

glossaries-extra.sty v1.49: documented code

Nicola L.C. Talbot

Dickimaw Books

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

2022-10-14

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

Contents

1	Main Package Code (<code>glossaries-extra.sty</code>)	2
1.1	Package Initialisation and Options	2
1.2	Extra Utilities	38
1.3	Modifications to Commands Provided by <code>glossaries</code>	55
1.3.1	Existence Checks	62
1.3.2	Document Definitions	76
1.3.3	Existing Glossary Style Modifications	83
1.3.4	Entry Formatting, Hyperlinks and Indexing	87
1.3.5	Entry Counting	153
1.3.6	Acronym Modifications	170
1.3.7	Indexing and Displaying Glossaries	174
1.4	Link Counting	225
1.5	Integration with <code>glossaries-accsupp</code>	227
1.6	Categories	282
1.7	Abbreviations	311
1.7.1	Abbreviation Styles Setup	339
1.7.2	Predefined Styles	345
1.8	Using Entries in Headings	345
1.9	Prefixes	368
1.10	Multi (Combined/Compound) Entries	374
1.11	Multi-Lingual Support	422
2	Predefined Abbreviation Styles (<code>glossaries-extra-abbrstyles.def</code>)	423
2.1	Predefined Styles (Default Font)	442
2.2	Predefined Styles (Small Capitals)	463
2.3	Predefined Styles (Fake Small Capitals)	481

2.4	Predefined Styles (Emphasized)	498
2.5	Predefined Styles (User Parentheses Hook)	525
2.6	Predefined Styles (Hyphen)	538
2.7	Predefined Styles (No Short on First Use)	577
3	Commands Specific to bib2gls (glossaries-extra-bib2gls.sty)	582
4	Style Adjustments (glossaries-extra-stylemods.sty)	642
4.1	Package Initialisation	642
4.2	List-Like Styles	643
4.3	Longtable Styles	647
4.4	Long Ragged Styles	649
4.5	Supertabular Styles	651
4.6	Super Ragged Styles	654
4.7	Inline Style	656
4.8	Tree Styles	656
4.9	Multicolumn Styles	680
5	bookindex style (glossary-bookindex.sty)	689
6	longextra styles (glossary-longextra.sty)	697
7	topic styles (glossary-topic.sty)	729
8	table styles (glossary-table.sty)	735
9	Rollback Files	762
9.1	Rollback v1.48 (glossaries-extra-2021-11-22.sty)	762
9.2	Rollback v1.48 (glossaries-extra-bib2gls-2021-11-22.sty)	1064
9.3	Rollback v1.48 (glossaries-extra-stylemods-2021-11-22.sty)	1099
9.4	Rollback v1.48 (glossary-bookindex-2021-11-22.sty)	1129
9.5	Rollback v1.48 (glossary-longextra-2021-11-22.sty)	1133
9.6	Rollback v1.48 (glossary-topic-2021-11-22.sty)	1151

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{v1.49}{2022-10-14}

Declare package:
\ProvidesPackage{glossaries-extra}[2022/10/14 v1.49 (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{\glstr@dooption}[1]{\setupglossaries{#1}}%
\let\glstr@declareoption\@gls@declareoption
}
{%
```

Not already loaded, so pass options to `glossaries`.

```
\newcommand{\glstr@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*{\glstr@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
}
```

Declare package options.

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

```
\newcommand*{\glstrundefaction}[2]{%
\glstrundeftag\PackageError{glossaries-extra}{#1}{#2}%
}
```

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

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

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

```
\newcommand*{\glstrundeftag}{??}
\newcommand*{\glstrundeftag}{}
```

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

```

\@glxtr@warn@undefaction This is how \glxtrundefaction should behave if undefaction=warn is set.
    \newcommand*{\@glxtr@warn@undefaction}[2]{%
        \@glxtrundeftag\GlossariesExtraWarning{#1}%
    }

\@glxtr@err@undefaction This is how \glxtrundefaction should behave if undefaction=error is set.
    \newcommand*{\@glxtr@err@undefaction}[2]{%
        \@glxtrundeftag\PackageError{glossaries-extra}{#1}{#2}%
    }

\@glxtr@warn@onexistsordo This is how \glxtr@warnonexistsordo should behave if undefaction=warn is
set.
    \newcommand*{\@glxtr@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.)}%
    }

\@glxtr@redef@forlentries
    \newcommand*{\@glxtr@redef@forlentries}{}

\@glxtr@do@redef@forlentries
    \newcommand*{\@glxtr@do@redef@forlentries}{%
        \renewcommand*{\forlentries}[3][\glstypetype]{%
            \protected@edef\@glo@list{\csname glo@list@##1\endcsname}%
            \ifdefstring{\@glo@list}{,}%
            {%
                \GlossariesExtraWarning{No entries defined in glossary '##1'}%
            }%
            {%
                \@for##2:=\@glo@list\do
                {%
                    \ifdefempty{##2}{\@glo@list}%
                }%
            }%
        }%
    }%

undefaction
    \define@choicekey{glossaries-extra.sty}{undefaction}%
    [\@glxtr@undefaction@val\@glxtr@undefaction@nr]%
    {warn,error}%
    {%
        \ifcase\@glxtr@undefaction@nr\relax
            \let\glxtrundefaction\@glxtr@warn@undefaction
            \let\glxtr@warnonexistsordo\@glxtr@warn@onexistsordo
            \let\@glxtr@redef@forlentries\@glxtr@do@redef@forlentries
        \or

```

```

\let\glxtrundefaction\@glxtr@err@undefaction
\let\glxtr@warnonexistssordo\@gobble
\let\@glxtr@redef@forglsentries\relax
\fi
}

```

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

```

\@glxtr@record Does nothing by default.
\newcommand*{\@glxtr@record}[3]{}

```

```

\glxtr@recordsee Does nothing by default.
\newcommand*{\glxtr@recordsee}[2]{}

```

```

\@glxtr@defaultnumberformat
\newcommand*{\@glxtr@defaultnumberformat}{\glsnnumberformat}%

\GlsXtrSetDefaultNumberFormat
\newcommand*{\GlsXtrSetDefaultNumberFormat}[1]{%
  \renewcommand*{\@glxtr@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.

```

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

```

```

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

    \protected@edef\@gls@label{\glsetoklabel{#1}}%
    \let\glslabel\@gls@label
    \glswriteentry{#1}%
  {%
    \ifdefempty{\@glxtr@thevalue}%
    {%
      \ifx\@glxtr@org@theHvalue\@glxtr@theHvalue
      \else
        \let\theHglentrycounter\@glxtr@theHvalue
      \fi
    }
  }
}

```

```

\glxtr@saveentrycounter
\let\@@do@wrglossary\glxtr@dorecord
}%
{%
\let\theglentrycounter\glxtr@thevalue
\let\theHglentrycounter\glxtr@theHvalue
\let\@@do@wrglossary\glxtr@dorecordnodefer
}%
\ifx\glxtr@record@setting\glxtr@record@setting@alsoindex
\glxtr@@do@wrglossary{#1}%
\else
\@@glxtrwrglossmark

```

Increment associated counter.

```

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

```

`\glxtr@do@alsoindex@wrglossary` The `record=alsoindex` option needs to both record and index.

```

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

```

`\@@glxtr@record` The `record=only` option sets `\glxtr@record` to this. This performs the recording 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*{\@@glxtr@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{\glstoklabel{#2}}%
\let\glslabel\gls@label
\ifglentryexists{#2}{%
{%
\@@glxtrwrglossmark
\begingroup
\let\glsnumberformat\glxtr@defaultnumberformat
\def\glxtr@thevalue{%
\def\glxtr@theHvalue{\glxtr@thevalue}%
\let\glxtr@org@theHvalue\glxtr@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@glstr@equations
  \@glstr@use@equation@counter
\fi
```

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

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

Implement any pre-key settings.

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

Assign keys.

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

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

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

Check post-key hook.

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

Increment associated counter.

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

Check if `noindex` option has been used.

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

Check if `thevalue` has been set.

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

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

```
\ifx\@glstr@org@theHvalue\@glstr@theHvalue
\else
  \let\theHglentrycounter\@glstr@theHvalue
\fi
```

Save the entry counter.

```
\glstr@saveentrycounter
```

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

```
\let\@do@wrglossary\@glstr@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\@glxtr@thevalue
\let\theHglentrycounter\@glxtr@theHvalue
\let\@do@wrglossary\@glxtr@dorecordnodefer
}%
\ifx\@glxtr@record@setting\@glxtr@record@setting@alsoindex
\glxtr@do@wrglossary{#2}%
\else

```

No need to escape special characters.

```

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

```

\@glxtr@glslink@prekeys

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

\@glxtr@glslink@postkeys

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

\@glxtr@glossadd@prekeys

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

\@glxtr@glossadd@postkeys

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

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

```

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

```

\@@glxtr@dorecord

```

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

```

Protect against non-expandable commands occurring in the location.

```

\protected@edef\@glxtr@theentrycounter{\theglsentrycounter}%
\protected@edef\@glxtr@theHentrycounter{\theHglentrycounter}%
\@onelevel@sanitize\@glxtr@theentrycounter

```



```

\@onelevel@sanitize\@glxtr@theHentrycounter
\@xp@glx@getcounterprefix
{\@glxtr@theentrycounter}{\@glxtr@theHentrycounter}%
\fi

```

Don't protect the `\@glxrecordloc` 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\@glxtr@record@setting\@glxtr@record@setting@nameref
#3%
{\#1}{\@glo@counterprefix}{\@glx@counter}{\@glxnumberformat}%
{\@glxrecordloc}%
\else
\@bibglx@write@aux{\string#2%
{\#1}{\@glo@counterprefix}{\@glx@counter}{\@glxnumberformat}%
{\@glxrecordloc}}%
\fi
\@glxtr@counterrecordhook
\let\@glo@counterprefix\@glxtr@orgprefix
}

```

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

```

\newcommand*\@glxtr@dorecordnodefer{%
\ifx\theglentrycounter\theHglentrycounter
\ifx\@glxtr@record@setting\@glxtr@record@setting@nameref
\@glxtr@do@nameref@record
{\@glx@label}{\@glx@counter}{\@glxnumberformat}%
{\theglentrycounter}%
\else
\@bibglx@write@aux{\string\glxtr@record
{\@glx@label}{\@glx@counter}{\@glxnumberformat}%
{\theglentrycounter}}%
\fi
\else
\@xp@glx@getcounterprefix{\theglentrycounter}{\theHglentrycounter}%
\ifx\@glxtr@record@setting\@glxtr@record@setting@nameref
\@glxtr@do@nameref@record
{\@glx@label}{\@glo@counterprefix}{\@glx@counter}%
{\@glxnumberformat}{\theglentrycounter}%
\else
\@bibglx@write@aux{\string\glxtr@record
{\@glx@label}{\@glo@counterprefix}{\@glx@counter}{\@glxnumberformat}%
{\theglentrycounter}}%
\fi
\fi
\@glxtr@counterrecordhook
}

```

`\@glxtr@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{\@glxtr@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
}
```

`\@glxtr@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 `\theHglentrycounter`, which is useful if it can't be obtained by prefixing `\theHglentrycounter`.

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

`\@@glxtr@recordcounter`

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

`\@glxtr@noop@recordcounter`

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

```

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

\@glxtr@recordsee Deal with \glssee in record mode. (This doesn't increment the associated
counter.)
    \newcommand*{\@glxtr@recordsee}[2]{%
        \@glxtrwrglossmark
        \def\@gls@xref{#2}%
        \@onelevel@sanitize\@gls@xref
        \@bibgls@write@aux{}\string\@glxtr@recordsee{#1}{\@gls@xref}}%
    }

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

\glxtr@setup@record Initialise.
    \newcommand*{\glxtr@setup@record}{\let\@do@wrglossary\glxtr@do@wrglossary}

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

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

    The loclist field is just a comma-separated list. The location field is the format-
    ted list.
    \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}%
}%
{}%

```

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 include `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
}

```

```

\@glxtr@record@setting@off
\newcommand*{\@glxtr@record@setting@off}{off}

tr@warn@hybrid@noprintgloss Used by hybrid method if \printglossary isn't used.
\newcommand\@glxtr@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=\@glxtr@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}%
  }%
}

\@glxtr@record@only@setup Initialisation code for record=only and record=nameref
\newcommand*{\@glxtr@record@only@setup}{%
  \def\glxtr@setup@record{%
    \@glxtr@autoseeindexfalse
    \let\@do@seeglossary\@glxtr@recordsee
    \let\@glxtr@record\@glxtr@record
    \let\@do@wrglossary\@glxtr@do@record@wrglossary
    \let\@gls@saveentrycounter\relax
    \let\glxtrundefaction\@glxtr@warn@undefaction
    \let\glxtr@warnonexistssordo\@glxtr@warn@onexistssordo
    \glxtr@addloclistfield
    \renewcommand*{\@glxtr@autoindexcrossrefs}{}%
    \let\@glxtr@recordcounter\@glxtr@op@recordcounter
    \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
  }
  Switch off the index suppression for aliased entries. (bib2gls will deal with
  them.)
  \def\glxtrsetaliasnoindex{}%
  \@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\glxtrNoGlossaryWarning{\@glxtr@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}
[\@glxtr@record@setting\glxtr@record@nr]%
{off,only,alsoindex,nameref,hybrid}%
[only]%
{%
  \ifcase\glxtr@record@nr\relax
```

Don't record.

```
\def\glxtr@setup@record{%
  \renewcommand*{\@do@seeglossary}{\@glxtr@doseeglossary}%
  \renewcommand*{\@glxtr@record}[3]{}%
  \let\@do@wrglossary\glxtr@do@do@wrglossary
  \let\@glxtr@saveentrycounter\glxtr@indexonly@saveentrycounter
  \let\glxtrundefaction\glxtr@err@undefaction
  \let\glxtr@warnonexistssordo\@gobble
  \let\@glxtr@recordcounter\glxtr@noop@recordcounter
  \def\printunsrtglossaryunit{\print@noop@unsrtglossaryunit}%
  \undef\glxtrsetaliasnoindex
}%
\or
```

Only record (don't index).

```
\@glxtr@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\glxtr@setup@record{%
  \renewcommand*{\@glxtr@record@setting@alsoindex}{alsoindex}%
  \renewcommand*{\@do@seeglossary}{\@glxtr@dosee@alsoindex@glossary}%
  \let\@glxtr@record\@glxtr@record
  \let\@do@wrglossary\glxtr@do@alsoindex@wrglossary
  \let\@glxtr@saveentrycounter\glxtr@indexonly@saveentrycounter
  \let\glxtrundefaction\glxtr@warn@undefaction
  \let\glxtr@warnonexistssordo\glxtr@warn@onexistssordo
  \glxtr@addloclistfield
  \let\@glxtr@recordcounter\glxtr@op@recordcounter
  \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
  \undef\glxtrsetaliasnoindex
}%
\or
```

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

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

```

    the document doesn't support hyperlinks}}%
  {}%

```

```

\or

```

Hybrid record (use bib2gls to fetch definitions) and index (use makeindex/xindy to sort and collate).

```

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

```

bibglsaux Provide an option to put the records in a different aux file that will only be read by bib2gls and not by L^AT_EX. A large number of records in the aux file can slow down the document build as L^AT_EX 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}{%
  \glxtrsetbibglsaux{#1}%
}

```

```

\glxtrsetbibglsaux

```

```

\newcommand{\glxtrsetbibglsaux}[1]{%
  \renewcommand{\@glxtr@setup@bibglsaux}{\@glxtr@setup@bibglsaux{#1}}%
}

```

```

\@glxtr@setup@bibglsaux

```

```

\newcommand{\@glxtr@setup@bibglsaux}{%
  \renewcommand{\glxtrsetbibglsaux}[1]{%
    \@glxtr@setup@bibglsaux{##1}%
  }%
}
\AtBeginDocument{\@glxtr@setup@bibglsaux}

```

```

\@glxtr@setup@bibglsaux

```

```

\newcommand{\@glxtr@setup@bibglsaux}[1]{%
  \ifstrempy{#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.

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

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

Need to provide conditional commands that are backward compatible:

`\if@glxstrdocdef`

```
\newcommand*{\if@glxstrdocdef}{\ifnum\@glxstr@docdefval>0 }
```

`\@glxstrdocdeftrue`

```
\newcommand*{\@glxstrdocdeftrue}{\def\@glxstr@docdefval{1}}
```

`\@glxstrdocdeffalse`

```
\newcommand*{\@glxstrdocdeffalse}{\def\@glxstr@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}
[\@glxstr@docdefsetting\@glxstr@docdefval]%
{false,true,restricted,atom}[true]%
{%
\ifnum\@glxstr@docdefval>1\relax
\renewcommand*{\@glsdofexistsorwarn}{\glsdofexists}%
\else
\renewcommand*{\@glsdofexistsorwarn}{\glsdofexistsorwarn}%
\fi
}
```



```

\if@glxtrdocdefrestricted
    \newcommand*{\if@glxtrdocdefrestricted}{\ifnum\@glxtr@docdefval>1 }

\@glxtrdocdefrestricted 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*{\@glxtrdocdefrestricted}{\@glxtrdocdefrestricted}

indexcrossrefs Automatically index cross references at the end of the document
    \define@boolkey{glossaries-extra.sty}[\@glxtr]{indexcrossrefs}[true]{%
        \if@glxtrindexcrossrefs
        \else
        \renewcommand*{\@glxtr@autoindexcrossrefs}{}%
        \fi
    }

Switch off since this can increase the build time.
    \@glxtrindexcrossrefsfalse

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

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

autoseeindex Provide a boolean option to allow the user to prevent the automatic indexing
of the cross-referencing keys see, seealso and alias.
    \define@boolkey{glossaries-extra.sty}[\@glxtr]{autoseeindex}[true]{%
    }
    \@glxtrautoseeindextrue

equations Provide a boolean option to automatically switch to the equation counter when
in a numbered maths environment.
    \define@boolkey{glossaries-extra.sty}[\@glxtr]{equations}[true]{%
    }
    \@glxtr@equationsfalse

\glxtr@float
    \let\glxtr@float\@float

\glxtr@dblfloat
    \let\glxtr@dblfloat\@dblfloat

floats Provide a boolean option to automatically switch to the the corresponding
counter when in a float.
    \define@boolkey{glossaries-extra.sty}[\@glxtr]{floats}[true]{%
        \if@glxtrfloats
        \renewcommand*{\@float}[1]{\renewcommand{\glscounter}{##1}\glxtr@float{##1}}%
        \renewcommand*{\@dblfloat}[1]{\renewcommand{\glscounter}{##1}\glxtr@dblfloat{##1}}%
        \else

```

```

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

\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@doption{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@doption{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@doption{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@doption{nopostdot=false}%
\ifstrequal{#1}{dot}%
{%
\renewcommand*{\@glsxtr@defpostpunc}{%

```

```

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

```

`\glstrabbrvtype` Glossary type for abbreviations.

```
\newcommand*{\glstrabbrvtype}{\gldefaulttype}
```

`\@glstr@abbreviationsdef` Set by abbreviations option.

```
\newcommand*{\@glstr@abbreviationsdef}{}%
```

`\@glstr@doabbreviationsdef`

```

\newcommand*{\@glstr@doabbreviationsdef}{%
\ifpackageloaded{babel}%
{\providecommand{\abbreviationsname}{\acronymname}}%
{\providecommand{\abbreviationsname}{Abbreviations}}%
\newglossary[glg-abr]{abbreviations}{gls-abr}{glo-abr}{\abbreviationsname}%
\renewcommand*{\glstrabbrvtype}{abbreviations}%
\newcommand*{\printabbreviations}[1][1]{%
\printglossary[type=\glstrabbrvtype,##1]%
}%
\disable@keys{glossaries-extra.sty}{abbreviations}%

```

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

```

\ifglsacronym
\else
\renewcommand*{\acronymtype}{\glstrabbrvtype}%

```

```

\fi
}%

```

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

```

\@glstr@declareoption{abbreviations}{%
\let\@glstr@abbreviationsdef\@glstr@doabbreviationsdef
}

```

\shortcut@gls

```

\newcommand{\shortcut@gls}{\cglgs}

```

\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}

```

DefineAbbreviationShortcuts 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}{\glstrshort}%
\newcommand*{\asp}{\glstrshortpl}%
\newcommand*{\al}{\glstrlong}%
\newcommand*{\alp}{\glstrlongpl}%
\newcommand*{\af}{\glstrfull}%
\newcommand*{\afp}{\glstrfullpl}%
\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.)

`\@glxtr@setupshortcuts` Command used to set the shortcuts option.

```

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

```

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

```
\newcommand*{\@glxtr@shortcutsval}{\ifglxtrshortcutsacro\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}%
[\@glxtr@shortcutsval\@glxtr@shortcutsnr]%
{acronyms,acro,abbreviations,abbr,other,all,true,ac,acother,abother,none,false}[true]{%
\ifcase\@glxtr@shortcutsnr\relax % acronyms
\renewcommand*{\@glxtr@setupshortcuts}{%
\glsacrshortcutstrue
\DefineAcronymSynonyms
}%
\or % acro
\renewcommand*{\@glxtr@setupshortcuts}{%
\glsacrshortcutstrue
\DefineAcronymSynonyms
}%
\or % abbreviations
\renewcommand*{\@glxtr@setupshortcuts}{%
\GlsXtrDefineAbbreviationShortcuts
}%
\or % abbr
\renewcommand*{\@glxtr@setupshortcuts}{%
\GlsXtrDefineAbbreviationShortcuts
}%
\or % other
\renewcommand*{\@glxtr@setupshortcuts}{%
\GlsXtrDefineOtherShortcuts
}%
\or % all
\renewcommand*{\@glxtr@setupshortcuts}{%
\glsacrshortcutstrue

\GlsXtrDefineAcShortcuts
\GlsXtrDefineAbbreviationShortcuts
\GlsXtrDefineOtherShortcuts
}%
\or % true
\renewcommand*{\@glxtr@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}
[\@glstr@nomissingglstextval\@glstr@nomissingglstextnr]%
{true,false}[true]{%
  \ifcase\@glstr@nomissingglstextnr\relax % true
    \renewcommand{\glstrNoGlossaryWarning}[1]{\null}%
  \else % false
    \renewcommand{\glstrNoGlossaryWarning}[1]{%
      \@glstr@defaultnoglossarywarning{#1}%
    }%
  \fi
}
```

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

`\@glstr@redefstyles`

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

`stylemods`

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

```

`\GlsXtrInternalLocationHyperlink`

The first two arguments are always control sequences.

