand{\glstopicInit}{}
\newcommand*{\glstopicAssignSubIndent}[1]{%
  \par
  \parindent\dimexpr#1\glstopicSubIndent-\glstopicSubIndent\relax
  \glstopicAssignWidest{#1}%
  \glstopicsubitemhangindent=\dimexpr\parindent+\glstopicwidest\relax
  \hangindent\glstopicsubitemhangindent\relax
  \everypar{\hangindent\glstopicsubitemhangindent\relax
    \parindent\dimexpr\glstopicSubItemParIndent+\glstopicsubitemhangindent\relax}%
}
\newlength\glstopicsubitemhangindent
\newlength\glstopicSubItemParIndent
\glstopicSubItemParIndent\parindent
\newlength\glstopicwidest
\newcommand*{\glstopicAssignWidest}[1]{%
  \ifcsundef{@glswidestlength\romannumeral#1}%
  {%
    \ifcsdef{@glswidestname\romannumeral#1}%
    {%
      \settowidth{\glstopicwidest}{%
       \glstopicSubNameFont{\csuse{@glswidestname\romannumeral#1}}%
```

```
        \glstopicSubItemSep
      }%
    }%
    {\setlength{\glstopicwidest}{0pt}}%
    \csedef{@glswidestlength\romannumeral#1}{\the\glstopicwidest}%
  }%
  {\setlength{\glstopicwidest}{\csuse{@glswidestlength\romannumeral#1}}}%
}
\newcommand*{\glstopicPreSkip}{\medskip}
\newcommand*{\glstopicMidSkip}{\smallskip}
\newcommand*{\glstopicPostSkip}{\smallskip}
\newcommand*{\glstopicSubItem}[3]{%
  \glstopicSubItemBox{#1}{\glstopicSubNameFont{\glsentryitem{#2}%
    \glstarget{#2}{\glossentryname{#2}}}%
    \glstopicSubItemSep
  }%
  \ifglshassymbol{#2}{(\glossentrysymbol{#2})\space}{}%
  \ifglshasdesc{#2}%
   {\glossentrydesc{#2}\glspostdescription\glstopicSubPreLocSep}{}%
  \glstopicSubLoc{#2}{#3}%
}
\newcommand*{\glstopicSubItemSep}{\quad}
\newcommand*{\glstopicSubItemBox}[2]{%
  \ifdim\glstopicwidest>0pt\relax\makebox[\glstopicwidest][l]{#2}\else#2\fi
}
\newcommand*{\glstopicSubNameFont}[1]{\textbf{#1}}
\newcommand*{\glstopicSubPreLocSep}{\space}
\newcommand*{\glstopicSubLoc}[2]{#2}
\newcommand*{\glstopicCols}{2}
\newcommand*{\glstopicColsEnv}{multicols}
\newglossarystyle{topicmcols}{%
  \renewenvironment{theglossary}%
  {%
    \glstopicInit
    \def\glstopic@prechildren{}%
    \def\glstopic@postchildren{}%
    \def\glstopic@prevlevel{-1}%
  }%
  {%
    \ifnum\glstopic@prevlevel>0\relax\glstopic@postchildren\fi
    \par
  }%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
    \ifnum\glstopic@prevlevel>0\relax\glstopic@postchildren\fi
    \def\glstopic@prevlevel{-1}%
    \glstopicGroupHeading{##1}%
  }%
  \renewcommand{\glossentry}[2]{%
    \ifnum\glstopic@prevlevel>0\relax\glstopic@postchildren\fi
```

```
        \def\glstopic@prevlevel{0}%
        \hangindent0pt\relax
        \parindent\glstopicParIndent\relax
        \glstopicItem{##1}{##2}%
        \ifnum\glstopicCols>1\relax
          \ifglshasdesc{##1}%
          {%
            \edef\glstopic@prechildren{%
              \noexpand\begin{\glstopicColsEnv}{\glstopicCols}%
            }%
          }%
          {%
            \edef\glstopic@prechildren{%
              \noexpand\nopagebreak
              \noexpand\begin{\glstopicColsEnv}{\glstopicCols}%
            }%
          }%
          \edef\glstopic@postchildren{\noexpand\end{\glstopicColsEnv}}%
        \fi
      }%
      \renewcommand{\subglossentry}[3]{%
        \ifnum\glstopic@prevlevel=0\relax\glstopic@prechildren\fi
        \def\glstopic@prevlevel{##1}%
        \glstopicAssignSubIndent{##1}%
        \glstopicSubItem{##1}{##2}{##3}%
      }%
      \renewcommand*{\glsgroupskip}{}%
  }
```

Change History

1216

1217

1222

1227

1228

1235

1247

1248

1249

1255