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

```

`\wrglossary@locationhyperlink`

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

```

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.

```
\@glstr@declareoption{indexcounter}{%
\glstr@doption{counter=wr glossary}%
\ifundef\c@wr glossary
{%
\newcounter{wr glossary}%
\renewcommand{\thewr glossary}{\arabic{wr glossary}}%
}%
}%
\renewcommand*{\glstr@inc@wr glossaryctr}[1]{%
```

Only increment if the current counter is wr glossary.

```
\ifdefstring\@glsc@counter{wr glossary}%
{%
\refstepcounter{wr glossary}%
\label{wr glossary.\thewr glossary}%
\@glstrwr glosscountermark{\thewr glossary}%
}%
}%
\renewcommand*{\GlsXtrInternalLocationHyperlink}[3]{%
\ifdefstring\glsc@entrycounter{wr glossary}%
{%
\@glstr@wr glossary@locationhyperlink{##1}{##2}{##3}%
}%
{\glstrhyperlink{##1##2##3}{##3}}%
}%
}
```

`\@glstrwr glossmark` Marks the place where indexing occurs. Does nothing by default.

```
\newcommand*{\@glstrwr glossmark}{}%
```

`\@@glstrwr glossmark` Since `\glscadd` can be used in the preamble, this action needs to be disabled until the start of the document.

```
\newcommand*{\@@glstrwr glossmark}{}%
\AtBeginDocument{\renewcommand*{\@glstrwr glossmark}{\@glstrwr glossmark}}
```

`\glstrwr glossmark`

```
\newcommand*{\glstrwr glossmark}{\ensuremath{\cdot}}
```

`\@glstrwr glosscountermark` Marks the place where wr glossary counter is incremented. Does nothing by default.

```
\newcommand*{\@glstrwr glosscountermark}[1]{}%
```

`\@@glstrwr glosscountermark`

```
\newcommand*{\@@glstrwr glosscountermark}[1]{}%
\AtBeginDocument{\renewcommand*{\@glstrwr glosscountermark}{\@glstrwr glosscountermark}}
```

```

\glxtrwrglosscountermark
    \newcommand*{\glxtrwrglosscountermark}[1]{\glsshowtargetfonttext{[#1]}}

\@glxtr@doshowtarget
    \newcommand\@glxtr@doshowtarget[2]{#2}

\glxtrundefdebug Don't do anything until after the document environment has begun.
    \newcommand*{\glxtrundefdebug}[1]{}

\@glxtrundefdebug Use the same font as the targets.
    \newcommand*{\@glxtrundefdebug}[1]{%
        \if@glx@debug \glsshowtargetfonttext{[#1]}\fi
    }

debug Provide extra debug options.
\define@choicekey{glossaries-extra.sty}{debug}
[\@glxtr@debugval\@glxtr@debugnr]%
{true,false,showtargets,showwrgloss,all,showaccsupp}[true]{%
    \ifcase\@glxtr@debugnr\relax % true
        \glxtr@doption{debug=true}%
        \renewcommand*{\@glxtrwrglossmark}{}%
        \renewcommand*{\@glxtrwrglosscountermark}[1]{}%
    \or % false
        \glxtr@doption{debug=false}%
        \renewcommand*{\@glxtrwrglossmark}{}%
        \renewcommand*{\@glxtrwrglosscountermark}[1]{}%
        \let\@glxtr@doshowtarget\@secondoftwo
    \or % showtargets
        \glxtr@doption{debug=showtargets}%
        \def\@glxtr@doshowtarget{\@glxtr@showtargetleft}%
    \or % showwrgloss
        \glxtr@doption{debug=true}%
        \renewcommand*{\@glxtrwrglossmark}{\glxtrwrglossmark}%
        \renewcommand*{\@glxtrwrglosscountermark}{\glxtrwrglosscountermark}%
    \or % all
        \glxtr@doption{debug=true,debug=showaccsupp}%
        % debug=showwrgloss:
        \renewcommand*{\@glxtrwrglossmark}{\glxtrwrglossmark}%
        \renewcommand*{\@glxtrwrglosscountermark}{\glxtrwrglosscountermark}%
        % debug=showtargets:
        \def\@glxtr@doshowtarget{\@glxtr@showtargetleft}%
    \or % showaccsupp
        \glxtr@doption{debug=showaccsupp}%
        \fi
    }

\glxtrshowtargetouter
    \newcommand*{\glxtrshowtargetouter}{\glsshowtargetouter}

```

```

\glxtrshowtargetinner
\newcommand*{\glxtrshowtargetinner}[1]{\glsshowtargetinner{#1}}

    Debugging show targets.

\glxtrshowtargetleft
\newcommand{\glxtrshowtargetleft}[2]{\glsshowtarget{#1}#2\glxtrshowtargetmark}%

\glxtrshowtargetright
\newcommand{\glxtrshowtargetright}[2]{\glxtrshowtargetmark#2\glsshowtarget{#1}}%

\glxtrshowtargetmark
\newcommand{\glxtrshowtargetmark}{}%

```

`showtargets` Implements `debug=showtargets` and provides extra adjustments.

```

\define@choicekey{glossaries-extra.sty}{showtargets}
[\@glxtr@showtargetsval\@glxtr@showtargetsnr]%
{left,right,innerleft,inerright,annoteleft,annoteright}%
{%
  \glxtr@doooption{debug=showtargets}%
  \ifcase\@glxtr@showtargetsnr\relax
    \def\glxtr@doshowtarget{\@glxtrshowtargetleft}%
    \def\glxtrshowtargetouter{\glsshowtargetouter}%
    \def\glxtrshowtargetinner{\glsshowtargetinner}%
    \let\glxtrshowtargetmark\empty
  \or
    \def\glxtr@doshowtarget{\@glxtrshowtargetright}%
    \def\glxtrshowtargetouter{\glsshowtargetouter}%
    \def\glxtrshowtargetinner{\glsshowtargetinner}%
    \let\glxtrshowtargetmark\empty
  \or
    \def\glxtr@doshowtarget{\@glxtrshowtargetleft}%
    \def\glxtrshowtargetouter{\glxtrshowtargetinner}%
    \def\glxtrshowtargetinner{\glsshowtargetinnersymleft}%
    \let\glxtrshowtargetmark\empty
  \or
    \def\glxtr@doshowtarget{\@glxtrshowtargetright}%
    \def\glxtrshowtargetouter{\glxtrshowtargetinner}%
    \def\glxtrshowtargetinner{\glsshowtargetinnersymright}%
    \let\glxtrshowtargetmark\empty
  \or
    \def\glxtr@doshowtarget{\@glxtrshowtargetleft}%
    \def\glxtrshowtargetouter{\glxtrshowtargetinner}%
    \def\glxtrshowtargetinner{\glsshowtargetinnersymleft}%
    \def\glxtrshowtargetmark{\@glsshowtargetmarkfmt\glxtrshowtargetsymbolsymbolright}%
  \or
    \def\glxtr@doshowtarget{\@glxtrshowtargetright}%
    \def\glxtrshowtargetouter{\glxtrshowtargetinner}%
    \def\glxtrshowtargetinner{\glsshowtargetinnersymright}%
  \or

```

```

\def\@glxtrshowtargetmark{\@glsshowtargetmarkfmt\glxtrshowtargetsymbolleft}%
\fi
}

```

Pass all other options to glossaries.

`\glxtr@processunknownoptions` Need to compensate for the problem identified in <https://www.dickimaw-books.com/bugtracker.php?key=171>

```

\newcommand*\@glxtr@processunknownoptions{}

\@ifpackageloaded{glossaries}
{%
  \DeclareOptionX*{%
    \edef\glxtr@processunknownoptions{%
      \noexpand\setupglossaries{\expandonce\CurrentOption}}
  }
  {%
    \DeclareOptionX*{%
      \expandafter\glxtr@doooption\expandafter{\CurrentOption}}
  }
}

```

Process options.

```
\ProcessOptionsX
```

Load glossaries if not already loaded.

```

\RequirePackage{glossaries}
\glxtr@processunknownoptions

```

Load the glossaries-accsupp package if required.

```
\@glxtr@doaccsupp
```

Load the glossaries-prefix package if required.

```
\@glxtr@doloadprefix
```

Redefine `\glspostdescription` if required.

```
\@glxtr@defpostpunc
```

`\glsexindexsetting` 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 `\glsexindexsetting`.

```

\let\@glxtr@org@indexsetting\glsexindexsetting
\providecommand{\glsexindexsetting}{\ifglsexindy xindy\else makeindex\fi}
\ifx\@glxtr@org@indexsetting\glsexindexsetting
  \renewcommand{\glsexindexsetting}{%
    \@glxtr@if@record@only{bib2gls}{\ifglsexindy xindy\else makeindex\fi}%
  }
\else
  \@glxtr@if@record@only{\renewcommand{\glsexindexsetting}{bib2gls}}{}%
\fi

```

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

```

\glsentencecase
    \providecommand{\glsentencecase}[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}}

\glsupercase Not using \unexpanded because ditto the above.
    \providecommand{\glsupercase}[1]{\mfirstucMakeUppercase{#1}}

\@Glsentryfield
    \providecommand{\@Glsentryfield}[2]{%
        \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
        {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
    }

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

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

    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,
    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

```
\@glxtr@saveMakeUppercase
\newcommand{\@glxtr@saveMakeUppercase}{}

\@glxtr@restoreMakeUppercase
\newcommand{\@glxtr@restoreMakeUppercase}{}

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


Finished L^AT_EX3 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{%`

```

\glxtrtitleorpdforheading
{%
  \ifmode
    \nfss@text{\glxtrshowtargetinner{#1}}%
  \else
    \ifinner
      \glxtrshowtargetinner{#1}%
    \else
      \glxtrshowtargetouter{#1}%
    \fi
  \fi
}%
{[#1]}%
{{\protect\glsshowtargetinner{#1}}}%
}

```

`\@glsshowtargetmarkfmt`

```

\newcommand*{\@glsshowtargetmarkfmt}[1]{%
  \glxtrtitleorpdforheading
  {%
    \ifmode \nfss@text{#1}\else #1\fi
  }%
  {}%
  {\ifmode \nfss@text{#1}\else #1\fi}%
}

```

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

```

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

```

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

```

\newcommand*{\@glxtr@doseeglossary}[2]{%
  \glsoifexists{#1}%
  {%
    \@glxtrwrglossmark
    \@glxtr@org@doseeglossary{#1}{#2}%
  }%
}

```

`\@glxtr@dosee@alsoindex@glossary`

```

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

```

`\@glxtr@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\@glxtr@org@gloautosee\@glo@autosee

```

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

```
\if@glxtr@autoseeindex
\else
\ifdef\@glxtr@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@glxtr@autoseeindex\@glxtr@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@glxtr@autoseeindex\@gls@see@noindex\fi
}
```

Define abbreviations glossaries if required.

```
\@glxtr@abbreviationsdef
\let\@glxtr@abbreviationsdef\relax
```

Setup shortcuts if required.

```
\@glxtr@setupshortcuts
```

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

```
\@glxtr@redef@for@gl@sentries
```

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

```
\renewcommand{\@glxtr@doption}[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\@glxtr@setup@record\relax
\let\@glxtr@setupshortcuts\relax
\let\@glxtr@redef@for@gl@sentries\relax
\let\@glxtr@doloadprefix\relax
\setkeys{glossaries-extra.sty}{#1}%
}
```

```

\@glxtr@abbreviationsdef
\let\@glxtr@abbreviationsdef\relax
\@glxtr@setupshortcuts
\glxtr@setup@record
\@glxtr@redef@for@gl@sentries
\@glxtr@doloadprefix
}

\glxtr@org@@do@wrglossary Save original definition of \@do@wrglossary.
\let\glxtr@org@@do@wrglossary\@do@wrglossary

\glxtr@@do@wrglossary The new version adds code that can show a marker for debugging and increments
the associated counter if enabled.
\newcommand*{\glxtr@@do@wrglossary}[1]{%
\@glxtrwrglossmark
\glxtr@inc@wrglossaryctr{#1}%
\glxtr@org@@do@wrglossary{#1}%
}

\glxtr@saveentrycounter Save original definition of \@gls@saveentrycounter.
\let\glxtr@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\glxtr@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{%
\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@gl@getcounterprefix`. Any commands that were using `\gl@getcounterprefix` directly need to be use `\xp@gl@getcounterprefix` instead.

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

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

```
\ifx\@glo@counterprefix\@empty
  \ifx\@gl@xtr@record@setting\@gl@xtr@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\@gl@counter^^Jotherwise you
      will get the warning: “name{\@gl@counter.#1}’ has been^^J
      referenced but does not exist”%
    \ifx\@gl@xtr@record@setting\@gl@xtr@record@setting@only
      . You may want to consider using record=nameref instead%
    \fi}%
  \fi
\fi
\fi
}
```

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

`\@gl@xtr@dialecthook`

```
\newcommand*\@gl@xtr@dialecthook{}
```

Set up record option if required.

```
\gl@xtr@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\gl@xtrundefdebug{\@gl@xtrundefdebug}%
  \def\gl@xtrundeftag{\gl@xtrundeftag}%
}
```

1.2 Extra Utilities

`\GlsXtrIfUnusedOrUndefined{<label>}{<true>}{<false>}`

`\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]{%
  \ifglentryexists{#1}%
  {\ifbool{glo@\glsetoklabel{#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{glo@type@#1@out}{#3}{#2}%
}
```

`\s@ifglossaryexists`

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

}
```

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

`\glxtrifemptyglossary`

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{\glxtrifemptyglossary}[3]{%
  \ifcsdef{glolist@#1}%
  {%
    \ifcsstring{glolist@#1}{,}{#2}{#3}%
  }
```

```

}%
{%
  \glsxtrundefaction{Glossary type ‘#1’ doesn’t exist}{}%
  #2%
}%
}

```

`\GlsXtrIfInGlossary{<label>}{<type>}{<true>}{<false>}`

`\GlsXtrIfInGlossary`

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.

```

\newcommand*{\GlsXtrIfInGlossary}[4]{%
  \ifcsdef{glolist@#2}%
  {%
    \protected@edef\@glsxtr@tmp{#1}%
    \letcs\@glsxtr@tmplist{glolist@#2}%
    \expandafter\DTLifinlist\expandafter{\@glsxtr@tmp}{\@glsxtr@tmplist}%
    {#3}{#4}%
  }%
  {%
    \glsxtrundefaction{Glossary type ‘#1’ doesn’t exist}{}%
    #4%
  }%
}

```

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

```

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

```

`\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}%
}
```

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}
```


`\@glstrfmt` Unstarred form.

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

`\s@glstrfmt` Starred form.

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

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

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

`\@glstrfmt` Actual inner working.

```
\newcommand*{\@glstrfmt}[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}%
\glstoifexistsordo{#2}%
{%
  \ifglshasfield{\GlsXtrFmtField}{#2}%
  {%
    \let\do@glsl@link@checkfirsthyper\relax
    \expandafter\@glsl@link\expandafter[\GlsXtrFmtDefaultOptions,#1]{#2}%
    {\glstrfmtdisplay{\glscurrentfieldvalue}{#3}{#4}}%
  }%
  {\glstrfmtdisplay{@firstofone}{#3}{#4}}%
}%
%
```

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

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

```
\Glsxtrfmt[<options>]{<entry-label>}{<text>}[<insert>]
```

`\Glsxtrfmt`

As `\glstrfmt` 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{\glxtrfmt}{\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}}{}}%
}

\glxtrfmtdisplay The command used internally by \glxtrfmt 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{\glxtrfmtdisplay}[3]{\csuse{#1}{#2}{#3}}

\glxtrentryfmt No link or indexing.
\newcommand*{\glxtrentryfmt}[2]{\%
\glstexorpdfstring{\@glxtrentryfmt{#1}{#2}}{\glxtrpdfentryfmt{#1}{#2}}%
}

\glxtrpdfentryfmt Used for the PDF bookmarks.
\newcommand*{\glxtrpdfentryfmt}[2]{#2}

\@glxtrentryfmt
\newrobustcmd*{\@glxtrentryfmt}[2]{\%
Locally define \glslabel in case the helper command needs to access the label.

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

\Glsxtrentryfmt Sentence-case version.
\newcommand*{\Glsxtrentryfmt}[2]{\%
\glstexorpdfstring
{\@glxtrentryfmt{#1}{\glssentencecase{#2}}}%
{\Glsxtrpdfentryfmt{#1}{#2}}%
}
\glsmfuaddmap{\glxtrentryfmt}{\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}%
}
```

`\glsxtrfieldlistead` Similarly but uses `\listcseadd`.

```
\newcommand*{\glsxtrfieldlistead}[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:

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

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

`\glxtrfieldxifinlist` Expands item.

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

`\glxtrforcsvfield`

`\glxtrforcsvfield{<label>}{<field>}{<cs handler>}`

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

`\@glxtrforcsvfield` Unstarred version.

```
\newcommand*\@glxtrforcsvfield}[3]{%
  \@glxtrifhasfield{#2}{#1}%
  {%
    \let\glxtrendfor\@endfortrue
    \@for\@glxtr@label:=\glscurrentfieldvalue\do
      {\expandafter#3\expandafter{\@glxtr@label}}}%
  }%
```

`\s@glxtrforcsvfield` Starred version.

```
\newcommand*\s@glxtrforcsvfield}[3]{%
  \s@glxtrifhasfield{#2}{#1}%
  {%
    \let\glxtrendfor\@endfortrue
    \@for\@glxtr@label:=\glscurrentfieldvalue\do
      {\expandafter#3\expandafter{\@glxtr@label}}}%
  }%
```

`\glxtrfieldformatcsvlist`

```
\newrobustcmd*\glxtrfieldformatcsvlist}[2]{%
  \@glxtrifhasfield{#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}%
}
```

\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>}
```

\GlsXtrIfValueInFieldCsvList

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
}
```

\GlsXtrIfValueInFieldCsvList Unstarred version.

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

`\GlsXtrIfValueInFieldCsvList` Unstarred version.

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

`\xGlsXtrIfValueInFieldCsvList{<label>}{<field>}{<value>}`
`{<true>}{<false>}`

`\GlsXtrIfValueInFieldCsvList`

As above but fully expand `<value>`.

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

`\GlsXtrIfValueInFieldCsvList` Unstarred version.

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

`\GlsXtrIfValueInFieldCsvList` Unstarred version.

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

`\glstrifhasfield`

`\glstrifhasfield{<field>}{<label>}{<true>}{<false>}`

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{\glstrifhasfield}{%
\@ifstar{\s@glstrifhasfield}{\@glstrifhasfield}%
}
```

`\@glxtrifhasfield` Unstarred version adds grouping.

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

`\s@glxtrifhasfield` Starred version omits grouping.

```
\newcommand{\s@glxtrifhasfield}[4]{%
  \letcs{\glscurrentfieldvalue}{glo@\glsetoklabel{#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}%
}
```

```
\GlsXtrIfFieldEqNum{<field>}{<label>}{<value>}{<true>}
{<false>}
```

`\GlsXtrIfFieldEqNum`

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{<field>}{<label>}{<comparison>}{<value>}{<true>}{<false>}
```

\GlsXtrIfFieldCmpNum

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{<field>}{<label>}{<true>}{<false>}
```

\GlsXtrIfFieldUndef

Just uses \ifcsundef.

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

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

```
\newcommand*\glstrusefield[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]{%
  \glstrsetfieldifexists{#1}{#2}%
  {\cslet{glo@\glstetoklabel{#1}@#2}{#3}}%
}

```

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

```

\newrobustcmd*{\csGlsXtrLetField}[3]{%
  \glstrsetfieldifexists{#1}{#2}%
  {\csletcs{glo@\glstetoklabel{#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]{%
  \glstrsetfieldifexists{#1}{#2}%
  {\csletcs{glo@\glstetoklabel{#1}@#2}{glo@\glstetoklabel{#3}@#4}}%
}

```

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

```

\newrobustcmd*{\glsXtrSetField}[3]{%
  \glstrsetfieldifexists{#1}{#2}%
  {\csgdef{glo@\glstetoklabel{#1}@#2}{#3}}%
}

```

`\xGlsXtrSetField`

```

\newrobustcmd*{\xGlsXtrSetField}[3]{%
  \glstrsetfieldifexists{#1}{#2}%
  {\protected@csxdef{glo@\glstetoklabel{#1}@#2}{#3}}%
}

```

`\eGlsXtrSetField`

```

\newrobustcmd*{\eGlsXtrSetField}[3]{%
  \glstrsetfieldifexists{#1}{#2}%
  {\protected@csedef{glo@\glstetoklabel{#1}@#2}{#3}}%
}

```

`\GlsXtrIfFieldEqStr` Starred version uses starred version of `\glstrifhasfield` (that is, no grouping).

```

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

```

`\@GlsXtrIfFieldEqStr`

```

\newrobustcmd*{\@GlsXtrIfFieldEqStr}[5]{%
  \glstrifhasfield{#1}{#2}%
  {%

```

```

        \ifdefstring{\glscurrentfieldvalue}{#3}{#4}{#5}%
    }%
    {#5}%
}

```

`\s@GlsXtrIfFieldEqStr`

```

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

```

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

```

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

```

`\@GlsXtrIfFieldEqXpStr`

```

\newrobustcmd*{\@GlsXtrIfFieldEqXpStr}[5]{%
    \@glstrifhasfield{#1}{#2}%
    {%
        \protected@edef\@glstmp{#3}%
        \ifdefequal{\glscurrentfieldvalue}{\@glstmp}{#4}{#5}%
    }%
    {#5}%
}

```

`\s@GlsXtrIfFieldEqXpStr`

```

\newrobustcmd*{\s@GlsXtrIfFieldEqXpStr}[5]{%
    \s@glstrifhasfield{#1}{#2}%
    {%
        \protected@edef\@glstmp{#3}%
        \ifdefequal{\glscurrentfieldvalue}{\@glstmp}{#4}{#5}%
    }%
    {#5}%
}

```

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

```

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

```

`\@GlsXtrIfXpFieldEqXpStr`

```

\newrobustcmd*{\@GlsXtrIfXpFieldEqXpStr}[5]{%
    \@glstrifhasfield{#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@glxtrifhasfield{#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 `\glxtrifhasfield`, save and restore `\glscurrentfieldvalue`.

```

\let\@glxtr@org@currentfieldvalue\glscurrentfieldvalue
\glxtrifhasfield{\GlsXtrForeignTextField}{#1}%
{%
  \expandafter\GetTrackedDialectFromLanguageTag\expandafter
  {\glscurrentfieldvalue}{\@glxtr@dialect}%
  \let\@glxtr@locale\glscurrentfieldvalue
  \let\glscurrentfieldvalue\@glxtr@org@currentfieldvalue
  \ifdefempty\@glxtr@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\@glstr@dialect\TrackedDialectClosestSubMatch}%
}%
{}%
\ifdefempty\@glstr@dialect
{%
No tracked dialect found for the root language.
}%
{}%
Check if there's a caption hook for the given dialect label.
\ifcsundef{captions\@glstr@dialect}{}%
{%
Dialect label not recognised. Check if there's a known mapping.
\IfTrackedDialectHasMapping{\@glstr@dialect}%
{%
    \edef\@glstr@dialect{%
        \GetTrackedDialectToMapping{\@glstr@dialect}}%
Does a caption hook exist for this?
\ifcsundef{captions\@glstr@dialect}{}%
{%
No mapping. Try root language label instead.
\ifcsundef{captions\@tracklang@lang}{}%
{%
    \let\@glstr@dialect\@tracklang@lang
}%
}%
}%
{}%
No mapping. Try root language label instead.
\ifcsundef{captions\@tracklang@lang}{}%
{%
    \let\@glstr@dialect\@tracklang@lang
}%
}%
}%
\ifdefempty\@glstr@dialect
{%
    \GlsXtrUnknownDialectWarning{\@glstr@locale}{\@tracklang@lang}%
    #2%
}%
    {\foreignlanguage{\@glstr@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)}%
}

```

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

```

```

        \gls{#1}%
      \fi
    \fi
  }
}%

\apptoglossarypreamble
\newcommand{\apptoglossarypreamble}[2][\glsdefaultttype]{%
  \ifcsdef{glolist@#1}%
  {%
    \ifcsundef{@glossarypreamble@#1}%
    {\csdef{@glossarypreamble@#1}{}}%
    {}%
    \csappto{@glossarypreamble@#1}{#2}%
  }%
  {%
    \GlossariesExtraWarning{Glossary ‘#1’ is not defined}%
  }%
}

```

```

\pretoglossarypreamble
\newcommand{\preglossarypreamble}[2][\glsdefaultttype]{%
  \ifcsdef{glolist@#1}%
  {%
    \ifcsundef{@glossarypreamble@#1}%
    {\csdef{@glossarypreamble@#1}{}}%
    {}%
    \cspreto{@glossarypreamble@#1}{#2}%
  }%
  {%
    \GlossariesExtraWarning{Glossary ‘#1’ is not defined}%
  }%
}

```

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.

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

```

\renewcommand*{\@pglossarysection}[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`

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

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}}}%
}%
}%
{%
    \glusifcategoryattributehasitem{\@glo@category}{encapinnerfmt}{#2}%
    {%
        \csxdef{glo@#1@#2}{\noexpand\glstrgenentrytextfmt{\expandonce{#3}}}%
        \glsexclapplyinnerfmtfield{#1}{#2}%
    }%
    {%
        \expandafter\global\expandafter\let\csname glo@#1@#2\endcsname#3%
    }%
}%
}%
}

```

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

```

\renewcommand{\@@gls@expand@field}[3]{%
    \glusifcategoryattributehasitem{\@glo@category}{encapnocaseinnerfmt}{#2}%
    {%
        \protected@csxdef{glo@#1@#2}{\noexpand\NoCaseChange
            {\noexpand\glstrgenentrytextfmt{#3}}}%
        \glsexclapplyinnerfmtfield{#1}{#2}%
    }%
    {%
        \glusifcategoryattributehasitem{\@glo@category}{encapnocase}{#2}%
        {%
            \glusifcategoryattributehasitem{\@glo@category}{encapinnerfmt}{#2}%
            {%
                \protected@csxdef{glo@#1@#2}{\noexpand\NoCaseChange
                    {\noexpand\glstrgenentrytextfmt{#3}}}%
                \glsexclapplyinnerfmtfield{#1}{#2}%
            }%
            {%
                \protected@csxdef{glo@#1@#2}{\noexpand\NoCaseChange{#3}}%
            }%
        }%
        {%
            \glusifcategoryattributehasitem{\@glo@category}{encapinnerfmt}{#2}%
            {%
                \protected@csxdef{glo@#1@#2}{\noexpand\glstrgenentrytextfmt{#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 `\glstrpostlongdescription` instead.

```

\longnewglossaryentry
\renewcommand*{\longnewglossaryentry}{%
  \ifstar\@glstr@s@longnewglossaryentry\@glstr@longnewglossaryentry
}

```

`\@glstr@s@longnewglossaryentry` Starred version.

```

\newcommand{\@glstr@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
}%
}

```

`\@glstr@longnewglossaryentry` Unstarred version.

```

\newcommand{\@glstr@longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\glstrpostlongdescription}%
      \@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
    }%
}

\glxtrpostlongdescription Hook at the end of the description when using the unstarred \longnewglossaryentry.
\newcommand*{\glxtrpostlongdescription}{\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\glxtr@s@newignoredglossary\glxtr@org@newignoredglossary
}

\glxtr@org@newignoredglossary The original definition is patched to check for existence.
\newcommand*{\glxtr@org@newignoredglossary}[1]{%
\ifcsdef{glolist@#1}
{%
\glxtrundefaction{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}%
}%
}%
}

\glxtr@s@newignoredglossary Starred form.
\newcommand*{\glxtr@s@newignoredglossary}[1]{%
\ifcsdef{glolist@#1}

```

```

{%
  \glxstrundefaction{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}%
  {%
    \defglssentryfmt[#1]{\glssentryfmt}%
  }%
  {}%
}%
}

```

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

```

\glusifusetranslator
{%
  \renewcommand*{\glssettocitle}[1]{%
    \ifcsdef{gls@tr@set@#1@tocitle}%
    {%
      \csuse{gls@tr@set@#1@tocitle}%
    }%
    {%
      \ifcsdef{@glotype@#1@title}%
      {\def\glossarytocitle{\csname @glotype@#1@title\endcsname}}%
      {\def\glossarytocitle{\glossarytitle}}%
    }%
  }%
}
{
  \renewcommand*{\glssettocitle}[1]{%
    \ifcsdef{@glotype@#1@title}%
    {\def\glossarytocitle{\csname @glotype@#1@title\endcsname}}%
    {\def\glossarytocitle{\glossarytitle}}%
  }
}

```

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

```

\newcommand{\provideignoredglossary}{%
  \ifstar\glxtr@s@provideignoredglossary\glxtr@provideignoredglossary
}

```

gsxtr@provideignoredglossary Unstarred version.

```
\newcommand*{\glxtr@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
    {%
      \renewcommand*{\@gls@nohyperlist}{#1}%
    }%
    {}%
    \protected@eappto\@gls@nohyperlist{,#1}%
  }%
}
```

gsxtr@s@provideignoredglossary Starred form.

```
\newcommand*{\glxtr@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}%
    }%
    {}%
  }%
}
```

`\glxtrcopytoglossary` 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*\glxtrcopytoglossary}{%
  \@ifstar\s@glxtrcopytoglossary\glxtrcopytoglossary
}
```

`\@glxtrcopytoglossary`

```
\newcommand*\@glxtrcopytoglossary}[2]{%
  \glsdoifexists{#1}%
  {%
    \ifcsdef{glolist@#2}
    {%
      \protected@cseappto{glolist@#2}{#1,%}
    }%
    {%
      \glxtrundefaction{Glossary type ‘#2’ doesn’t exist}{}%
    }%
  }%
}
```

`\s@glxtrcopytoglossary`

```
\newcommand*\s@glxtrcopytoglossary}[2]{%
  \glsdoifexists{#1}%
  {%
    \ifcsdef{glolist@#2}
    {%
      \protected@csxappto{glolist@#2}{#1,%}
    }%
    {%
      \glxtrundefaction{Glossary type ‘#2’ doesn’t exist}{}%
    }%
  }%
}
```

1.3.1 Existence Checks

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

```
\renewcommand{\glsdoifexists}[2]{%
  \ifglstryexists{#1}{#2}{\glxtr@doifexists{#1}}%
}
```

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

```
\newrobustcmd{\glxtr@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\glstrundefdebug\expandafter
  {\expandafter\detokenize\expandafter{\glslabel}}%
\glstrundefaction{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]{%
  \ifglentryexists{#1}{\glstr@doifnoexists{#1}}{#2}%
}

```

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

```

\newrobustcmd{\glstr@doifnoexists}[1]{%
  \glstrundefaction{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]{%
    \ifglentryexists{#1}{#2}%
    {%
      \glstrundefaction{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}{You need to define a glossary entry
before you can use it.}%
      #3%
    }%
  }%
}
{%
  \glstr@warnonexistsordo\glsdoifexistsordo
  \newcommand{\glsdoifexistsordo}[3]{%
    \ifglentryexists{#1}{#2}%
    {%
      \glstrundefaction{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}%
    {%
      \glstrundefaction{Glossary type ‘#1’ already exists}{}%
      #3%
    }%
    {#2}%
  }%
}
{%
  \glstr@warnonexistsordo\doifglossarynoexistsordo
  \newcommand{\doifglossarynoexistsordo}[3]{%
    \ifglossaryexists*{#1}%
    {%
      \glstrundefaction{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@glstr@autoindex
      \@glstr@autoindexcrossrefs
    \fi
  }%
}
\appto\@gls@keymap{,{see}{see}}

```

`\glstrseelistsencap{<content>}`

`\glstrseelistsencap`

Encapsulates cross-reference list.

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

`\glstrseelistsdelim` Delimiter in cross-reference list.


```
\newcommand*{\glxtrseelistsdelim}{, }
```

```
\glxtrseelists{<label>}
```

\glxtrseelists

```
\newcommand*{\glxtrseelists}[1]{%
  \glsdoidexists{#1}%
  {%
    \def\@glxtr@seelists{%
      \letcs{\@glo@see}{glo\@glsdetoklabel{#1}@see}%
      \ifdefempty\@glo@see
      {}%
      {%
        \protected@edef\@glxtr@seelists{%
          \noexpand\glxtr@usesee\@glo@see\noexpand\@end@glxtr@usesee
        }%
      }%
      \letcs{\@glo@see}{glo\@glsdetoklabel{#1}@seealso}%
      \ifdefempty\@glo@see
      {}%
      {%
        \ifdefempty\@glxtr@seelists{%
          {\appto\@glxtr@seelists{\glxtrseelistsdelim}}%
          \protected@edef\@glxtr@seelists{%
            \noexpand\glxtruseseealsoformat{\@glo@see}%
          }%
        }%
        \letcs{\@glo@see}{glo\@glsdetoklabel{#1}@alias}%
        \ifdefempty\@glo@see
        {}%
        {%
          \ifdefempty\@glxtr@seelists{%
            {\appto\@glxtr@seelists{\glxtrseelistsdelim}}%
            \protected@edef\@glxtr@seelists{%
              \noexpand\glxtruseseeformat{\noexpand\seename}{\@glo@see}%
            }%
          }%
          \ifdefempty\@glxtr@seelists{%
            {\glxtrseelistsencap\@glxtr@seelists}%
          }%
        }%
      }%
    }%
  }
```

\glxtrusesee Apply \glssseeformat to the see key if not empty.

```
\newcommand*{\glxtrusesee}[1]{%
  \glsdoidexists{#1}%
  {%
    \letcs{\@glo@see}{glo\@glsdetoklabel{#1}@see}%
    \ifdefempty\@glo@see
    {}%
  }
```

```

    {%
      \expandafter\glxtr@usesee\@glo@see\@end@glxtr@usesee
    }%
  }%
}

```

```

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

```

```

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

```

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

```

\newcommand*{\glxtruseseeformat}[2]{%
  \glseeformat{#1}{#2}{}%
}

```

\glseeitemformat glossaries originally defined `\glseeitemformat` 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 `\glsfmtext` and `\glsfmname` instead. (The `text` field is chosen rather than the `short` field to allow for the “noshort” styles.)

```

\renewcommand*{\glseeitemformat}[1]{%
  \ifglshasshort{#1}{\glsfmtext{#1}}{\glsfmname{#1}}%
}

```

```
\glxtrhiername{<label>}
```

\glxtrhiername

Displays the hierarchical name for the given entry. The cross-reference format `\glseeitemformat` may be redefined to use this command to show the hierarchy, if required. This now uses `\glsfmtext` and `\glsfmname` instead of `\glaccessshort` and `\glaccessname` to allow for style formatting.

```

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

```

`\@glxtrhiername` Provide robust inner command.

```
\newrobustcmd*{\@glxtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glxtrifhasfield{parent}{#1}%
    {\expandafter\glxtrhiername\expandafter
      {\glscurrentfieldvalue}\glxtrhiernamesep}%
    {}%
    \ifglshasshort{#1}{\glsfmtext{#1}}{\glsfmname{#1}}%
  }%
}
```

`\Glsxtrhiername{<label>}`

`\Glsxtrhiername`

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

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

`\@Glsxtrhiername` Provide robust inner command.

```
\newrobustcmd*{\@Glsxtrhiername}[1]{%
  \glsdoifexists{#1}%
  {%
    \glxtrifhasfield{parent}{#1}%
    {%
      \expandafter\Glsxtrhiername\expandafter
        {\glscurrentfieldvalue}\glxtrhiernamesep
      \ifglshasshort{#1}{\glsfmtext{#1}}{\glsfmname{#1}}%
    }%
    {\ifglshasshort{#1}{\Glsfmtext{#1}}{\Glsfmname{#1}}}%
  }%
}
\glsmfuaddmap{\glxtrhiername}{\Glsxtrhiername}
```

`\GlsXtrhiername{<label>}`

`\GlsXtrhiername`

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}}%
  {\glstryname{#1}}%
}
```

`\@GlsXtrhiername` Provide robust inner command.

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

`\GLSxtrhiername{<label>}`

`\GLSxtrhiername`

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}%
  {%
    \glstrifhasfield{parent}{#1}%
    {%
      \expandafter\GLSxtrhiername\expandafter
      {\glscurrentfieldvalue}\glstrhiernamesep
      \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}}%
}
```

`\@GLSXRhiername` Provide robust inner command.

```
\newrobustcmd*{\@GLSXRhiername}[1]{%
\glsdoifexists{#1}%
{%
\glxstrifhasfield{parent}{#1}%
{\expandafter\GLSXRhiername\expandafter
{\glscurrentfieldvalue}\glxtrhiernamesep}%
}%
\ifglshashshort{#1}{\GLSfmttext{#1}}{\GLSfmtname{#1}}%
}%
}
\glsmfublocker{\GLSXRhiername}
```

`\glxtrhiernamesep` Separator used in `\glxtrhiername` and variants.

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

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

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

`\glxtrusealias` 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*{\glxtrusealias}[1]{%
\glsdoifexists{#1}%
{%
\letcs{\@glo@see}{glo\@glsdetoklabel{#1}@alias}%
\ifdefempty\@glo@see
{}%
{%
\expandafter\glxtrusealiasformat\expandafter{\@glo@see}%
}%
}%
}
```

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

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

`\glxtruseseealsoformat` The format used by `\glxtruseseealso`. The argument is the comma-separated list of cross-referenced labels.

```

\newcommand*{\glxtruseealsoformat}[1]{%
  \glseeformat[\seealso]{#1}{}%
}

```

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

```

\newrobustcmd{\glxtrseelist}[1]{%
  \protected@edef\glo@tmp{\noexpand\glseelist{#1}}\glo@tmp
}

```

`\glseelist` Redefine to make `\glseelist` more flexible.

```

\renewrobustcmd*{\glseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \let\@glsseeitem\@glxtr@seefirstitem
  \let\@glsseelastsep\glseelastsep
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\@glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glseelastsep
    \let\@gls@donext\glseesep
    \let\@glsseeitem\@glxtr@seeitem
    \let\@glsseelastsep\glseelastoxfordsep
  }%
}

```

`\glxtrtaggedlistsep` Separator between the tag and the list in `\glxtrtaggedlist`

```

\newcommand{\glxtrtaggedlistsep}{\space}

```

```

\glxtrtaggedlist{<singular tag>}{<plural
tag>}{<label prefix>}{<label list>}

```

`\glxtrtaggedlist`

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

```

\newrobustcmd*{\glxtrtaggedlist}[4]{%
  \begingroup
  \protected@edef\@gls@taggedlist@labels{#4}%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \let\@glsseeitem\@glxtr@seefirstitem
  \let\@glsseelastsep\glseelastsep
  \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}{\@glsseeitem{#1}}{\@glsseeitem{#1}}%
}

\@glsxtr@seefirstitem

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

\@glsseeitem Multi-entry cross-reference

\newcommand*{\@glsseeitem}[1]{%
  \glsname[all={noindex},setup={hyper=allmain}]{#1}%
}

\@glsseefirstitem Multi-entry cross-reference

\newcommand*{\@glsseefirstitem}{\@glsseeitem}

\@glsseefirstitem

\newcommand*{\@glsseefirstitem}{\@glsseeitem}

```

`\glsseelastoxfordsep`

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

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

```
\ifdef\alsiname
{\providecommand{\seealso}{\alsiname}}
{\providecommand{\seealso}{see also}}
```

`\glxtrindexseealso` If `\@xdycrossrefhook` is defined, provide a `seealso` crossref class. Otherwise this just does `\glsee` with `\seealso` 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\glxtruseseealsoformat\glsoopenbrace\string"
: close \string"\glsclosebrace\string")}%
}
```

Append to class list.

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

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

```
\newrobustcmd*{\glxtrindexseealso}[2]{%
\glxtr@wrglossary@encap{#1}
{%
\ifx\@glxtr@record@setting\@glxtr@record@setting@alsoindex
\@glxtr@recordsee{#1}{#2}%
\fi
\glsoifexists{#1}%
{%
\@glxtrwrglossmark
\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*{\glxtrindexseealso}{\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@glxtr@autoindex
      \glxtr@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}%
  \glxtr@aliashook{\@glo@label}%
}%
}

```

Provide user-level commands to access the values.

`\glxtralias`

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

`\glxtrseealsolabels`

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

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

```
\appto\@glo@autoseehook{%
  \ifdefvoid\@glo@alias
  {%
    \ifdefvoid\@glo@seealso
    {}%
    {%
      \protected@edef\@do@glssee{\noexpand\glxtrindexseealso
        {\@glo@label}{\@glo@seealso}}%
      \@do@glssee
    }%
  }%
  {%
    \protected@edef\@do@glssee{\noexpand\glxtrindexseealso
      {\@glo@label}{\@glo@seealso}}%
    \@do@glssee
  }%
}
```

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

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

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

`\glxtralias`

```
\glsaddstoragekey*{alias}{\glxtralias}
```

`\glxtrseealsolabels`

```
\glsaddstoragekey*{seealso}{\glxtrseealsolabels}
```

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}%
}
```

```

{%
\ifcvoid{glo@\@glo@label @seealso}%
}%
{%
\protected@edef\@do@glsee{\noexpand\glxtrindexseealso
{\@glo@label}{\csuse{glo@\@glo@label @seealso}}}%
\@do@glsee
}%
}%
{%

```

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

```

\ifdefvoid\@glo@see
{%
\protected@edef\@do@glsee{\noexpand\glsee
{\@glo@label}{\csuse{glo@\@glo@label @alias}}}%
\@do@glsee
}%
}%
}%
}
}

```

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

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

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

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

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

```

\newcommand*{\glxtraddallcrossrefs}{%
\forallglossaries{\@glo@type}%
{%
\forallsentries[\@glo@type]{\@glo@label}%
{%
\ifglused{\@glo@label}{\glxtraddunusedxrefs{\@glo@label}}}%
}%
}%
}

```

`\glxtraddunusedxrefs` Added user-level command in case user wants to redefine `\glxtraddallcrossrefs`

```
\newcommand*{\glxtraddunusedxrefs}[1]{\expandafter\@glxtr@addunusedxrefs\expandafter{#1}}
```

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

```
\newcommand*{\@glxtr@addunusedxrefs}[1]{%
```

```

\letcs{\@glo@see}{glo@glstdetoklabel{#1}@see}%
\ifvoid\@glo@see
{%
\expandafter\glstr@addunused\@glo@see\endglstr@addunused
}%
\letcs{\@glo@see}{glo@glstdetoklabel{#1}@seealso}%
\ifvoid\@glo@see
{%
\expandafter\glstr@addunused\@glo@see\endglstr@addunused
}%
}

```

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

```

\newcommand*{\glstr@addunused}[1][]{%
\@glstr@addunused
}

```

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

```

\def\@glstr@addunused#1\endglstr@addunused{%
\@for\@glstr@label:=#1\do
{%
\glstrifmulti\@glstr@label
{%
\letcs\@glstr@labellist{gls@combined@\@glstr@label @list}%
\@for\@glstr@multilabel:=\@glstr@labellist\do
{\@glstr@addunused\@glstr@multilabel\endglstr@addunused}%
}%
{%
\ifglused{\@glstr@label}{}%
{%
\gladd[format=glstrunusedformat]{\@glstr@label}%
\glunset{\@glstr@label}%
\expandafter\@glstr@addunusedxrefs\expandafter{\@glstr@label}%
}%
}%
}%
}

```

`\glstrunusedformat`

```

\newcommand*{\glstrunusedformat}[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 `\@glstr@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}{\relax}%
\@gls@restoreat
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}else
\ifnum\@glsxtr@docdefval=3\relax
```

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\@glstr@record@setting\@glstr@record@setting@off
{%
  \glstr@orgmakenoidxglossaries
Add marker to \@do@seeglossary but don't increment associated counter.
  \renewcommand{\@do@seeglossary}[2]{%
    \@@glstrwrglossmark

    \protected@edef\@gls@label{\glsdetoklabel{##1}}%
    \protected@write\@auxout{}{%
      \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {%
        \string\glsseeformat##2}%
      }%
    }%
  }%
}

Check for docdefs=restricted:
  \if@glstrdocdefrestricted

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

Disable document definitions.
  \@glstrdocdeffalse
  \fi
  \disable@keys{glossaries-extra}{docdef}%
}%
{%
  \PackageError{glossaries-extra}{\string\makenoidxglossaries\space
not permitted\MessageBreak
with record=\@glstr@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
}%
}

```

```

\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}%
}

\makenoidxgloss@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\glxtr@keylist{##1}%
\@Glsxtrpl
}
\glsmfuaddmap{\glxtrpl}{\Glsxtrpl}

```

```

\@Glsxtrpl
\newcommand*\@Glsxtrpl}[2][]{%
\ifglentryexists{##2}
{%
\ifblank{##1}{\GlsXtrWarning{##1}{##2}}%
}%
{%
\gls@defglossaryentry{##2}{name={##2},category=\glxtrcat,
description={\nopostdesc},##1}%
}%
\expandafter\Glspl\expandafter[\glxtr@keylist]{##2}%
}

```

```

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

```

Disable commands after the glossary:

```

\renewcommand\@printglossary[2]{%
\def\glxtr@printglossopts{##1}%
\glxtr@orgprintglossary{##1}{##2}%
\def\glxtr{\glxtr@disabledflycommand\glxtr}%
\def\glxtrpl{\glxtr@disabledflycommand\glxtrpl}%
\def\@Glsxtr{\glxtr@disabledflycommand\Glsxtr}%
\def\@Glsxtrpl{\glxtr@disabledflycommand\Glsxtrpl}%
}

```

\glxtr@disabledflycommand

```

\newcommand*\@glxtr@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}
```

`\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`:

```
\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 `nonumberlist` 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@pagetag}{#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

\@glstrpreloctag
  \newcommand*{\@glstrpreloctag}{%
    \let\@glstr@org@delimN\delimN
    \let\@glstr@org@delimR\delimR
    \let\@glstr@org@glignore\glignore
  \gdef is required as the delimiters may occur inside a scope.
    \gdef\@glstr@thisloctag{\@glstr@pagetag}%
    \renewcommand*{\delimN}{%
      \gdef\@glstr@thisloctag{\@glstr@pagetag}%
      \@glstr@org@delimN}%
    \renewcommand*{\delimR}{%
      \gdef\@glstr@thisloctag{\@glstr@pagetag}%
      \@glstr@org@delimR}%
    \renewcommand*{\glignore}[1]{%
      \gdef\@glstr@thisloctag{\relax}%
      \@glstr@org@glignore{##1}}%
    \@glstr@doloctag
  }

\@glstrpreloctag
  \newcommand*{\@glstrpreloctag}{}

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

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

\@glstrpostloctag
  \newcommand*{\@glstrpostloctag}{%
    \let\delimN\@glstr@org@delimN
    \let\delimR\@glstr@org@delimR
    \let\glignore\@glstr@org@glignore
    \protected@write\@auxout{%
      {\string\@glstr@savepreloctag{\glscurrententrylabel}\@glstr@thisloctag}}%
    }

\@glstrpostloctag
  \newcommand*{\@glstrpostloctag}{}

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

`\@glstr@doloctag`

```
\newcommand*{\@glstr@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{%
\@glstr@preloctag
\GlsXtrFormatLocationList{##1}%
\@glstr@postloctag
\gls@save@numberlist{##1}}%
\fi
}%
}
```

1.3.4 Entry Formatting, Hyperlinks and Indexing

`\glentryfmt` 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 `\glentryfmt` will need redefining as appropriate (or use `\defglentryfmt`). 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*{\glentryfmt}{%
\ifglshasshort{\glslabel}{\glsssetabbrvfmt{\glscategory{\glslabel}}}{}%
\glsifregular{\glslabel}%
{\glstrregularfont{\glsgenentryfmt}}%
{%
\ifglshasshort{\glslabel}%
{\glstrabbreviationfont{\glstrgenabbrvfmt}}%
{\glstrregularfont{\glsgenentryfmt}}%
}%
}
```

`\glstrregularfont` Font used for regular entries.

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

`\glstrabbreviationfont` Font used for abbreviation entries.

```
\newcommand*{\glstrabbreviationfont}[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.

`\glxtrdefaultentrytextfmt` This is the default command that `\glxtrgenentrytextfmt` is initialised to within `\@gls@link`.

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

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

`\glxtrgenentrytextfmt` 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 `\glxtrdefaultentrytextfmt` to change the default definition for this command.

```
\newcommand*{\glxtrgenentrytextfmt}{\glxtrdefaultentrytextfmt}
```

`\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}
```

```
\Glsfmtfield{\langle insert \rangle}{\langle cs \rangle}{\langle label \rangle}{\langle field \rangle}
```

\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
    \glssupercase\expandafter\expandafter\expandafter
    {%
      \csname glo@\glsdetoklabel{#3}@#4\endcsname #1%
    }%
  \else
    \expandafter\expandafter\expandafter
    \glssupercase\expandafter\expandafter\expandafter
    {%
      \expandafter\expandafter\expandafter
      #2\expandafter\expandafter\expandafter
      {\csname glo@\glsdetoklabel{#3}@#4\endcsname #1}%
    }%
  \fi
}
```

```

    }%
  \fi
}
\glsmfublocker{\GLSfmtfield}

\glsfmtinsert Formats \glsinsert.
\newcommand*{\glsfmtinsert}{%
  \ifdefempty\glsinsert{}%
    {\expandafter\glstrgenentrytextfmt\expandafter{\glsinsert}}%
  }

\GLSfmtinsert As above but all caps.
\newcommand*{\GLSfmtinsert}{%
  \ifdefempty\glsinsert{}%
    {%
      \expandafter\glssupercase\expandafter
        {\expandafter\glstrgenentrytextfmt\expandafter{\glsinsert}}%
    }%
  }

```

```
\glsifapplyinnerfmtfield{<label>}{<field>}{<true>}{<false>}
```

`\glsifapplyinnerfmtfield`

Does *<true>* if `\glsgenentryfmt` should encapsulate the given field with the inner format otherwise does *<false>*.

```

\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 `\glstrgenentrytextfmt`

```

\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
    {%
      \glsifplural
        {%

```

Plural form

```

      \glscapscase
    }%
  }

```

Don't adjust case

```
\ifglused\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

```
\ifglused\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

```
\ifglused\glslabel
```

```

        {%
Subsequent use
        \glsifapplyinnerfmtfield{\glslabel}{plural}%
        {%
            \expandafter\GLSaccessfmtplural\expandafter
            {\glsinsert}{\glstrgenentrytextfmt}{\glslabel}%
        }%
        {\GLSaccessplural{\glslabel}\GLSfmtinsert}%
    }%
    {%
First use
        \glsifapplyinnerfmtfield{\glslabel}{firstpl}%
        {%
            \expandafter\GLSaccessfmtfirstplural\expandafter
            {\glsinsert}{\glstrgenentrytextfmt}{\glslabel}%
        }%
        {\GLSaccessfirstplural{\glslabel}\GLSfmtinsert}%
    }%
    }%
    }%
    {%
Singular form
        \glscapscase
        {%
Don't adjust case
        \ifglused\glslabel
        {%
Subsequent use
        \glsifapplyinnerfmtfield{\glslabel}{text}%
        {%
            \expandafter\glsaccessfmttext\expandafter
            {\glsinsert}{\glstrgenentrytextfmt}{\glslabel}%
        }%
        {\glsaccesstext{\glslabel}\glsfmtinsert}%
    }%
    {%
First use
        \glsifapplyinnerfmtfield{\glslabel}{first}%
        {%
            \expandafter\glsaccessfmtfirst\expandafter
            {\glsinsert}{\glstrgenentrytextfmt}{\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}
```

`\glsxtrifwasglslikeandfirstuse`

```
\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\glxtrcurrentfield\glxtr@shortfieldname{#1}{#2}}%
}
```

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

```
\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).

```
\glxtr@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\glxtrorg@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}%
\@glxtr@field@linkdefs
#1%
\@gls@link[#2]{#3}{#4}%
\let\ifKV@glslink@hyper\glxtrorg@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 `\@glxtr@record`.

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

```
\let\@glxtr@org@gls@\@gls@
\def\@gls@#1#2{%
\def\glxtrcurrentfield{}%
\@glxtr@record{#1}{#2}{glslink}%
\@glxtr@org@gls@{#1}{#2}%
}%
```

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

```
\let\@glxtr@org@glspl@\@glspl@
\def\@glspl@#1#2{%
```

```

\def\glstrcurrentfield{}%
\@glstr@record{#1}{#2}{glslink}%
\@glstr@org@glsp1@{#1}{#2}%
}%

\@Gls@ Save the original definition and redefine.
\let\@glstr@org@Gls@\@Gls@
\def\@Gls@#1#2{%
\def\glstrcurrentfield{}%
\@glstr@record{#1}{#2}{glslink}%
\@glstr@org@Gls@{#1}{#2}%
}%

\@Glspl@ Save the original definition and redefine.
\let\@glstr@org@Glspl@\@Glspl@
\def\@Glspl@#1#2{%
\def\glstrcurrentfield{}%
\@glstr@record{#1}{#2}{glslink}%
\@glstr@org@Glspl@{#1}{#2}%
}%

\@GLS@ Save the original definition and redefine.
\let\@glstr@org@GLS@\@GLS@
\def\@GLS@#1#2{%
\def\glstrcurrentfield{}%
\@glstr@record{#1}{#2}{glslink}%
\@glstr@org@GLS@{#1}{#2}%
}%

\@GLSpl@ Save the original definition and redefine.
\let\@glstr@org@GLSpl@\@GLSpl@
\def\@GLSpl@#1#2{%
\def\glstrcurrentfield{}%
\@glstr@record{#1}{#2}{glslink}%
\@glstr@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\glstrcurrentfield{}%
\@glstr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}{%
\let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\def\glscustomtext{\glstrgenentrytextfmt{#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@glspresreset\@gobble
  \or
    \def\@gls@do@glspresreset{%
      \let\@gls@link@postkeys@checkfirsthyper\do@gls@link@checkfirsthyper
      \let\glsxtrifwasfirstuse\@firstoftwo\glslocalreset}%
  \or
    \def\@gls@do@glspresreset{%
      \let\@gls@link@postkeys@checkfirsthyper\do@gls@link@checkfirsthyper
      \let\glsxtrifwasfirstuse\@firstoftwo\glsreset}%
  \fi
}

```

\@gls@do@glspreunset

```

\newcommand*{\@gls@do@glspreunset}[1]{%

```

s@glslink@hyper@update@hook 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@gls@preunset{%
  \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@gls@preunset\@gobble
  \let\@gls@do@gls@prereset\@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

```

```

\current@textformat@csname
    \newcommand*{\@glxtr@current@textformat@csname}{\glstextformat}

\current@innertextformat@csname
    \newcommand*{\@glxtr@current@innertextformat@csname}{\glxtrdefaultentrytextfmt}

\glxtrassignlinktextfmt Used to assign \glstextformat and \glxtrgenentrytextfmt in the post-link
hook for “postfootnote” abbreviation styles.
    \newcommand*{\glxtrassignlinktextfmt}{}

\@glxtr@local@textformat Provide a key to locally change the text format.
    \define@key{glslink}{textformat}{%
        \ifcsdef{#1}
        {%
            \letcs{\@glxtr@local@textformat}{#1}%
            \def\@glxtr@current@textformat@csname{#1}%
        }%
        {%
            \PackageError{glossaries-extra}{Unknown control sequence name ‘#1’}{}%
        }%
    }

\glxtr@local@innertextformat Provide a key to locally change the inner text format.
    \define@key{glslink}{innertextformat}{%
        \ifcsdef{#1}
        {%
            \letcs{\@glxtr@local@innertextformat}{#1}%
            \def\@glxtr@current@innertextformat@csname{#1}%
        }%
        {%
            \PackageError{glossaries-extra}{Unknown control sequence name ‘#1’}{}%
        }%
    }

    \define@key{glslink}{prefix}{\def\glolinkprefix{#1}}

\glxtrinithyperoutside Set the default if the hyperoutside is omitted.
    \newcommand*{\glxtrinithyperoutside}{%
        \glsifattribute{\glslabel}{hyperoutside}{false}%
        {%
            \glxtr@hyperoutsidefalse
        }%
        {%
            \glxtr@hyperoutsidetrue
        }%
    }

\glxtr@inc@linkcount Does nothing by default.
    \newcommand*{\glxtr@inc@linkcount}{}

```

`\glslinkpresetkeys` User hook performed immediately before options are set. Does nothing by default.

$$\backslash\mathrm{newcommand}\ast\{\backslash\mathrm{glslinkpresetkeys}\}\{\}$$

`\GlsXtrExpandedFmt` Helper command that (protected) fully expands second argument and then applies it to the first, which must be a command that takes a single argument.

```
\newrobustcmd*{\GlsXtrExpandedFmt}[2]{%
\protected@edef\@glxstr@tmp{#2}%
\expandafter#1\expandafter{\@glxstr@tmp}%
}
```

\extracuse@equation@counter@or If in a numbered equation, change the counter to `equation`. This can be overridden 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*{\glxtr@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},%
              }%
            }%
          }%
        }%
      }%
    \glsxtr@do@autoadd
  \endgroup
}%
}

```

`\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\gls@type{\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{gls@textformat}%
```

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 `\gls@textformat` (new to v1.21).

```
\glsxtrinithyperoutside
```

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

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

Increment link counter if enabled (new to v1.26).

```
\glxtr@inc@linkcount
```

Check if the `equations` option has been set (new to v1.37).

```
\ifglxtr@equations
\@glxtr@use@equation@counter
\fi
```

As the original definition.

```
\do@gl:disablehyperinlist
\do@gl@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)

```
\glxtr@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\glxtrgenentrytextfmt\@glxtr@local@innertextformat
```

Check the value and the Hvalue before saving (v1.19).

```
\ifdefempty{\@glxtr@thevalue}%
{%
\@gls@saveentrycounter
}%
{%
\let\theglentrycounter\@glxtr@thevalue
\def\theHglentrycounter{\@glxtr@theHvalue}%
}%
\@gls@setsort{\glslabel}%
```

Check if the `textformat` key has been used.

```
\ifx\@glxtr@local@textformat\relax
```

Check textformat attribute (new to v1.21).

```
\glshasattribute{\glslabel}{textformat}%
{%
  \protected@edef\@glsxtr@attrval{\glsgetattribute{\glslabel}{textformat}}%
  \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:

```
\ifglsxtrinittwrglossbefore
  \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
```



```

\@glstr@textformat{\@noglslink@prefix@label{#3}}%
\fi
\fi

```

Do write if it should occur after the link text:

```

\ifglstrinitwrglossbefore
\else
\glstr@wrglossary@encap{#2}{\@do@wrglossary{#2}}%
\fi
}%

```

Restore original value of \glslinkprefix:

```

\let\glslinkprefix\@glstr@org@glslinkprefix

```

As the original definition:

```

\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
\@gls@restore@glslocal
}

```

```

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

```

```

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

```

```

\glsaddpresetkeys

```

```

\newcommand*{\glsaddpresetkeys}{}

```

```

\glsaddpostsetkeys

```

```

\newcommand*{\glsaddpostsetkeys}{}

```

\glsadd Redefine to include \@glstr@record and suppress in headings

```

\renewrobustcmd*{\glsadd}[2][\%
\glstr@trifinmark
\}%
{\%
\@gls@adjustmode
\begin@group
\@glstr@record{#1}{#2}{glossadd}%
\glsdoifexists{#2}%
{\%
\let\@glsnumberformat\@glstr@defaultnumberformat

\protected@edef\@gls@counter{\csname glo@glsdetoklabel{#2}@counter\endcsname}%
\def\@glstr@thevalue{}%
\def\@glstr@theHvalue{\@glstr@thevalue}%

```

Implement any default settings (before options are set)

```

\glsaddpresetkeys
\setkeys{glossadd}{#1}%

```

Implement any default settings (after options are set)

```
\glsaddpostsetkeys
\ifdefempty{\@glsxtr@thevalue}%
{%
  \@gls@saveentrycounter
}%
{%
  \let\theglsentrycounter\@glsxtr@thevalue
  \def\theHglentrycounter{\@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}}%
}%
\endgroup
}%
}
```

`\glsaddeach` Performs `\glsadd` for each entry listed in the mandatory argument.

```
\newrobustcmd{\glsaddeach}[2][]{%
  \@for\@gls@thislabel:=#2\do{\glsadd[#1]{\@gls@thislabel}}%
}
```

`\@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]{%
  \ifglentryexists{#1}%
  {%
    \ifglshashshort{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}%
      \glsifregular{#1}%
    }
  }
}
```

```

        {\let\@gls@field@font\glxtrregularfont}%
        {\let\@gls@field@font\@firstofone}%
    }%
    {%
        \glsifnotregular{#1}%
        {\let\@gls@field@font\@firstofone}%
        {\let\@gls@field@font\glxtrregularfont}%
    }%
}%
{%
    \let\@gls@field@font\@gobble
}%
}

```

```
\glxtrsaveinsert{<entry-label>}{<insert>}
```

`\glxtrsaveinsert`

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 `\glxtrfull` 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*{\glxtrsaveinsert}[2]{\def\glsinsert{}}
```

`\glxtrfullsaveinsert` As above but specifically for commands like `\glxtrfull`

```
\newcommand*{\glxtrfullsaveinsert}{\glxtrsaveinsert}
```

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

```

\def\@glstext@#1#2[#3]{%
    \def\glxtrcurrentfield{text}%
    \glxtrassignfieldfont{#2}%
    \glxtrsaveinsert{#2}{#3}%
    \@gls@field@link{#1}{#2}%
    {%
        \glsifapplyinnerfmtfield{#2}{text}%
        {%
            \@gls@field@font{\glsaccessfmttext{#3}{\glxtrgenentrytextfmt}{#2}}%
        }%
        {%
            \@gls@field@font{\glsaccesstext{#2}\glxtrgenentrytextfmt{#3}}%
        }%
    }%
}

```

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

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

```

\def\glstrcurrentfield{text}%
\glstrassignfieldfont{#2}%
\glstrsaveinsert{#2}{#3}%
\@gls@field@link[\let\glscapscase\@thirdofthree]{#1}{#2}%
{%
  \gl@ifapplyinnerfmtfield{#2}{text}%
  {%
    \@gls@field@font{\GLSaccessfmttext{#3}{\glstrgenentrytextfmt}{#2}}%
  }%
  {%
    \ifx\glscapscase\@thirdofthree
      \@gls@field@font{\GLSaccesstext{#2}%
        \glssupercase{\glstrgenentrytextfmt{#3}}}%
    \else
      \@gls@field@font{\glssaccestext{#2}\glstrgenentrytextfmt{#3}}%
    \fi
  }%
}%
}

```

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

```

\def\@Glsfirst@#1#2[#3]{%
  \def\glstrcurrentfield{text}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@secondofthree]{#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{text}%
    {%
      \@gls@field@font{\GLSaccessfmttext{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\GLSaccesstext{#2}\glstrgenentrytextfmt{#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.

\glstrchecknohyperfirst

```

\newcommand*{\glstrchecknohyperfirst}[1]{%
  \gl@ifattribute{#1}{nohyperfirst}{true}{\KV@glslink@hyperfalse}{}%
}

```

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

```

\def\@glsfirst@#1#2[#3]{%
  \def\glstrcurrentfield{first}%
  \glstrassignfieldfont{#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\gls@scapscase\@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\glstrifwasfirstuse\@firstoftwo
\let\glscapscase\@thirdofthree
\glstrchecknohyperfirst{#2}%
\glstr@check@complexstyle{#2}{#3}%
]%
{#1}{#2}%
{%
\ifx\glscapscase\@thirdofthree
\glrifapplyinnerfmtfield{#2}{first}%
{%
\@gls@field@font{\GLSaccessfmtfirst{#3}{\glstrgenentrytextfmt}{#2}}%
}%
{%
\@gls@field@font{\GLSaccessfirst{#2}%
\glsuppercase{\glstrgenentrytextfmt{#3}}}%
}%
\else
\glrifapplyinnerfmtfield{#2}{first}%
{%
\@gls@field@font{\glsaccessfmtfirst{#3}{\glstrgenentrytextfmt}{#2}}%
}%
{%
\@gls@field@font{\glsaccessfirst{#2}\glstrgenentrytextfmt{#3}}%
}%
\fi
}%
}

```

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

```

\def\@glsplural@#1#2[#3]{%
\def\glstrcurrentfield{text}%
\glstrassignfieldfont{#2}%
\glstrsaveinsert{#2}{#3}%
\@gls@field@link
[\let\glrifplural\@firstoftwo
\glstr@check@complexstyle{#2}{#3}%
]{#1}{#2}%
{%
\glrifapplyinnerfmtfield{#2}{plural}%
{%
\@gls@field@font{\glsaccessfmtplural{#3}{\glstrgenentrytextfmt}{#2}}%
}%
{%
\@gls@field@font{\glsaccessplural{#2}\glstrgenentrytextfmt{#3}}%
}%
}%
}

```

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

```

\def\@Glsplural@#1#2[#3]{%
  \def\glstrcurrentfield{text}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glsifplural\@firstoftwo
   \let\glscapscase\@secondofthree
   \glstr@check@complexstyle{#2}{#3}%
  ]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{plural}%
    {%
      \@gls@field@font{\Glsaccessfmtplural{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccessplural{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

```

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

```

\def\@Glsplural@#1#2[#3]{%
  \def\glstrcurrentfield{text}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glsifplural\@firstoftwo
   \let\glscapscase\@thirdofthree
   \glstr@check@complexstyle{#2}{#3}%
  ]%
  {#1}{#2}%
  {%
    \ifx\glscapscase\@thirdofthree
      \glsifapplyinnerfmtfield{#2}{plural}%
      {%
        \@gls@field@font{\Glsaccessfmtplural{#3}{\glstrgenentrytextfmt}{#2}}%
      }%
      {%
        \@gls@field@font{\Glsaccessplural{#2}%
          \glssupercase{\glstrgenentrytextfmt{#3}}}%
      }%
    \else
      \glsifapplyinnerfmtfield{#2}{plural}%
      {%
        \@gls@field@font{\glsaccessfmtplural{#3}{\glstrgenentrytextfmt}{#2}}%
      }%
      {%
        \@gls@field@font{\glsaccessplural{#2}\glstrgenentrytextfmt{#3}}%
      }%
    \fi
  }%
}

```

```

\fi
}%
}

```

`\@glsfirstplural@` No case changing 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
\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\gls@capscase\@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\glsaps\@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}}%
}%
}%
}

```

\@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\gls caps case\@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\gls caps case\@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\glstrcurrentfield{description}%
\glstrassignfieldfont{#2}%
\glstrsaveinsert{#2}{#3}%
\@gls@field@link{#1}{#2}%
{%
  \gl@ifapplyinnerfmtfield{#2}{desc}%
  {%
    \@gls@field@font{\gl@accessfmtdesc{#3}{\glstrgenentrytextfmt}{#2}}%
  }%
  {%
    \@gls@field@font{\gl@accessdesc{#2}\glstrgenentrytextfmt{#3}}%
  }%
}%
}

\@Glsdesc@ First letter uppercase version.
\def\@Glsdesc@#1#2[#3]{%
  \def\glstrcurrentfield{description}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]{#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{desc}%
    {%
      \@gls@field@font{\Glsaccessfmtdesc{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccessdesc{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

\@GLSdesc@ All uppercase version.
\def\@GLSdesc@#1#2[#3]{%
  \def\glstrcurrentfield{description}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{desc}%
    {%
      \@gls@field@font{\GLSaccessfmtdesc{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\GLSaccessdesc{#2}%
        \gl@uppercase{\glstrgenentrytextfmt{#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}}}%
}%
}%
}

\@glsymbol@
\def\@glsymbol@#1#2[#3]{%
\def\glsxtrcurrentfield{symbol}%
\glsxtrassignfieldfont{#2}%
\glsxtrsaveinsert{#2}{#3}%
\@gls@field@link{#1}{#2}%
{%
\glsifapplyinnerfmtfield{#2}{symbol}%
{%
\@gls@field@font{\glsaccessfmtymsymbol{#3}{\glsxtrgenentrytextfmt}{#2}}%
}%
{%
\@gls@field@font{\glsaccesssymbol{#2}\glsxtrgenentrytextfmt{#3}}%
}%
}%
}

\@Glsymbol@ First letter uppercase version.
\def\@Glsymbol@#1#2[#3]{%
\def\glsxtrcurrentfield{symbol}%
\glsxtrassignfieldfont{#2}%
\glsxtrsaveinsert{#2}{#3}%
\@gls@field@link
[\let\glscapscale\@secondofthree]%
{#1}{#2}%
{%
\glsifapplyinnerfmtfield{#2}{symbol}%
{%
\@gls@field@font{\Glsaccessfmtymsymbol{#3}{\glsxtrgenentrytextfmt}{#2}}%
}%
{%
\@gls@field@font{\Glsaccesssymbol{#2}\glsxtrgenentrytextfmt{#3}}%
}%
}%
}

```

\@GLSsymbol@ All uppercase version.

```

\def\@GLSsymbol@#1#2[#3]{%
  \def\glstrcurrentfield{symbol}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{symbol}%
    {%
      \@gls@field@font{\GLSaccessfmtsymbol{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\GLSaccesssymbol{#2}%
        \glssupercase{\glstrgenentrytextfmt{#3}}}%
    }%
  }%
}

```

\@glssymbolplural@ No case-changing version.

```

\def\@glssymbolplural@#1#2[#3]{%
  \def\glstrcurrentfield{symbol}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree
  \let\gl@ifplural\@firstoftwo
  ]{#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{symbolplural}%
    {%
      \@gls@field@font{\glaccessfmtsymbolplural{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\glaccesssymbolplural{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

```

\@Glsymbolplural@ First letter uppercase version.

```

\def\@Glsymbolplural@#1#2[#3]{%
  \def\glstrcurrentfield{symbol}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree
  \let\gl@ifplural\@firstoftwo
  ]{#1}{#2}%
  {%

```

```

\glsifapplyinnerfmtfield{#2}{symbolplural}%
{%
  \gls@field@font{\Glsaccessfmts symbolplural{#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\gls@scapscase\@thirdofthree
   \let\glsifplural\@firstoftwo
  ]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{symbolplural}%
    {%
      \gls@field@font{\Glsaccessfmts symbolplural{#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\glstrcurrentfield{user1}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]{#1}{#2}%
  {%
    \glrifapplyinnerfmtfield{#2}{useri}%
    {%
      \@gls@field@font{\Glsaccessfmtuseri{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccessuseri{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

```

\@GLSuseri@ All uppercase version.

```

\def\@GLSuseri@#1#2[#3]{%
  \def\glstrcurrentfield{user1}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]{%
    #1}{#2}%
  {%
    \glrifapplyinnerfmtfield{#2}{useri}%
    {%
      \@gls@field@font{\GLSaccessfmtuseri{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \ifx\glscapscase\@thirdofthree
        \@gls@field@font{\GLSaccessuseri{#2}%
          \glssupercase{\glstrgenentrytextfmt{#3}}}%
      \else
        \@gls@field@font{\glsaccessuseri{#2}\glstrgenentrytextfmt{#3}}%
      \fi
    }%
  }%
}

```

\@glsuserii@ User 2 field.

```

\def\@glsuserii@#1#2[#3]{%
  \def\glstrcurrentfield{user2}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#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\glsapsacase\@thirdofthree]%
  {#1}{#2}%
  {%
    \glsifapplyinnerfmtfield{#2}{userii}%
    {%
      \@gls@field@font{\GLSaccessfmtuserii{#3}{\glsxtrgenentrytextfmt}{#2}}%
    }%
    {%
      \ifx\glsapsacase\@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\glstrcurrentfield{user3}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  {#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{useriii}%
    {%
      \@gls@field@font{\glsaccessfmtuseriii{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\glsaccessuseriii{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

\@Glsuseriii@ First letter uppercase version.
\def\@Glsuseriii@#1#2[#3]{%
  \def\glstrcurrentfield{user3}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]%
  {#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{useriii}%
    {%
      \@gls@field@font{\Glsaccessfmtuseriii{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccessuseriii{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

\@GLSuseriii@ All uppercase version.
\def\@GLSuseriii@#1#2[#3]{%
  \def\glstrcurrentfield{user3}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{useriii}%
    {%

```

```

        \@gls@field@font{\GLSaccessfmtuseriii{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
        \ifx\glscapscase\@thirdofthree
            \@gls@field@font{\GLSaccessuseriii{#2}%
                \glssupercase{\glstrgenentrytextfmt{#3}}}%
        \else
            \@gls@field@font{\glssaccessuseriii{#2}\glstrgenentrytextfmt{#3}}%
        \fi
    }%
}%
}

```

\@glsuseriv@ User 4 field.

```

\def\@glsuseriv@#1#2[#3]{%
    \def\glstrcurrentfield{user4}%
    \glstrassignfieldfont{#2}%
    \glstrsaveinsert{#2}{#3}%
    \@gls@field@link
    {#1}{#2}%
    {%
        \glrifapplyinnerfmtfield{#2}{useriv}%
        {%
            \@gls@field@font{\glssaccessfmtuseriv{#3}{\glstrgenentrytextfmt}{#2}}%
        }%
        {%
            \@gls@field@font{\glssaccessuseriv{#2}\glstrgenentrytextfmt{#3}}%
        }%
    }%
}

```

\@Glsuseriv@ First letter uppercase version.

```

\def\@Glsuseriv@#1#2[#3]{%
    \def\glstrcurrentfield{user4}%
    \glstrassignfieldfont{#2}%
    \glstrsaveinsert{#2}{#3}%
    \@gls@field@link
    [\let\glscapscase\@secondofthree]%
    {#1}{#2}%
    {%
        \glrifapplyinnerfmtfield{#2}{useriv}%
        {%
            \@gls@field@font{\Glsaccessfmtuseriv{#3}{\glstrgenentrytextfmt}{#2}}%
        }%
        {%
            \@gls@field@font{\Glsaccessuseriv{#2}#3}%
        }%
    }%
}

```

\@GLSuseriv@ All uppercase version.

```

\def\@GLSuseriv@#1#2[#3]{%
  \def\glstrcurrentfield{user4}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{useriv}%
    {%
      \@gls@field@font{\GLSaccessfmtuseriv{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \ifx\glscapscase\@thirdofthree
        \@gls@field@font{\GLSaccessuseriv{#2}%
          \glssupercase{\glstrgenentrytextfmt{#3}}}%
      \else
        \@gls@field@font{\glssaccessuseriv{#2}\glstrgenentrytextfmt{#3}}%
      \fi
    }%
  }%
}

```

\@glsuserv@ User 5 field.

```

\def\@glsuserv@#1#2[#3]{%
  \def\glstrcurrentfield{user5}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  {#1}{#2}%
  {%
    \gl@ifapplyinnerfmtfield{#2}{userv}%
    {%
      \@gls@field@font{\glssaccessfmtuserv{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\glssaccessuserv{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

```

\@Glsuserv@ First letter uppercase version.

```

\def\@Glsuserv@#1#2[#3]{%
  \def\glstrcurrentfield{user5}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#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\glstrcurrentfield{user6}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link
  [\let\glscapscase\@secondofthree]%
  {#1}{#2}%
  {%
    \glrifapplyinnerfmtfield{#2}{uservi}%
    {%
      \@gls@field@font{\Glsaccessfmtuservi{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \@gls@field@font{\Glsaccessuservi{#2}\glstrgenentrytextfmt{#3}}%
    }%
  }%
}

\@GLSuservi@ All uppercase version.
\def\@GLSuservi@#1#2[#3]{%
  \def\glstrcurrentfield{user6}%
  \glstrassignfieldfont{#2}%
  \glstrsaveinsert{#2}{#3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {#1}{#2}%
  {%
    \glrifapplyinnerfmtfield{#2}{uservi}%
    {%
      \@gls@field@font{\GLSaccessfmtuservi{#3}{\glstrgenentrytextfmt}{#2}}%
    }%
    {%
      \ifx\glscapscase\@thirdofthree
        \@gls@field@font{\GLSaccessuservi{#2}%
          \glssupercase{\glstrgenentrytextfmt{#3}}}%
      \else
        \@gls@field@font{\glsaccessuservi{#2}\glstrgenentrytextfmt{#3}}%
      \fi
    }%
  }%
}

```

Commands like `\acrshort` already set `\glrifplural`, but they don't set `\glstrifwasfirstuse` 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.

`\@@glstr@base@acrcmd@warn` Warn user that they need to use to new abbreviation commands.

```

\newcommand*{\@@glxtr@base@acrcmd@warn}[2]{%
  \GlossariesExtraWarning{Base acronym command \string#1\space
    should not be used with new abbreviation definitions. Use
    \string#2\space instead}%
}

\@glxtr@base@acrcmd Warn user that they need to use to new abbreviation commands.
\let\@glxtr@base@acrcmd\@@glxtr@base@acrcmd@warn

The following acr commands don't support innertextformat.

\@acrshort No case change.
\def\@acrshort#1#2[#3]{%
  \def\glxtrcurrentfield{short}%
  \@glxtr@base@acrcmd\acrshort\glxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasglslike\@secondoftwo
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsaccessshort{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}

\@Acrshort First letter uppercase.
\def\@Acrshort#1#2[#3]{%
  \def\glxtrcurrentfield{short}%
  \@glxtr@base@acrcmd\Acrshort\Glsxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasglslike\@secondoftwo
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccessshort{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}

```

\@ACRshort All uppercase.

```

\def\@ACRshort#1#2[#3]{%
  \def\glxtrcurrentfield{short}%
  \@glxtr@base@acrcmd\ACRshort\GLSxtrshort
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasglslike\@secondoftwo
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsaccessshort{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}

```

\@acrshortpl No case change.

```

\def\@acrshortpl#1#2[#3]{%
  \def\glxtrcurrentfield{short}%
  \@glxtr@base@acrcmd\acrshortpl\glxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasglslike\@secondoftwo
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsaccessshortpl{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}

```

\@Acrshortpl First letter uppercase.

```

\def\@Acrshortpl#1#2[#3]{%
  \def\glxtrcurrentfield{short}%
  \@glxtr@base@acrcmd\Acrshortpl\Glsxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasglslike\@secondoftwo
    \let\glxtrifwasfirstuse\@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\glstrcurrentfield{short}%
  \@glstr@base@acrcmd\ACRshortpl\GLSxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glstrifwasglslike\@secondoftwo
    \let\glstrifwasfirstuse\@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\glstrcurrentfield{long}%
  \@glstr@base@acrcmd\acrlong\glstrlong
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glstrifwasglslike\@secondoftwo
    \let\glstrifwasfirstuse\@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\glstrcurrentfield{long}%
\@glstr@base@acrcmd\Acrlong\Glsxtrlong
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glstrifwasglslike\@secondoftwo
\let\glstrifwasfirstuse\@secondoftwo
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsaccesslong{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
```

\@ACRlong All uppercase.

```
\def\@ACRlong#1#2[#3]{%
\def\glstrcurrentfield{long}%
\@glstr@base@acrcmd\ACRlong\GLSxtrlong
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glstrifwasglslike\@secondoftwo
\let\glstrifwasfirstuse\@secondoftwo
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glssupercase{\acronymfont{\Glsaccesslong{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
```

\@acrlongpl No case change.

```
\def\@acrlongpl#1#2[#3]{%
\def\glstrcurrentfield{long}%
\@glstr@base@acrcmd\acrlongpl\glxtrlongpl
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glstrifwasglslike\@secondoftwo
\let\glstrifwasfirstuse\@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\glstrcurrentfield{long}%
  \@glsxtr@base@acrcmd\Acrlongpl\Glsxtrlongpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glstrifwasglslike\@secondoftwo
    \let\glstrifwasfirstuse\@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\glstrcurrentfield{long}%
  \@glsxtr@base@acrcmd\ACRlongpl\GLSxtrlongpl
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glstrifwasglslike\@secondoftwo
    \let\glstrifwasfirstuse\@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 `\glstrfull` etc) is present.

```
\@acrfull
\def\@acrfull#1#2[#3]{%
  \def\glstrcurrentfield{%
    \@glstr@base@acrcmd\acrfull\glstrfull
    \acrfullfmt{#1}{#2}{#3}%
  }

\@Acrfull
\def\@Acrfull#1#2[#3]{%
  \def\glstrcurrentfield{%
    \@glstr@base@acrcmd\Acrfull\Glsstrfull
    \Acrfullfmt{#1}{#2}{#3}%
  }

\@ACRfull
\def\@ACRfull#1#2[#3]{%
  \def\glstrcurrentfield{%
    \@glstr@base@acrcmd\ACRfull\GLSstrfull
    \ACRfullfmt{#1}{#2}{#3}%
  }

\@acrfullpl
\def\@acrfullpl#1#2[#3]{%
  \def\glstrcurrentfield{%
    \@glstr@base@acrcmd\acrfullpl\glstrfullpl
    \acrfullplfmt{#1}{#2}{#3}%
  }

\@Acrfullpl
\def\@Acrfullpl#1#2[#3]{%
  \def\glstrcurrentfield{%
    \@glstr@base@acrcmd\Acrfullpl\Glsstrfullpl
    \Acrfullplfmt{#1}{#2}{#3}%
  }

\@ACRfullpl
\def\@ACRfullpl#1#2[#3]{%
  \def\glstrcurrentfield{%
    \@glstr@base@acrcmd\ACRfullpl\GLSstrfullpl
    \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\glstrcurrentfield{#1}%
  \glstrassignfieldfont{##2}%
  \glstrsaveinsert{##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\glstrcurrentfield{#1}%
  \glstrassignfieldfont{##2}%
  \glstrsaveinsert{##2}{##3}%
  \@gls@field@link[\let\glscapscase\@thirdofthree]%
  {##1}{##2}{\@gls@field@font{\glssupercase{#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}{}

\@gls@link@postkeys@checkfirsthyper Need another check after preunset and postunset options have been applied.
\newcommand*{\@gls@link@postkeys@checkfirsthyper}{}

\@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 com-
mands 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@glsglslisablehyperinlist
{%
\let\@glsglslxtr@do@glsglslisablehyperinlist\do@glsglslisablehyperinlist
\renewcommand*{\do@glsglslisablehyperinlist}{%
\@glsglslxtr@do@glsglslisablehyperinlist
\glsglsifattribute{\glsglslabel}{nohyper}{true}{\KV@glsglslink@hyperfalse}{}}%
}
}
{}

```

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

```

\define@boolkey{glsglslink}{noindex}[true]{}
\KV@glsglslink@noindexfalse

```

`\@glsglslsave@glsglsllocal` Defined in glossaries v4.50 so may not be defined.

```

\providecommand*{\@glsglslsave@glsglsllocal}{%
\let\if@org@KV@glsglslink@local\ifKV@glsglslink@local
}

```

`\@glsglslrestore@glsglsllocal` Defined in glossaries v4.50 so may not be defined.

```

\providecommand*{\@glsglslrestore@glsglsllocal}{%
\ifKV@glsglslink@local
\let\@glsglsl@do@glsglslunset\glsglsllocalunset
\else
\let\@glsglsl@do@glsglslunset\glsglslunset
\fi
\let\ifKV@glsglslink@local\if@org@KV@glsglslink@local
}

```

`\glsglsl@default@restore@glsglsllocal` Save default definition of `\@glsglslrestore@glsglsllocal`

```

\let\@glsglsl@default@restore@glsglsllocal\@glsglslrestore@glsglsllocal

```

`\glsglsl@ignore@restore@glsglsllocal`

```

\newcommand*{\@glsglsl@ignore@restore@glsglsllocal}{%
\let\@glsglsl@do@glsglslunset\@gobble
\let\ifKV@glsglslink@local\if@org@KV@glsglslink@local
}

```

`\@glsglsl@do@glsglslunset` Defined in glossaries v4.50 so may not be defined.

```

\providecommand*{\@glsglsl@do@glsglslunset}[1]{\glsglslunset{#1}}

```

`\@glsglsl@default@glsglslink@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 `\glsglsl` then it won't get reset if the default option list doesn't set it.

```

\newcommand*{\@glsglsl@default@glsglslink@opts}{}

```

If `\@glsglsl@setdefault@glsglslink@opts` has been defined (glossaries v4.20) use it to set the default keys in `\@glsglslink`.

`\@gls@setdefault@glslink@opts`

```
\ifdef\@gls@setdefault@glslink@opts
{
  \renewcommand*{\@gls@setdefault@glslink@opts}{%
    \KV@glslink@noindexfalse
    \expandafter\setupglslink\expandafter{\@gls@default@glslink@opts}%
    \@glstrsetaliasnoindex
  }
}
{
```

Not defined so prepend it to `\do@glsglslink@hyperinlist` to achieve the same effect.

```
\newcommand*{\@gls@setdefault@glslink@opts}{%
  \KV@glslink@noindexfalse
  \expandafter\setupglslink\expandafter{\@gls@default@glslink@opts}%
  \@glstrsetaliasnoindex
}
\preto\do@glsglslink@hyperinlist{\@gls@setdefault@glslink@opts}
}
```

`\glstrsetaliasnoindex` 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*{\glstrsetaliasnoindex}{%
  \KV@glslink@noindextrue
}
```

`\@glstrsetaliasnoindex` 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 `\ifcvoid`.

```
\newcommand*{\@glstrsetaliasnoindex}{%
  \ifcvoid{glo@glsglslink@label{\glslabel}@alias}%
  {}%
}%
\let\glstrindexaliased\glstrindexaliased
\glstrsetaliasnoindex
\let\glstrindexaliased\@no@glstrindexaliased
}%
}
```

`\@glstrindexaliased`

```
\newcommand{\@glstrindexaliased}{%
  \ifKV@glslink@noindex
  \else
  \begingroup
  \let\@glsnumberformat\@glstr@defaultnumberformat

  \protected@edef\@gls@counter{\csname glo@glsglslink@label{\glslabel}@counter\endcsname}%
}
```

```

\glxtr@saveentrycounter
\glxtr@wrglossary@encap{\glxtralias{\glslabel}}{\do@wrglossary{\glxtralias{\glslabel}}}%
\endgroup
\fi
}

```

`\@no@glxtrindexaliased`

```

\newcommand{\@no@glxtrindexaliased}{%
  \PackageError{glossaries-extra}{\string\glxtrindexaliased\space
    not permitted outside definition of \string\glxtrsetaliasnoindex}%
  {}%
}

```

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

```

\let\glxtrindexaliased\@no@glxtrindexaliased

```

`\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,}%
}

```

`\glxtrifindexing` Provide user level command to access it in `\glswriteentry`.

```

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

```

```

\glxtr@wrglossary@encap{\label}{\whatsit}

```

`\glxtr@wrglossary@encap`

Encapsulate indexing `whatsit` and increment indexed count. See also `\glxtrdowrglossaryhook`

```

\newcommand*{\glxtr@wrglossary@encap}[2]{\glsecapwrcontent{#2\@glxtr@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 L^AT_EX run before calling `bib2gls`.

`\@glxtr@inc@indexcount`

```
\newcommand*{\@glxtr@inc@indexcount}[1]{%
  \ifcsdef{glo@\glstetoklabel{#1}@indexed}
  {%
    \csxdef{glo@\glstetoklabel{#1}@indexed}{%
      \expandafter\number\expandafter\numexpr\csname glo@\glstetoklabel{#1}@indexed\endcsname+1}%
    }%
    {%
      \csgdef{glo@\glstetoklabel{#1}@indexed}{1}%
    }%
  }
}
```

`\glstentryindexcount`

```
\newcommand*{\glstentryindexcount}[1]{%
  \ifcsdef{glo@\glstetoklabel{#1}@indexed}
  {\csuse{glo@\glstetoklabel{#1}@indexed}}%
  {0}%
}
```

`\glstifindexed`

```
\newcommand*{\glstifindexed}[3]{%
  \ifcsdef{glo@\glstetoklabel{#1}@indexed}%
  {\expandafter\ifnum\csname glo@\glstetoklabel{#1}@indexed\endcsname>0 #2\else#3\fi}%
  {#3}%
}
```

`\glstaddallunindexed`

```
\newcommand*{\glstaddallunindexed}[1][\@glo@types]{%
  \forallglstentries[#1]{\@glo@entry}%
  {%
    \glstifindexed{\@glo@entry}{\glstadd[format=glstignore]{\@glo@entry}}%
  }%
}
```

`\glstencapwrcontent` This command was added to glossaries v4.50 so may not be defined.

```
\providecommand*{\glstencapwrcontent}[1]{#1}
```

`\glstwriteentry` Redefine to test for `indexonlyfirst` category attribute. This needs to use `\GlsXtrIfUnusedOrUndefined` instead of `\ifglstused` to allow it to work with `bib2gls`.

```
\renewcommand*{\glstwriteentry}[2]{%
  \glstxtrifindexing
  {%
    \ifglstindexonlyfirst
      \GlsXtrIfUnusedOrUndefined{#1}
      {#2}%
      {\glstxrdoautoindexname{#1}{dualindex}}%
    \else
      \glstifattribute{#1}{indexonlyfirst}{true}%
    \fi
  }
```

```

    {%
      \GlsXtrIfUnusedOrUndefined{#1}%
      {#2}%
      {\glstrdoautoindexname{#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{\@glstr@do@@wrindex
\glstrdowrglossaryhook{\@gls@label}%
}

```

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

Similarly for the “noidx” version:

```

\gls@noidxglossary
\appto\gls@noidxglossary{\@glstr@do@@wrindex
\glstrdowrglossaryhook{\@gls@label}%
}

```

```

\@glstr@do@@wrindex
\newcommand*{\@glstr@do@@wrindex}{%
\glstrdoautoindexname{\@gls@label}{dualindex}%
}

```

`\glstrdowrglossaryhook` 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*{\glstrdowrglossaryhook}[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@glshyp@opt User version
\newcommand*{\@alt@glshyp@opt}[1][]{%
  \let\glslinkvar\@firstofthree
  \expandafter\@glshyp@opt@cs\expandafter[\@glshyp@opt@keys,#1]}

\@glshyp@opt@char Contains the character used as the command modifier.
\newcommand*{\@glshyp@opt@char}{ }

\@glshyp@opt@keys Contains the option list used as the command modifier.
\newcommand*{\@glshyp@opt@keys}{ }

\GlsXtrSetAltModifier
\newcommand*{\GlsXtrSetAltModifier}[2]{%
  \let\@glshyp@opt\@glshyp@opt
  Check that the supplied character isn't "+" or "*"
  \ifstrequal{#1}{+}%
  {\PackageError{glossaries-extra}%
    {Can't use '+' as modifier (it's already in use)}{}}%
  {%
    \ifstrequal{#1}{*}%
    {\PackageError{glossaries-extra}%
      {Can't use '*' as modifier (it's already in use)}{}}%
    {%
      }%
    \def\@glshyp@opt@char{#1}%
    \def\@glshyp@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@glshyp@opt}[1][]{%
    \let\glslinkvar\@thirdofthree
    \@glshyp@opt@cs[#1,#1]%
  }%
}

\GlsXtrSetStarModifier Allow user to override the star modifier.
\newcommand*{\GlsXtrSetStarModifier}[1]{%

```

```

\renewcommand*{\s@glshyp@opt}[1][\glshyp@opt]{%
  \let\glslinkvar\@secondofthree
  \glshyp@opt@cs[#1,##1]%
}
}

```

`\glstr@org@dohyperlink`

```

\let\glstr@org@dohyperlink\glsdohyperlink

```

`\glshyp@opt` Since `\glshyp@opt` uses `\glslink`, it's necessary to patch it uses `\glsdohyperlink` instead of `\glstr@dohyperlink`. The simplest way to achieve this is to locally let `\glstr@dohyperlink` to `\glsdohyperlink`.

This command is provided by glossary-hypernav so it may not exist.

```

\ifdef\glshyp@opt
{
  \renewcommand*{\glshyp@opt}[3][\glshyp@opt]{%
    \protected@edef\glshyp@optlabel{#2}\protected@edef\glshyp@opttitle{#3}%
  }
}

```

Scope:

```

{
  \let\glstr@dohyperlink\glstr@org@dohyperlink
  \glslink{\glshyp@optlabel{#1}{#2}}{#3}%
}
}
{}

```

Patch if glossaries pre 4.50.

```

\ifdef\@glshyp@opt
{}
{}

```

`\glshyp@opt`

```

\renewcommand*{\glshyp@opt}{\protect\@glshyp@opt}

```

`\@glshyp@opt`

```

\newcommand*{\@glshyp@opt}[3][\glshyp@opt]{%
  \glshyp@opt{#1}{#2}{#3}%
}
}

```

`\glshyp@opt` Similarly for `\glshyp@opt`

```

\ifdef\@glshyp@opt
{}
\renewcommand*{\glshyp@opt}[3]{%
  \protected@write\@auxout{\string\glshyp@opt{#1}{#2}}{#3}%
  \glstr@do@org@target{\glshyp@optlabel{#1}{#2}}{#3}%
}

```

```

\ifcsdef{@gls@hypergroup@list@#1}%
{%
  \letcs{@gls@list@}{@gls@hypergroup@list@#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@hypergroup@rerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
    }%
  }%
  {%
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef@gls@hypergroup@rerun{%
      \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.

```

\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 problems.

```

\renewrobustcmd*{\glshyperlink}[2][\glstrytext{\@glo@label}]{%
\glsdoifexists{#2}%
{%

```



```

\def\@glo@label{#2}%

{\protected@edef\glslabel{#2}%
\@glslink{\glo@linkprefix\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\@glsxtrp@short@
\let\@Glsxtrshort\@Glsxtrp@short@
\let\@GLSxtrshort\@GLSxtrp@short@
\let\@glsxtrlong\@glsxtrp@long@
\let\@Glsxtrlong\@Glsxtrp@long@
\let\@GLSxtrlong\@GLSxtrp@long@
\let\@glsxtrshortpl\@glsxtrp@shortpl@
\let\@Glsxtrshortpl\@Glsxtrp@shortpl@
\let\@GLSxtrshortpl\@GLSxtrp@shortpl@
\let\@glsxtrlongpl\@glsxtrp@longpl@
\let\@Glsxtrlongpl\@Glsxtrp@longpl@
\let\@GLSxtrlongpl\@GLSxtrp@longpl@
\let\@acrshort\@glsxtrp@acrshort@
\let\@Acrshort\@Glsxtrp@acrshort@
\let\@ACRshort\@GLSxtrp@acrshort@
\let\@acrshortpl\@glsxtrp@acrshortpl@
\let\@Acrshortpl\@Glsxtrp@acrshortpl@
\let\@ACRshortpl\@GLSxtrp@acrshortpl@
\let\@acrlong\@glsxtrp@acrlong@
\let\@Acrlong\@Glsxtrp@acrlong@
\let\@ACRlong\@GLSxtrp@acrlong@
\let\@acrlongpl\@glsxtrp@acrlongpl@
\let\@Acrlongpl\@Glsxtrp@acrlongpl@
\let\@ACRlongpl\@GLSxtrp@acrlongpl@
}

```

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

```

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

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

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

\@glsxtrp@plural@
\def\@glsxtrp@plural@#1#2[#3]{\@glsxtrp@plural@{#1}{#2}{#3}}

\@Glsxtrp@plural@
\def\@Glsxtrp@plural@#1#2[#3]{\@Glsxtrp@plural@{#1}{#2}{#3}}

\@GLSxtrp@plural@
\def\@GLSxtrp@plural@#1#2[#3]{\@GLSxtrp@plural@{#1}{#2}{#3}}

\@glsxtrp@short@
\def\@glsxtrp@short@#1#2[#3]{%

```

```

        {%
        \glsssetabbrvfmt{\glscategory{#2}}%
        \glsabbrvfont{\glentryshort{#2}}#3%
        }%
    }

\@Glsxtr@p@short@
\def\@Glsxtr@p@short@#1#2[#3]{%
    {%
        \glsssetabbrvfmt{\glscategory{#2}}%
        \glsabbrvfont{\glentryshort{#2}}#3%
    }%
}

\@GLSxtr@p@short@
\def\@GLSxtr@p@short@#1#2[#3]{%
    {%
        \glsssetabbrvfmt{\glscategory{#2}}%
        \glsupercase{\glsabbrvfont{\glentryshort{#2}}#3}%
    }%
}

\@glsxtr@p@shortpl@
\def\@glsxtr@p@shortpl@#1#2[#3]{%
    {%
        \glsssetabbrvfmt{\glscategory{#2}}%
        \glsabbrvfont{\glentryshortpl{#2}}#3%
    }%
}

\@Glsxtr@p@shortpl@
\def\@Glsxtr@p@shortpl@#1#2[#3]{%
    {%
        \glsssetabbrvfmt{\glscategory{#2}}%
        \glsabbrvfont{\glentryshortpl{#2}}#3%
    }%
}

\@GLSxtr@p@shortpl@
\def\@GLSxtr@p@shortpl@#1#2[#3]{%
    {%
        \glsssetabbrvfmt{\glscategory{#2}}%
        \glsupercase{\glsabbrvfont{\glentryshortpl{#2}}#3}%
    }%
}

\@glsxtr@p@long@
\def\@glsxtr@p@long@#1#2[#3]{\{\glentrylong{#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]{%
  {\glsupercase{\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]{%
  {\glsupercase{\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]{%
  {\glsupercase{\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]{%
  {\glsupercase{\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]{%
  {\glsupercase{\@glsentrylong{#2}}#3}}}

```

```

\@glsxtrp@acrlongpl@
\def\@glsxtrp@acrlongpl@#1#2[#3]{\@glsentrylongpl{#2}#3}}

\@Glsxtrp@acrlongpl@
\def\@Glsxtrp@acrlongpl@#1#2[#3]{\@Glsentrylongpl{#2}#3}}

\@GLSxtrp@acrlongpl@
\def\@GLSxtrp@acrlongpl@#1#2[#3]{%
  {\glsupercase{\@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}%
}

\@@glsxtrp
\newrobustcmd*{\@@glsxtrp}[2]{%
Add scope.
  {%
    \let\glspostlinkhook\relax
    \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*{\glxtr@headentry@p}[2]{%
\glxtrifheaduc{#1}%
{%
\glssupercase{\@gls@entry@field{#1}{#2}}%
}%
{%
\@gls@entry@field{#1}{#2}%
}%
}

```

`\glxtrp` Not robust as it needs to expand somewhat.

```

\newcommand{\glxtrp}[2]{%
\protect\NoCaseChange
{%
\protect\glstexorpdfstring
{%
\protect\glxtrifinmark
{%
\ifcsdef{glxtrhead#1}%
{%
{\protect\csuse{glxtrhead#1}{#2}}%
}%
{%
\glxtr@headentry@p{#2}{#1}%
}%
}%
}%
{%
\@glxtrp{#1}{#2}%
}%
}%
\protect\@gls@entry@field{#2}{#1}%
}%
}%
}

```

Provide short synonyms for the most common option.

```

\glsp
\newcommand*{\glsp}{\glxtrp{short}}

\glsp
\newcommand*{\glsp}{\glxtrp{text}}

```

`\Glsxtrp` As above but use first letter upper case.

```

\newcommand{\Glsxtrp}[2]{%
\protect\NoCaseChange
{%
\protect\glstexorpdfstring
{%

```

```

\protect\glxtrifinmark
{%
  \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{\glxtrp}{\Glsxtrp}

```

\GLSxtrp As above but all upper case. The bookmarks use \glssupercase, which is expandable as from mfirstuc v2.08+.

```

\newcommand{\GLSxtrp}[2]{%
  \protect\NoCaseChange
  {%
    \protect\glstexorpdfstring
    {%
      \protect\glxtrifinmark
      {%
        \ifcsdef{GLSxtr#1}%
        {%
          {\protect\GLSxtrshort[noindex,hyper=false]{#1}[]}%
        }%
        {%
          \protect\glssupercase
          {%
            \protect\@Gls@entry@field{#2}{#1}%
          }%
        }%
      }%
    }%
  }%
  {%
    \@GLSxtrp{#1}{#2}%
  }%
}%
{%
  \protect\GLSxtrusefield{#2}{#1}%
}

```



```

    }%
  }%
}
\glsmfublocker{\GLSxtrp}

```

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 `\cgl`s 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 be 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
      }%
    }

\@GlsXtrStartUnsetBuffering Starred version checks for duplicates.
\newcommand*{\@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]{%
  \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\xifinlist\expandafter{#1}{\@glsxtr@unset@buffer}{}%
  {\listxadd\@glsxtr@unset@buffer{#1}}%
}

\@glsxtrbuffer@check@repeats
\newcommand*{\@glsxtrbuffer@check@repeats}{}

\@glsxtrbuffer@check@repeats
\newcommand*{\@glsxtrbuffer@check@repeats}{}

\@glsxtrbuffer@check@repeats@notused
\newcommand*{\@glsxtrbuffer@check@repeats@notused}{}

\@glsxtrbuffer@do@check@repeat
\newrobustcmd*{\@glsxtrbuffer@do@check@repeat}{%
  \expandafter\xifinlist\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{}%
}

\unsetBufferDisableRepeatLocal
\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\@glstr@unset@buffer
}
```

`\@glsllocalunset` Local unset.

```
\renewcommand*\@glsllocalunset}[1]{%
  \@@glsllocalunset{#1}%
  \glstrpostlocalunset{#1}%
}%
```

`\glstrpostlocalunset`

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

`\@glrsreset` Global reset.

```
\renewcommand*\@glrsreset}[1]{%
  \@@glrsreset{#1}%
  \glstrpostreset{#1}%
}%
```

`\glstrpostreset`

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

`\@glsllocalreset` Local reset.

```
\renewcommand*\@glsllocalreset}[1]{%
  \@@glsllocalreset{#1}%
  \glstrpostlocalreset{#1}%
}%
```

`\glstrpostlocalreset`

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

`\glsllocalreseteach` Locally reset a list of entries.

```
\newcommand*\glsllocalreseteach}[1]{%
  \gl@ifnotmeasuring
  {%
    \@for\@gl@thislabel:=#1\do{%
      \gl@ifexists{\@gl@thislabel}%
      {%
        \@glsllocalreset{\@gl@thislabel}%
      }%
    }%
  }%
}
```

`\glsllocalunseteach` Locally unset a list of entries.

```
\newcommand*\glsllocalunseteach}[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}{\cglsl}%
\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}}%
}

```

`\@glsxtr@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
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\@cgl@=\@cgl@
\let\@cgl@=\@cgl@

\let\@cgl@=\@cgl@
\let\@cgl@=\@cgl@
\let\@cgl@=\@cgl@
\let\@cgl@=\@cgl@

```

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}{%
\gls@hasattribute{\@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`

`\glsxtrifcounttrigger{<label>}{<trigger format>}{<normal>}`

```

\newcommand*{\glsxtrifcounttrigger}[3]{%
  \glshasattribute{#1}{entrycount}%
  {%
    \ifnum\glsentryprevcount{#1}>\glsgetattribute{#1}{entrycount}\relax
    #3%
    \else
    #2%
    \fi
  }%
  {#3}%
}

```

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

`\@@cgl`s@

```

\def\@@cgl#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
  {%
    \cglformat{#2}{#3}%
    \glunset{#2}%
  }%
  {%
    \@gls@{#1}{#2}[#3]%
  }%
}%

```

`\@@cgl`sp1@

```

\def\@@cglsp1#1#2[#3]{%
  \glsxtrifcounttrigger{#2}%
}

```



```

        {%
        \cglspformat{#2}{#3}%
        \glset{#2}%
        }%
        {%
        \@glsp1@{#1}{#2}[#3]%
        }%
    }%

\@@cGls@
\def\@@cGls@#1#2[#3]{%
    \glsetcounttrigger{#2}%
    {%
        \cGlsformat{#2}{#3}%
        \glset{#2}%
    }%
    {%
        \@Gls@{#1}{#2}[#3]%
    }%
}%

\@@cGlsp1@
\def\@@cGlsp1@#1#2[#3]{%
    \glsetcounttrigger{#2}%
    {%
        \cGlspformat{#2}{#3}%
        \glset{#2}%
    }%
    {%
        \@Glsp1@{#1}{#2}[#3]%
    }%
}%

\@@cGLS@
\def\@@cGLS@#1#2[#3]{%
    \glsetcounttrigger{#2}%
    {%
        \cGLSformat{#2}{#3}%
        \glset{#2}%
    }%
    {%
        \@GLS@{#1}{#2}[#3]%
    }%
}%

\@@cGLSp1@
\def\@@cGLSp1@#1#2[#3]{%
    \glsetcounttrigger{#2}%
    {%

```

```

\cGLSplformat{#2}{#3}%
\glunset{#2}%
}%
{%
\@GLSpl@{#1}{#2}[#3]%
}%
}%

```

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

```

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

\@cGl's@
\def\@cGl's@#1#2[#3]{\@GL's@{#1}{#2}[#3]}

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

\@cGl'spl@
\def\@cGl'spl@#1#2[#3]{\@GL'spl@{#1}{#2}[#3]}

```

Add all upper case versions not provided by glossaries.

```

\cGLS
\newrobustcmd*{\cGLS}{\@gl's@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 argument is the label, the second argument is the insert text.

```

\newcommand*{\cGLSformat}[2]{%
\expandafter\gl'suppercase\expandafter{\cgl'sformat{#1}{#2}}%
}

```

```

\cGLSpl
\newrobustcmd*{\cGLSpl}{\@gl's@hyp@opt\@cGLSpl}
\glsmfublocker{\cGLSpl}

```

`\@cGLSp1` Defined the un-starred form. Need to determine if there is a final optional argument

```

\newcommand*\@cGLSp1}[2][{}]{%
  \new@ifnextchar[{\@cGLSp1@{#1}{#2}}{\@cGLSp1@{#1}{#2}[]}%
}

\@cGLSp1@
\def\@cGLSp1@#1#2[#3]{\@GLSp1@{#1}{#2}[#3]}

```

`\cGLSp1format` Format used by `\cGLSp1` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```

\newcommand*\cGLSp1format}[2]{%
  \expandafter\glsuppercase\expandafter{\cglsp1format{#1}{#2}}%
}

```

Modify the trigger formats to check for the regular attribute.

`\cglsp1format`

```

\renewcommand*\cglsp1format}[2]{%
  \glsp1ifregular{#1}
  {\glsp1entryfirst{#1}}%
  {\ifglsp1haslong{#1}{\glsp1entrylong{#1}}{\glsp1entryfirst{#1}}}\#2%
}

```

`\cGlsformat`

```

\renewcommand*\cGlsformat}[2]{%
  \glsp1ifregular{#1}
  {\Glsentryfirst{#1}}%
  {\ifglsp1haslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}}\#2%
}

```

`\cglsp1plformat`

```

\renewcommand*\cglsp1plformat}[2]{%
  \glsp1ifregular{#1}
  {\glsp1entryfirstplural{#1}}%
  {\ifglsp1haslong{#1}{\glsp1entrylongpl{#1}}{\glsp1entryfirstplural{#1}}}\#2%
}

```

`\cGlsplformat`

```

\renewcommand*\cGlsplformat}[2]{%
  \glsp1ifregular{#1}
  {\Glsentryfirstplural{#1}}%
  {\ifglsp1haslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}}\#2%
}

```

New code similar to above for unit counting.

ossaryentry@defunitcounters

```
\newcommand*{\@newglossaryentry@defunitcounters}{%
  \protected@edef\@glo@countunit{\csuse{\glstr@categoryattr@{\@glo@category @unitcount}}}%
  \ifdefvoid\@glo@countunit
  {}%
  {%
    \glstr@ifunitcounter{\@glo@countunit}%
    {}%
    {\expandafter\glstr@addunitcounter\expandafter{\@glo@countunit}}%
  }%
}
```

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

```
\newcommand*{\@glstr@unitcountlist}{}
```

\@glstr@addunitcounter

```
\newcommand*{\@glstr@addunitcounter}[1]{%
  \listadd{\@glstr@unitcountlist}{#1}%
  \ifcsundef{glstr@theunit@#1}
  {%
    \ifcsdef{theH#1}%
    {\csdef{glstr@theunit@#1}{\csuse{theH#1}}}%
    {\csdef{glstr@theunit@#1}{\csuse{the#1}}}%
  }%
  {}%
}
```

\@glstr@ifunitcounter

```
\newcommand*{\@glstr@ifunitcounter}[3]{%
  \xifinlist{#1}{\@glstr@unitcountlist}{#2}{#3}%
}
```

\@glstr@currentunitcount

```
\newcommand*\@glstr@currentunitcount[1]{%
  glo@\glsdetoklabel{#1}@currunit@\glsggetattribute{#1}{unitcount}.%
  \csuse{glstr@theunit@\glsggetattribute{#1}{unitcount}}%
}
```

\@glstr@previousunitcount

```
\newcommand*\@glstr@previousunitcount[1]{%
  glo@\glsdetoklabel{#1}@prevunit@\glsggetattribute{#1}{unitcount}.%
  \csuse{glstr@theunit@\glsggetattribute{#1}{unitcount}}%
}
```

\glsg@increment@currunitcount

```
\newcommand*{\@glsg@increment@currunitcount}[1]{%
  \glshasattribute{#1}{unitcount}%
  {%
```

```

\protected@edef\@glxstr@csname{\@glxstr@currentunitcount{#1}}%
\ifcsundef{\@glxstr@csname}%
{%
  \csgdef{\@glxstr@csname}{1}%
  \listcsxadd
  {glo@glsdetoklabel{#1}@unitlist}%
  {\glsggetattribute{#1}{unitcount}.%
   \csuse{glxstr@theunit@glsggetattribute{#1}{unitcount}}}%
  }%
}%
{%
  \csxdef{\@glxstr@csname}%
  {\number\numexpr\csname\@glxstr@csname\endcsname+1}%
}%
}%
{}%
}

```

cal@increment@currunitcount

```

\newcommand*{\@glx@local@increment@currunitcount}[1]{%
  \glshasattribute{#1}{unitcount}%
  {%
    \protected@edef\@glxstr@csname{\@glxstr@currentunitcount{#1}}%
    \ifcsundef{\@glxstr@csname}%
    {%
      \csdef{\@glxstr@csname}{1}%
      \listcseadd
      {glo@glsdetoklabel{#1}@unitlist}%
      {\glsggetattribute{#1}{unitcount}.%
       \csuse{glxstr@theunit@glsggetattribute{#1}{unitcount}}}%
      }%
    }%
    {%
      \csedef{\@glxstr@csname}%
      {\number\numexpr\csname\@glxstr@csname\endcsname+1}%
    }%
  }%
  {}%
}

```

\@glxstr@currunitcount

```

\newcommand*{\@glxstr@currunitcount}[2]{%
  \ifcsundef
  {glo@glsdetoklabel{#1}@currunit@#2}%
  {0}%
  {\csuse{glo@glsdetoklabel{#1}@currunit@#2}}%
}%

```

\@glxstr@prevunitcount

```

\newcommand*{\@glstr@prevunitcount}[2]{%
\ifcsundef
{glo@\glstetoklabel{#1}@prevunit@#2}%
{0}%
{\csuse{glo@\glstetoklabel{#1}@prevunit@#2}}%
}%

```

\glsenableentryunitcount

```

\newcommand*{\glsenableentryunitcount}{%

```

Enable new fields:

```

\appto\@newglossaryentry@defcounters{\@newglossaryentry@defunitcounters}%

```

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

```

\renewcommand*{\glstetoklabel@newglossaryentry}{%
\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*{\glstentrycurrunitcount}[1]{%
\@glstr@currunitcount{##1}{\glstetoklabel{##1}{unitcount}}.%
\csuse{\glstr@theunit@\glstetoklabel{##1}{unitcount}}}%
}%
\newcommand*{\glstentryprevunitcount}[1]{%
\@glstr@prevunitcount{##1}{\glstetoklabel{##1}{unitcount}}.%
\csuse{\glstr@theunit@\glstetoklabel{##1}{unitcount}}}%
}%

```

Access total count:

```

\newcommand*{\glstentryprevtotalcount}[1]{%
\ifcsundef{glo@\glstetoklabel{##1}@prevunittotal}%
{0}%
{%
\number\csuse{glo@\glstetoklabel{##1}@prevunittotal}
}%
}%

```

Access max value:

```

\newcommand*{\glstentryprevmaxcount}[1]{%
\ifcsundef{glo@\glstetoklabel{##1}@prevunitmax}%
{0}%
{%
\number\csuse{glo@\glstetoklabel{##1}@prevunitmax}
}%
}%

```

Adjust post unset and reset:

```

\let\@glxstr@entryunitcount@org@unset\glxstrpostunset
\renewcommand*{\glxstrpostunset}[1]{%
  \@glxstr@entryunitcount@org@unset{##1}%
  \@glx@increment@currunitcount{##1}%
}%
\let\@glxstr@entryunitcount@org@localunset\glxstrpostlocalunset
\renewcommand*{\glxstrpostlocalunset}[1]{%
  \@glxstr@entryunitcount@org@localunset{##1}%
  \@glx@local@increment@currunitcount{##1}%
}%
\let\@glxstr@entryunitcount@org@reset\glxstrpostreset
\renewcommand*{\glxstrpostreset}[1]{%
  \glshasattribute{##1}{unitcount}%
  {%
    \protected@edef\@glxstr@csname{\@glxstr@currentunitcount{##1}}%
    \ifcsundef{\@glxstr@csname}%
    {}%
    {\ifglxresetcurrcount\csgdef{\@glxstr@csname}{0}\fi}%
  }%
  {%
  }%
}%
\let\@glxstr@entryunitcount@org@localreset\glxstrpostlocalreset
\renewcommand*{\glxstrpostlocalreset}[1]{%
  \@glxstr@entryunitcount@org@localreset{##1}%
  \glshasattribute{##1}{unitcount}%
  {%
    \protected@edef\@glxstr@csname{\@glxstr@currentunitcount{##1}}%
    \ifcsundef{\@glxstr@csname}%
    {}%
    {\ifglxresetcurrcount\csdef{\@glxstr@csname}{0}\fi}%
  }%
  {%
  }%
}%

```

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

```

\let\@cgl@ \@cgl@
\let\@cgl@sp@ \@cgl@sp@

\let\@cGl@ \@cGl@
\let\@cGl@sp@ \@cGl@sp@
\let\@cGL@ \@cGL@
\let\@cGL@sp@ \@cGL@sp@

```

Write information to the aux file.

```

\AtEndDocument{\@glx@write@entryunitcounts}%
\renewcommand*{\@glx@entry@unitcount}[3]{%
  \csgdef{glo\glx@detoklabel{##1}@prevunit@##3}{##2}%
}

```

```

\ifcsundef{glo@glstdetoklabel{##1}@prevunittotal}%
{\csgdef{glo@glstdetoklabel{##1}@prevunittotal}{##2}}%
{%
  \csxdef{glo@glstdetoklabel{##1}@prevunittotal}{
    \number\numexpr\csuse{glo@glstdetoklabel{##1}@prevunittotal}+##2}%
  }%
\ifcsundef{glo@glstdetoklabel{##1}@prevunitmax}%
{\csgdef{glo@glstdetoklabel{##1}@prevunitmax}{##2}}%
{%
  \ifnum\csuse{glo@glstdetoklabel{##1}@prevunitmax}<##2
    \csgdef{glo@glstdetoklabel{##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}%
    {%
      \ifglused{\@glsentry}%
      {%
        \forlistcsloop
          {\@gls@write@entryunitcounts@do}%
          {glo@glstdetoklabel{\@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
}

```

\GlsXtrEnableEntryUnitCounting 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
```

Redefine \gls etc:

```

\renewcommand*{\gls}{\cgl{s}}%
\renewcommand*{\Gls}{\cGls}%
\renewcommand*{\glspl}{\cgl{s}pl}%
\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}}%
}

```

@setentryunitcountunsetattr

```

\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@addtoacronymlists
{}%
{%
\GlossariesWarning{\string\SetGenericNewAcronym\space used
without restoring base acronym functions with
\string\RestoreAcronyms}%
}%
\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={\acronymssort{\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}{%
    \glssupercase{\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}{%
    \glssupercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*{\Glentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*{\glentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*{\Glentryfullpl}[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\@glstr@org@setacronymstyle\setacronymstyle
\let\@glstr@org@newacronymstyle\newacronymstyle

```

Save the list of acronyms in case they are required.

```

\@glstr@acronymlists

```

```

\let\@glstr@acronymlists\@glstr@acronymlists

```

```

\@glstr@org@addtoacronymlists

```

```

\let\@glstr@org@addtoacronymlists\@addtoacronymlists

```

```

\@glstr@org@setacronymlists

```

```

\let\@glstr@org@setacronymlists\SetAcronymLists

```

Need to provide a replacement for `\forallacronyms` since `\@glstr@acronymlists` isn't available.

```

\@glstr@abbrlists

```

```

\newcommand{\@glstr@abbrlists}{ }

```

\forall abbreviationlists

```
\newcommand*{\forall abbreviationlists}[2]{%
  \for#1:=\glstr@abbrlists\do{\ifempty{#1}{#2}}%
}
```

\glstr@addabbreviationlist

```
\newcommand*{\glstr@addabbreviationlist}[1]{%
  \protected@edef\glo@type{#1}%
  \ifempty\glstr@abbrlists
  {\let\glstr@abbrlists\glo@type}%
  {%
    \ifequal\glstr@abbrlists\glo@type
    {}%
    {%
      \expandafter\DTLifinlist\expandafter{\glo@type}{\glstr@abbrlists}{}%
      {\protected@eappto\glstr@abbrlists{,\glo@type}}%
    }%
  }%
}
```

\forall acronyms Modify to add warning.

```
\renewcommand*{\forall acronyms}[2]{%
  \glstr@base@acrcmd\forall acronyms\forall abbreviationlists
  \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 \newabbreviationstyle 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\glstr@acronymlists\glsacronymlists
\let\glsacronymlists@empty
\let\addtoacronymlists@gobble
\let\SetAcronymLists@gobble
```

Warn if \acrshort etc are used.

```
\let\glstr@base@acrcmd\@glstr@base@acrcmd@warn
```

Redefine \newacronym to use same interface as \newabbreviation.

```
\renewcommand*{\newacronym}[4][{}]{%
  \glstr@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@addtoacronymlists
\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]{%
        \settowidth{\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]{%
        \settowidth{\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}{\@gls@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@noprntglossary\@glsxtr@warn@hybrid@noprntgloss
\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\@glstr@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\@glstr@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\glstr@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 subsequent 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}}%
  }%
}%
}%
}%
}%

```

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@noprntglossary\@makeglossaries@warn@noprntglossary

```

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}}%
}

```

```

{\@glstr@noidx@displaynumberlist{##1}}%
}%

```

Adjust entry list:

```

\renewcommand*{\glstentrynumberlist}[1]{%
\expandafter\DTLifinlist\expandafter{##1}{\@glstr@reg@glosslist}%
{\@glstr@idx@entrynumberlist{##1}}%
{\@glstr@noidx@entrynumberlist{##1}}%
}%

```

Adjust number list loop

```

\renewcommand*{\glsnumberlistloop}[2]{%
\expandafter\DTLifinlist\expandafter{##1}{\@glstr@reg@glosslist}%
{%
\PackageError{glossaries-extra}{\string\glsnumberlistloop\space
not available for glossary '##1'}{%
}%
{\@glstr@noidx@numberlistloop{##1}{##2}}%
}%

```

Only sanitize sort for normal indexing glossaries.

```

\renewcommand*{\glsprestandardsort}[3]{%
\expandafter\DTLifinlist\expandafter{##2}{\@glstr@reg@glosslist}%
{%
\glsdosanitizesort
}%
{%
\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\@glstr@mixed@assign@sortkey
\renewcommand*\@printgloss@setsort{%

```

Need to extract just the type value.

```

\expandafter\@glstr@gettype\expandafter,\@glstr@printglossopts,%
type=\glsdefaulttype,\@end@glstr@gettype
\def\@glo@sorttype{\@glo@default@sorttype}%
}%

```

Check automake setting:

```

\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\@for\@gls@type:=\@glstr@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}{}%
\fi
}%

```

Prohibit the use of \glstrnoidxgroups.

```

\prohibit@glstrnoidxgroups

```

Activate warnings for incompatible options.

```

\let\gls@warn@makegloss@incompatible\@gls@warn@makegloss@incompatible
}%
}%
}

```

\@gls@warn@makegloss@incompatible

```

\newcommand*{\gls@warn@makegloss@incompatible}[2]{%

```

\@gls@warn@makegloss@incompatible

```

\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 \@glstr@orgprintglossary. This was later changed to actually defining \@glstr@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 L^AT_EX 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:

`sxtr@printgloss@checkexists`

```
\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{\gldefaulttype}%
```

Add check here.

```
\def\glossarytitle{%
\ifcsdef{\@glo@type\@glo@type @title}%
{\csuse{\@glo@type\@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 fallback. \MessageBreak
To fix this set the style with \MessageBreak
\string\setglossarystyle\space or use the \MessageBreak
style key=value option}%
\fi
}%
\def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\bgroup
\@printgloss@setsort
\setkeys{printgloss}{#1}%
\ifx\glossarytitle\org@glossarytitle
\else
\cslet{\@glo@type\@glo@type @title}{\glossarytitle}%
\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages

\glsextractivatenopost
\gls@dotoc@title
\@glossarystyle
\let\gls@org@glossaryentryfield\glossentry
\let\gls@org@glossarysubentryfield\subglossentry
\renewcommand{\glossentry}[1]{%
```

```

        \protected@xdef\glscurrententrylabel{\glsetoklabel{##1}}%
        \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
        \protected@xdef\glscurrententrylabel{\glsetoklabel{##2}}%
        \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    \glstr@printgloss@checkexists{\@gls@type}{#2}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprntglossary\relax
}

\glstractivatenopost Change \nopostdesc and \glstrnopostpunc to behave as they do in the
glossary.
\newcommand*{\glstractivatenopost}{%
    \let\nopostdesc\@nopostdesc
    \let\glstrnopostpunc\@glstr@nopostpunc
}

\glstrnopostpunc
\newrobustcmd*{\glstrnopostpunc}{}

\@glstr@nopostpunc Provide a command that works like \nopostdesc but only switches off the
punctuation without suppressing the post-description hook.
\newcommand{\@glstr@nopostpunc}{%
    \let\@glstr@org@postdescription\glspostdescription
    \ifglsnopostdot
        \renewcommand{\glspostdescription}{%
            \glsnopostdottrue
            \let\glspostdescription\@glstr@org@postdescription
            \let\glstrrestorepostpunc\@glstr@restore@postpunc
            \glstrpostdescription
            \@glstr@nopostpunc@postdesc}%
        \else
            \renewcommand{\glspostdescription}{%
                \let\glspostdescription\@glstr@org@postdescription
                \let\glstrrestorepostpunc\@glstr@restore@postpunc
                \glstrpostdescription
                \@glstr@nopostpunc@postdesc}%
            \fi
        \glsnopostdotfalse
    }

\@glstr@nopostpunc@postdesc
\newcommand*{\@glstr@nopostpunc@postdesc}{}

\@glstr@restore@postpunc

```

```

\newcommand*{\@glxtr@restore@postpunc}{%
\def\@glxtr@nopostpunc@postdesc{%
\@glxtr@org@postdescription
\let\@glxtr@nopostpunc@postdesc\@empty
\let\glxtrrestorepostpunc\@empty
}%
}

```

`\glxtrrestorepostpunc` Does nothing outside of glossary.

```

\newcommand*{\glxtrrestorepostpunc}{}

```

`\@printglossary` Redefine.

```

\renewcommand{\@printglossary}[2]{%
\def\@glxtr@printglossopts{#1}%
\@glxtr@orgprintglossary{#1}{#2}%
}

```

Add a key that switches off the entry targets:

```

\define@choicekey{printgloss}{target}
[{\@glxtr@printglossval\@glxtr@printglossnr}%
{true,false}[true]%
{%
\ifcase\@glxtr@printglossnr

\def\@glstarget{\@glsdohypertarget}%
\else
\let\@glstarget\@secondoftwo
\fi
}

```

`\@glxtrhypernameprefix`

```

\newcommand{\@glxtrhypernameprefix}{}

```

New to v1.20:

```

\define@key{printgloss}{targetnameprefix}{%
\renewcommand{\@glxtrhypernameprefix}{#1}%
}

```

```

\define@key{printgloss}{prefix}{%
\renewcommand{\@glolinkprefix}{#1}%
}

```

```

\define@key{printgloss}{label}{%
\glxtrsetglossarylabel{#1}%
}

```

```

\define@key{printgloss}{preamble}{%
\renewcommand{\@glossarypreamble}{#1}%
}

```

```

\define@key{printgloss}{postamble}{%
  \renewcommand{\glossarypostamble}{#1}%
}

\glstrsetglossarylabel 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{\glstrsetglossarylabel}[1]{%
  \renewcommand*{\@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{#1}%
  }%
}

\@glstr@leveloffset

\newcount\@glstr@leveloffset

New to v1.44:
\define@key{printgloss}{leveloffset}{%
  \@glstr@assign@leveloffset#1\relax
  \gls@warn@noidxmakegloss@incompatible{option 'leveloffset'}
  {\@glstr@leveloffset=0\relax}%
}

\@glstr@assign@leveloffset

\newcommand*{\@glstr@assign@leveloffset}{%
  \@ifnextchar+{\p@glstr@assign@leveloffset}{\np@glstr@assign@leveloffset}%
}

\p@glstr@assign@leveloffset Discard initial "+" character.

\newcommand*{\p@glstr@assign@leveloffset}[1]{%
  \@ifnextchar+{\pp@glstr@assign@leveloffset}{\np@glstr@assign@leveloffset}%
}

\np@glstr@assign@leveloffset

\def\np@glstr@assign@leveloffset#1\relax{\@glstr@leveloffset=#1\relax}

\pp@glstr@assign@leveloffset

\def\pp@glstr@assign@leveloffset#1\relax{\advance\@glstr@leveloffset by #1\relax}

\define@boolkey{printgloss}[glstr@printgloss@]{groups}[true]{%
  \ifglstr@printgloss@groups
  \else
  \gls@warn@noidxmakegloss@incompatible{option 'groups'}%
  {\glstr@printgloss@groupstrue}%
  \fi
}
\glstr@printgloss@groupstrue

```

```

\define@boolkey{printgloss}[glxtrprintgloss]{flatten}[true]{%
\ifglxtrprintglossflatten
\glswarn@noidxmakegloss@incompatible{option 'flatten'}%
{\glxtrprintglossflattenfalse}%
\fi
}
\glxtrprintglossflattenfalse

```

`\glsdohypertarget` Redefine to insert `\glxtrhypernameprefix` before the target name.

```

\let\@glxtr@org@glsdohypertarget@glsdohypertarget
\renewcommand{\glsdohypertarget}[2]{%
\@glxtr@org@glsdohypertarget{\glxtrhypernameprefix#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\@glxtr@org@glsdohypertarget
\def\glstarget{\glsdohypertarget}%
\fi

```

`\@glxtr@do@org@target` Provide a way to locally do the original.

```

\newcommand{\@glxtr@do@org@target}[2]{%
{%
\let\glsdohypertarget\@glxtr@org@glsdohypertarget
\glstarget{#1}{#2}%
}%
}

```

`\glxtr@makeglossaries` For the benefit of `makeglossaries`

```

\newcommand*{\glxtr@makeglossaries}[1]{}

```

`\@glxtr@gettype` Get just the type.

```

\def\@glxtr@gettype#1,type=#2,#3\@end@glxtr@gettype{%
\def\@glo@type{#2}%
}

```

`\@glxtr@mixed@assign@sortkey` Assign the sort key.

```

\newcommand{\@glxtr@mixed@assign@sortkey}[1]{%

\protected@edef\@glo@type{\@glo@type}%
\expandafter\DTLifinlist\expandafter{\@glo@type}{\@glxtr@reg@glosslist}%
{%
\@glo@no@assign@sortkey{#1}%
}%
{%
\@glo@assign@sortkey{#1}%
}%
}%

```

Display number list for the regular version:

`\glstr@idx@displaynumberlist`

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

Display number list for the “noidx” version:

`\glstr@noidx@displaynumberlist`

```
\newcommand*{\@glstr@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
  }%
  {%

    \glstrundeftag
    \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:

`\@glstr@noidx@numberlistloop`

```
\newcommand*{\@glstr@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}%
  }%
  {%
```

```

\glxtrundeftag
\glsoifexists{#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.

\glxtr@noidx@entrynumberlist

```

\newcommand*{\@glxtr@noidx@entrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{#1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    \glxtrundeftag
    \glsoifexists{#1}%
    {%
      \GlossariesWarning{Missing location list for ‘#1’. Either
        a rerun is required or you haven’t referenced the entry.}%
    }%
  }%
}%

```

\@glxtr@idx@entrynumberlist

```

\newcommand*{\@glxtr@idx@entrynumberlist}[1]{\glsentrynumberlist{#1}}

```

\@gls@noidx@getgrouptitle Patch.

```

\renewcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \protected@edef\@glxtr@titlelabel{#1}%
  \ifdefvoid\@glxtr@titlelabel
  {%
    \protected@edef\@glxtr@titlelabel{\csuse{\glxtr@grouptitle@#1}}%
  }%
  \ifdefvoid{\@glxtr@titlelabel}%
  {%
    \DTLifint{#1}%
    {%
      \ifnum#1<256\relax
        \edef#2{\char#1\relax}%
      \else
        \edef#2{#1}%
      \fi
    }%
  }%
}

```

```

        \fi
      }%
    {%
      \ifcsundef{#1groupname}%
      {\def#2{#1}}%
      {\letcs#2{#1groupname}}%
    }%
  }%
  {%
    \let#2\@glxtr@titlelabel
  }%
}

\glxtr@org@getgrouptitle Save original definition of \@gls@getgrouptitle
\let\glxtr@org@getgrouptitle\@gls@getgrouptitle

\glxtrnoidxgroups Provide the ability to switch from unsrt to noidx code, but only for record=off.
\newcommand*{\glxtrnoidxgroups}{%
  \ifdefequal\@glxtr@record@setting\@glxtr@record@setting@off
  {%
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\glxtr@org@getgrouptitle\@gls@getgrouptitle
  }%
  {\PackageError{glossaries-extra}{Can't use
    \string\glxtrunsrtgrouptonoidx\space with record=\@glxtr@record@setting}
  {\string\glxtrunsrtgrouptonoidx\space is only available with record=off}}%
  \global\let\prohibit@glxtrnoidxgroups\@glxtrnoidxgroups@nomakegloss
}

\glxtrnoidxgroups@nomakegloss
\newcommand{\@glxtrnoidxgroups@nomakegloss}{%
  \PackageError{glossaries-extra}{Can't use
    \string\glxtrunsrtgrouptonoidx\space with \string\makeglossaries}{%
  }

\prohibit@glxtrnoidxgroups
\newcommand{\prohibit@glxtrnoidxgroups}{%
  \global\let\glxtrnoidxgroups\@glxtrnoidxgroups@nomakegloss
}

\glxtrgetgrouptitle 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{\glxtrgetgrouptitle}[2]{%
  \protected@edef\@glxtr@titlecsname{\glxtr@group@title@#1}%
  \@onelevel@sanitize\@glxtr@titlecsname
  \ifcsdef{\@glxtr@titlecsname}
  {\letcs{#2}{\@glxtr@titlecsname}}%
  {\glxtr@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}%
    }

\glsnavigation Redefine to use new user-level command.
    \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
    \renewcommand*{\@print@noidx@glossary}{%
      \ifcsdef{\@gls@ref@\@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.

```
\glstrifemptyglossary{\@glo@type}%
{}%
{\glossarysection[\glossarytoctitle]{\glossarytitle}}%
\@gls@noref@warn{\@glo@type}%
}%
}
```

`\glsnoidxdisplayloc` Patch to check for range formations.

```
\renewcommand*{\glsnoidxdisplayloc}[4]{%
\setentrycounter[#1]{#2}%
\@glstr@display@loc#3\empty\end@glstr@display@loc{#4}%
}
```

`\@glstr@display@loc` Patch to check for range formations.

```
\def\@glstr@display@loc#1#2\end@glstr@display@loc#3{%
\ifx#1\relax
\glstrdisplaystartloc{#2}{#3}%
\else
\ifx#1\relax
\glstrdisplayendloc{#2}{#3}%
\else
\glstrdisplaysingleloc{#1#2}{#3}%
\fi
\fi
}
```

`\glstrdisplaysingleloc` Single location.

```
\newcommand*{\glstrdisplaysingleloc}[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 `\glstrlocrangefmt`.

```

\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\glsetentrycounter{#2}%
    }

\@gls@removespaces Redefine to allow adjustments to location hyperlink.
    \def\@gls@removespaces#1 #2\@nil{%
        \toks@=\expandafter{\the\toks@#1}%
        \ifx\@#2\@nil
            \edef\@glo@tmp{\the\toks@}%
            \ifx\@glo@tmp\empty
                \else

```

Expand location (just in case \toks@ is needed for something else).

```

\expandafter\glxstrlocationhyperlink\expandafter
\glstrycounter\expandafter\@glo@counterprefix\expandafter{\the\toks@}%
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}

```

`\glxstrlocationhyperlink{<counter>}{<prefix>}{<location>}`

`\glxstrlocationhyperlink`

```

\newcommand*{\glxstrlocationhyperlink}[3]{%
\ifvoid\glxstrsupplocationurl
{%
\GlsXtrInternalLocationHyperlink{#1}{#2}{#3}%
}%
{%
\hyperref{\glxstrsupplocationurl}{#1#2#3}{#3}%
}%
}

```

`\glxstrsupphypernumber`

```

\newcommand*{\glxstrsupphypernumber}[1]{%
{%
\glshasattribute{\glscurrententrylabel}{externalallocation}%
{%
\def\glxstrsupplocationurl{%
\glsgetaattribute{\glscurrententrylabel}{externalallocation}}%
}%
{%
\def\glxstrsupplocationurl{%
}%
\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}%
{%
\glxstrNoGlossaryWarning{\@glo@type}}%

```

```

\ifglxindy
\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}%
      }%
    }%
  }%
}%
\@do@auxoutstuff
\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
\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
  \ifglxindy 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}
}

```

```

\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 \ifglxsindy 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]%
        \glssopenbrace glossaries-extra\glsscbrace}

    \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.%
}

\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.%
}
```

| tr@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

```

```

}%
}%

\ifdequal\makeglossaries\@no@makeglossaries
{%
  \GlsXtrNoGlsWarningMisMatch
}%
{%
  \GlsXtrNoGlsWarningBuildInfo
}%
\fi
}%
{%
  \GlsXtrNoGlsWarningNoOut
  {\jobname.\csname @glotype@\@glo@type @out\endcsname}%
}%
\par
\GlsXtrNoGlsWarningTail
}

```

`\record@noglossarywarning` Warn about using `\printglossary` with `record`

```

\newcommand*{\@glxtr@record@noglossarywarning}[1]{%
  \GlossariesExtraWarning{\string\printglossary\space doesn't work\MessageBreak
with record=\@glxtr@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}{}

```

`\glxtrresourcefile` 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*{\glxtrresourcefile}[2][ ]{%
  \@glxtr@if@record@only
  {\renewcommand{\glsindexingsetting}{bib2gls}}%
  {\edef\glsindexingsetting{bib2gls-\ifglxindy xindy\else makeindex\fi}}%
}

```

The `record` option can't be set after this command.

```

\disable@keys{glossaries-extra.sty}{record}%
\glxtr@writefields
\glxtr@save@mfu
\ifdefempty\GlsXtrDefaultResourceOptions
{%

```

```

\protected@write\@auxout{\glxtrresourceinit}%
{\string\glxtr@resource{#1}{#2}}%
}%
{%
\protected@write\@auxout{\glxtrresourceinit}%
{\string\glxtr@resource{\GlsXtrDefaultResourceOptions,#1}{#2}}%
}%
\let\@glxtr@org@see@noindex\@gl@see@noindex
\let\@gl@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\@bibgl@restoreat{\noexpand\catcode\noexpand'\noexpand\@=\number\catcode'\@}%
\makeatletter
\@input{#2.glstex}%
\@bibgl@restoreat

```

If the record=nameref option has been set, check if this is supported by the installed version of bib2gls.

```

\@glxtr@check@bibgl@nameref
}%
{%
\GlossariesExtraWarning{No file '#2.glstex'}%
}%
\let\@gl@see@noindex\@glxtr@org@see@noindex
}
\@onlypreamble\glxtrresourcefile

```

\@glxtr@check@bibgl@nameref This will only warn after bib2gls has created the .glstex file, but there's way to check before.

```

\newcommand{\@glxtr@check@bibgl@nameref}{%
\ifx\@glxtr@record@setting\@glxtr@record@setting@nameref
\ifdef\bibglshrefchar
{}%
{%
\GlossariesExtraWarning{record=nameref requires at least
version 1.8 of bib2gls}%
}%
\fi
\let\@glxtr@check@bibgl@nameref\relax
}

```

\glxtrresourceinit Code used during the protected write operation.

```

\newcommand*{\glxtrresourceinit}{}

```

\glxtrresourcecount

```

\newcount\glxtrresourcecount

```

`\GlsXtrLoadResources` Short cut that uses `\glstrresourcefile` with `\jobname` as the mandatory argument.

```
\newcommand*{\GlsXtrLoadResources}[1][]{%
  \ifnum\glstrresourcecount=0\relax
    \glstrresourcefile[#1]{\jobname}%
  \else
    \glstrresourcefile[#1]{\jobname-\the\glstrresourcecount}%
  \fi
  \advance\glstrresourcecount by 1\relax
}
```

`\glstr@resource`

```
\newcommand*{\glstr@resource}[2]{}%
```

`\glstrMFUsave`

```
\newcommand*{\glstrMFUsave}{%
  \ifdef\MFUsave
    {%
      \AtBeginDocument{\MFUsave}%
    }%
  {%
    \GlossariesExtraWarning{mfirstuc.sty too old,
      \string\glstrMFUsave\space has no effect. You need to upgrade
      to mfirstuc v2.08}%
  }%
  \let\glstrMFUsave\relax
}
```

`\glstr@save@mfu`

```
\ifdef\MFUsave
{
  \newcommand*{\glstr@save@mfu}{%
    \glstrMFUsave
    \let\glstr@save@mfu\relax
  }
}
{
  \newcommand*{\glstr@save@mfu}{}
}
```

`\glstr@fields`

```
\newcommand*{\glstr@fields}[1]{}%
```

`\glstr@texencoding`

```
\newcommand*{\glstr@texencoding}[1]{}%
```

`\glstr@langtag`

```
\newcommand*{\glstr@langtag}[1]{}%
```

```

\glsxtr@pluralsuffixes
    \newcommand*{\glsxtr@pluralsuffixes}[4]{}

\glsxtr@shortcutsval
    \newcommand*{\glsxtr@shortcutsval}[1]{}

\glsxtr@linkprefix
    \newcommand*{\glsxtr@linkprefix}[1]{}

\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 documents, the required locale will have to be indicated in the "sort" key when using `\glsxtrresourcefile`.

```

\ifdef\CurrentTrackedLanguageTag
{%
    \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{\gls@linkprefix}}%
\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}%
{\immediate\write18{bib2gls \jobname}}}%
```

If `\makeglossaries` is also used, allow `makeindex/xindy` to also be run, otherwise disable the error message about requiring `\makeglossaries` with `automake`.

```
\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.

`\glxtr@record`

```
\newcommand*{\glxtr@record}[5]{}
```

`\glxtr@record@nameref` Used with `record=nameref` to include current label information.

```
\newcommand*{\glxtr@record@nameref}[8]{}
```

`\glxtr@counterrecord` Aux file command.

```

\newcommand*{\glxtr@counterrecord}[3]{%
    \glxtrfieldlistgadd{#1}{record.#2}{#3}%
    \glxtrAddCounterRecordHook{#1}{#2}{#3}%
}

```

`\glxtrAddCounterRecordHook` User hook.

```
\newcommand*{\glxtrAddCounterRecordHook}[3]{}
```

`\@glxtr@counterrecordhook` Hook used by `\@glxtr@dorecord`.

```
\newcommand*{\@glxtr@counterrecordhook}{}%
```

`\GlsXtrRecordCounter` Activate recording for a particular counter (identified in the argument).

```

\newcommand*{\GlsXtrRecordCounter}[1]{%
    \@glxtr@recordcounter{#1}%
}
\@onlypreamble\GlsXtrRecordCounter

```

`\@glxtr@docounterrecord`

```

\newcommand*{\@glxtr@docounterrecord}[1]{%
    \@bibgls@write@aux{}{\string\glxtr@counterrecord
        {\@gls@label}{#1}{\csuse{the#1}}}%
}

```

`\glxtrglossentry` 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*{\glxtrglossentry}[1]{%
```

```

\glstrtitleorpdforheading
{\@glstrglossentry{#1}}%
{\GlsXtrStandaloneEntryPdfName{#1}}%
{\GlsXtrStandaloneEntryHeadName{#1}}%
}

```

`\@glstrglossentry` Another test is needed in case `\@glstrglossentry` has been written to the table of contents.

```

\newrobustcmd*{\@glstrglossentry}[1]{%
\glstrtitleorpdforheading
{%
\glsoifexists{#1}%
{%
\begingroup

\protected@edef\glscurrententrylabel{\glsetoklabel{#1}}%
\protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
\ifglshasparent{#1}%
{\GlsXtrStandaloneSubEntryItem{#1}}%
{\glsetryitem{#1}}%
\GlsXtrStandaloneEntryName{#1}%
\endgroup
}%
}%
{\GlsXtrStandaloneEntryPdfName{#1}}%
{\GlsXtrStandaloneEntryHeadName{#1}}%
}

```

`\GlsXtrStandaloneEntryHeadName`

```

\newcommand*{\GlsXtrStandaloneEntryHeadName}[1]{%
\glstrheadname{#1}%
}

```

`\GlsXtrStandaloneEntryPdfName`

```

\newcommand*{\GlsXtrStandaloneEntryPdfName}[1]{%
\glsetryname{#1}%
}

```

`\GlsXtrStandaloneEntryName`

```

\newcommand*{\GlsXtrStandaloneEntryName}[1]{%
\glstarget{#1}{\glossentryname{#1}}%
}

```

`\GlsXtrStandaloneGlossaryType` To make it easier to adjust the definition of `\currentglossary` within `\glstrglossentry`, this expands to the default definition. (If redefined, it must fully expand to the appropriate label.)

```

\newcommand{\GlsXtrStandaloneGlossaryType}{\glsetrytype{\glscurrententrylabel}}

```

`\GlsXtrStandaloneSubEntryItem` Used for sub-entries in standalone format. The argument is the entry's label.

```
\newcommand*{\GlsXtrStandaloneSubEntryItem}[1]{%
  \GlsXtrIfFieldEqNum{level}{#1}{1}{\glssubentryitem{#1}}{}%
}
```

`\glsxtrglossentryother` 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]{%
  \ifstrempy{#1}%
  {%
    \glsxtrtitleorpdforheading
    {\@glsxtrglossentryother{#2}{#3}{\GlsXtrStandaloneEntryHeadOther{#3}{#2}}}%
    {\GlsXtrStandaloneEntryPdfOther{#2}{#3}}%
    {\GlsXtrStandaloneEntryHeadOther{#3}{#2}}%
  }%
  {%
    \glsxtrtitleorpdforheading
    {\@glsxtrglossentryother{#2}{#3}{#1}}%
    {\GlsXtrStandaloneEntryPdfOther{#2}{#3}}%
    {#1}%
  }%
}
```

`\GlsXtrStandaloneEntryHeadOther`

```
\newcommand*{\GlsXtrStandaloneEntryHeadOther}[2]{%
  \ifcsdef{glsxtrhead#2}%
  {\csuse{glsxtrhead#2}{#1}}%
  {\@gls@entry@field{\NoCaseChange{#1}}{#2}}%
}
```

`\GlsXtrStandaloneEntryPdfOther`

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

`\glsxtrglossentryother{<entry-label>}{<field>}{<header>}`

`\@glsxtrglossentryother`

As `\@glsxtrglossentry` but uses a different field.

```
\newrobustcmd*{\@glsxtrglossentryother}[3]{%
  \glsxtrtitleorpdforheading
  {%
    \glsdoifexists{#1}%
    {%
      \begingroup
```

```

\protected@edef\glscurrententrylabel{\glsetoklabel{#1}}%
\protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
\ifglshasparent{#1}%
{\GlsXtrStandaloneSubEntryItem{#1}}%
{\glsetentryitem{#1}}%

\GlsXtrStandaloneEntryOther{#1}{#2}%
\endgroup
}%
}%
{\GlsXtrStandaloneEntryPdfOther{#1}{#2}}%
{#3}%
}

\GlsXtrStandaloneEntryOther

\newcommand*{\GlsXtrStandaloneEntryOther}[2]{%
\glstarget{#1}{\glossentrynameother{#1}{#2}}%
}

\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=\glstypedefaulttype,#1}{\@print@unsrt@glossary}%
}

\s@printunsrtglossary Starred version.

\newcommand*{\s@printunsrtglossary}[2][]{%
\begingroup
#2%
\@printglossary{type=\glstypedefaulttype,#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

```
\glxtrifemptyglossary{\@glo@type}%
{%
  \GlossariesExtraWarning{No entries defined in glossary ‘\@glo@type’}%
}%
{%
```

Setup local commands.

```
\glxtr@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 `\glxtr@doglossary`, which will be built up in the loop.

```
\def\glxtr@doglossary{%
  \begin{theglossary}%
  \glossaryheader
  \glxtr@resetentrylist
}%
```

Iterate over all entries in the current glossary and add the relevant commands to `\glxtr@doglossary`.

```
\expandafter\@for\expandafter\glxtr@currententrylabel\expandafter
:\expandafter=\csname glist@\@glo@type\endcsname\do{%
  \ifdefempty{\glxtr@currententrylabel}
  {}%
  {%
```

Initialise hooks

```
\glxtr@initprocess
```

Process this entry (unless it has been skipped).

```
\glxtr@process
{%
  \ifglxtr@printgloss@groups
```

Check if the group heading should be added and, if so, add it. `\glxtr@groupheading` will be empty if no group heading.

```
\glxtr@addgroup\glxtr@currententrylabel
{%
  \glxtr@checkgroup\glxtr@currententrylabel
  \expandafter\appto\expandafter\glxtr@doglossary\expandafter
```

```

        {\@glstr@groupheading}%
    }%
\fi

\protected@eappto\@glstr@doglossary{%
    \noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}%
}%
}%
}%
\appto\@glstr@doglossary{\end{theglossary}}%
\printunsrtglossarypredoglossary
\@glstr@doglossary
}%
\glossarypostamble
}

\@glstr@unsrt@gloss@init Initialise hooks needed at the start.
    \newcommand*{\@glstr@unsrt@gloss@init}{%
Determine how to obtain the group information.
    \key@ifundefined{glossentry}{group}%
    {\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle}%
    {\let\@gls@getgrouptitle\@glstr@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).
    \ifglstrprintglossflatten
    \edef\glscurrententrylevel{\number\@glstr@leveloffset}%
    \else
    \edef\glscurrententrylevel{%
        \number\numexpr\csname glo@\glscurrententrylabel @level\endcsname
        + \@glstr@leveloffset}%
    \fi

```

If this level 0, update `\glscurrenttoplevelentry`

```
\ifnum\glscurrententrylevel=0\relax
  \let\glscurrenttoplevelentry\glscurrententrylabel
\fi
```

If this entry doesn't have a parent, update `\glscurrentrootentry`

```
\ifglxtrprintglossflatten
  \let\glscurrentrootentry\glscurrententrylabel
\else
  \ifglshasparent{\glscurrententrylabel}{}%
  {\let\glscurrentrootentry\glscurrententrylabel}%
\fi
```

Initialise to do the current entry.

```
\let\glxtr@process\@firstofone
```

Provide a way to skip the current entry. This will redefine `\glxtr@process` to ignore its argument.

```
\let\printunsrtglossaryskipentry\@glxtr@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\@glxtr@printglossopts{#1}%
  \def\@glo@type{\gldefaulttype}%
  \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\@glxtr@printglossopts{#1}%
    \def\@glo@type{\gldefaulttype}%
    \def\glossarytitle{\csname \@glo@type \@glo@type @title\endcsname}%
    \def\glossarytoctitle{\glossarytitle}%
    \let\org@glossarytitle\glossarytitle
    \def\@glossarystyle{%
      \ifx\@glossary@default@style\relax
```

The type key simply allows the title to be set if the title key isn't supplied.

208


```

\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprntglossary\relax
}

\@print@unsrt@innnerglossary This is much like \@print@unsrt@innnerglossary but only contains what would
normally be the content of the theglossary.
\newcommand*{\@print@unsrt@innnerglossary}{%
No section header or preamble.
\glstrifemptyglossary{\@glo@type}%
{%
\GlossariesExtraWarning{No entries defined in glossary ‘\@glo@type’}%
}%
}%
Setup local commands.
\@glstr@unsrt@gloss@init
No header or reset.
\def\@glstr@doglossary{%
Iterate over all entries in the current glossary and add the relevant commands
to \@glstr@doglossary.
\expandafter\@for\expandafter\glscurrententrylabel\expandafter
:\expandafter=\csname glolist@\@glo@type\endcsname\do{%
\ifdefempty{\glscurrententrylabel}
}%
}%
Initialise hooks
\@glstr@initprocess
Process this entry (unless it has been skipped).
\glstr@process
{%
\ifglstr@printgloss@groups
Check if the group heading should be added and, if so, add it. \@glstr@groupheading
will be empty if no group heading.
\glstraddgroup\glscurrententrylabel
{%
\@glstr@checkgroup\glscurrententrylabel
\expandafter\appto\expandafter\@glstr@doglossary\expandafter
{\@glstr@groupheading}%
}%
\fi
\protected@eappto\@glstr@doglossary{%
\noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}%
}%
}%
}%

```

```

        \printunsrtglossarypredoglossary
        \@glxtr@doglossary
    }%
No postamble.
}

\glxtraddgroup 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*{\glxtraddgroup}[2]{%
    \ifglxtrprintglossflatten
        #2%
    \else
        \ifglshasparent{#1}{#2}%
    \fi
}

\printunsrtglossaryentryprocesshook
\newcommand*{\printunsrtglossaryentryprocesshook}[1]{}

\printunsrtglossaryskipentry
\newcommand*{\printunsrtglossaryskipentry}{%
    \PackageError{glossaries-extra}{\string\printunsrtglossaryskipentry\space
can only be used within \string\printunsrtglossaryentryprocesshook}{}%
}

\printunsrtglossaryskipentry
\newcommand*{\@glxtr@printunsrtglossaryskipentry}{%
    \let\glxtr@process\@gobble
}

\printunsrtglossarypredoglossary
\newcommand*{\printunsrtglossarypredoglossary}{}

\printunsrt@glossary@handler
\newcommand*{\@printunsrt@glossary@handler}[1]{%
    \protected@xdef\glscurrententrylabel{#1}%
    \printunsrtglossaryhandler\glscurrententrylabel
}

\printunsrtglossaryhandler
\newcommand*{\printunsrtglossaryhandler}[1]{%
    \glxtrunsrtdo{#1}%
}

```

```
\glxtriflabelinlist{\label}\list\true\false}
```

`\glxtriflabelinlist`

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*\glxtriflabelinlist[4]{%
\protected@edef\@glxtr@doiflabelinlist{\noexpand\gls@ifinlist{#1}{#2}}%
\@glxtr@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]{%
\glxtrfieldxifinlist{##1}{record.#1}{\csuse{the#1}}
{\glxtrunsrtdo{##1}}%
}%
}
```

Only the target names should have the prefixes adjusted as `\gls` etc need the original `\glslinkprefix`. The `\@gobble` part discards `\glslinkprefix`.

```
\ifcsundef{theH#1}%
{%
\renewcommand*\@glxtrhypernameprefix{record.#1.\csuse{the#1}.\@gobble}%
}%
{%
\renewcommand*\@glxtrhypernameprefix{record.#1.\csuse{theH#1}.\@gobble}%
}%
\renewcommand*\glossarysection[2][{}]{%
\appto\glossarypostamble{\printunsrtglossaryunitpostskip}%
}
```

`\printunsrtglossaryunitpostskip`

```
\newcommand*\printunsrtglossaryunitpostskip{\glspare\medskip\glspare}
```

`\print@noop@unsrtglossaryunit`

```
\newcommand{\print@noop@unsrtglossaryunit}[2][{}]{%
\PackageError{glossaries-extra}{\string\printunsrtglossaryunit\space
requires the record=only or record=alsoindex package option}{}%
}
```

`\@glstr@unsrt@getgrouptitle`

```
\newrobustcmd*{\@glstr@unsrt@getgrouptitle}[2]{%
  \protected@edef\@glstr@titlelabel{glstr@grouptitle@#1}%
  \@onelevel@sanitize\@glstr@titlelabel
  \ifcsdef{\@glstr@titlelabel}
  {\letcs{#2}{\@glstr@titlelabel}}%
  {\def#2{#1}}%
}
```

`\glstrunsrtdo` Provide a user-level call to `\@glstr@noidx@do` to make it easier to define a new handler.

```
\newcommand{\glstrunsrtdo}{\@glstr@noidx@do}
```

`\glstrgroupfield` `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*{\glstrgroupfield}{group}
```

The `tabular`-like glossary styles cause quite a problem with the iterative approach. In particular for the `group skip`. To compensate for this, the groups are now determined while `\@glstr@doglossary` is being constructed rather than in the handler.

`\@glstr@checkgroup` The argument is the entry's label. (This block of code was formerly in `\@glstr@noidx@do`.) Now that this is no longer within a tabular environment, the global definitions aren't needed. The result is now stored in `\@glstr@groupheading`, which will be empty if no heading is required. The current hierarchical level must have first been saved to `\glscurrententrylevel`.

```
\newcommand*{\@glstr@checkgroup}[1]{%
  \def\@glstr@groupheading{}%
  \key@ifundefined{glossentry}{group}%
  {%
    \letcs{\@glstr@sort}{glo@glstdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@glstr@sort{}\@nil
  }%
  {%
    \protected@edef\@glo@thislettergrp{%
      \csuse{glo@glstdetoklabel{#1}@glstrgroupfield}}%
  }%
```

Need to keep track of the current group for the current level.

```
\ifcsundef{@glstr@currentlettergroup\romannumeral\glscurrententrylevel}%
{\csdef{@glstr@currentlettergroup\romannumeral\glscurrententrylevel}{}}%
```

Has the group label changed for the current level?

```
\ifcsequal{@glo@thislettergrp}{@glstr@currentlettergroup\romannumeral\glscurrententrylevel}%
{}%
{%
```

```

\ifdefempty\@glo@thislettergrp
{}%
{%
  \ifnum\glscurrententrylevel>0\relax
    \protected@eappto\@glstr@groupheading{%
      \noexpand\glssubgroupheading
        {\@gls@currentlettergroup@level}{\glscurrententrylevel}%
        {\csuse{glo@\glsdetoklabel{#1}@parent}}}%
      {\expandonce\@glo@thislettergrp}%
    }%
  \else
    \ifdefempty{\@gls@currentlettergroup}{}%
    {%
      \def\@glstr@groupheading{\glsgroupskip}%
    }%
    \protected@eappto\@glstr@groupheading{%
      \noexpand\glsgroupheading{\expandonce\@glo@thislettergrp}%
    }%
  \fi
  \let\@gls@currentlettergroup@level\glscurrententrylevel
  \cslet{\@gls@currentlettergroup\romannumeral\glscurrententrylevel}\@glo@thislettergrp
}%
}%
}

```

`\glssubgroupheading{previous level}{level}{parent}{group label}`

`\glssubgroupheading`

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}}
```

`\GlsXtrLocationField` Stores the internal name of the location field.

```
\newcommand*{\GlsXtrLocationField}{location}
```

`\@glstr@noidx@do` 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{\@glstr@noidx@do}[1]{%
  \ifglstryexists{#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.

```

\ifglstrprintglossflatten
  \gls@level=\@glstr@leveloffset\relax
\else
  \gls@level=\numexpr\csuse{glo@\glsdetoklabel{#1}@level}+\@glstr@leveloffset\relax
\fi

```

```

\fi
\ifnum\gls@level>0
  \let\@glstr@ifischild\@firstoftwo
\else
  \let\@glstr@ifischild\@secondoftwo
\fi

```

Some glossary styles (such as topicmccls) save the level using \def so make sure \gls@level is expanded before being passed to \subglossentry.

```

\@glstr@ifischild
{%
  \ifdefvoid{\@gls@location}%
  {%

```

If \GlsXtrLocationField has been changed then don't fallback on loclist.

```

  \ifdefstring{\GlsXtrLocationField}{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}{}%
  }%
  {%
    \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.

`\glxtrnewgls`

```
\newcount\@glxtrnewgls@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.)

`\glxtrdoidentify`

```

\newcommand*{\glxtrdoidentify}[1]{%
\ifdefequal\@glxtr@record@setting\@glxtr@record@setting@off{}\@glxtr@record@setting@off{#1}%
}

```

`\@glxtr@providenewgls`

```

\newcommand*{\@glxtr@providenewgls}{%
\protected@write\@auxout{}\@string\providecommand{\@string\@glxtr@newglslike}[2]{}%
\let\@glxtr@providenewgls\relax
}

```

`\glxtridentifyglslike` Identify the command given in the second argument for the benefit of `bib2gls` and also identify command as a blocker for `\makefirstuc`.

```

\newcommand{\glxtridentifyglslike}[2]{%
\glsmfublocker{#2}%
\glxtrdoidentify
}%

```

```

\@glxtr@providenewgls
\protected@write\@auxout{}\string\@glxtr@newglslike{#1}\string#2}}%
}%
}

```

\@glxtr@providenewglsfamily

```

\newcommand*\@glxtr@providenewglsfamily{%
\protected@write\@auxout{}\string\providecommand{\string\@glxtr@newglslikefamily}[8]{}%
\let\@glxtr@providenewglsfamily\relax
}

```

```

\glxtridentifyglsfamily{<options>}{<prefix>}{<gls>}{<glspl>}{<Gls>}{<Glspl>}{<GLS>}{<GLSpl>}}

```

\glxtridentifyglsfamily

Identify the family of commands for the benefit of bib2gls and also establishes a sentence-case mapping.

```

\newcommand{\glxtridentifyglsfamily}[8]{%
\glsmfuaddmap{#3}{#5}%
\glsmfuaddmap{#4}{#6}%
\glsmfublocker{#7}%
\glsmfublocker{#8}%
\glxtrdoidentify
{%
\@glxtr@providenewglsfamily
\protected@write\@auxout{}\string\@glxtr@newglslikefamily{\detokenize{#1}}{\detokenize{#2}}{\detokenize{#3}}{\detokenize{#4}}{\detokenize{#5}}{\detokenize{#6}}{\detokenize{#7}}{\detokenize{#8}}}%
}%
}

```

\@glxtr@providenewglslink

```

\newcommand*\@glxtr@providenewglslink{%
\protected@write\@auxout{}\string\providecommand{\string\@glxtr@newglslink}[2]{}%
\let\@glxtr@providenewglslink\relax
}

```

\glxtridentifyglslink Identify the command given in the second argument for the benefit of bib2gls and identify the command as a blocker for \makefirstuc.

```

\newcommand{\glxtridentifyglslink}[2]{%
\glsmfublocker{#2}%
\glxtrdoidentify
{%
\@glxtr@providenewglslink
\protected@write\@auxout{}\string\@glxtr@newglslink{#1}\string#2}}%
}%
}

```



```
\@glstrnewglslink[\langle options \rangle]{\langle prefix \rangle}{\langle cs \rangle}{\langle inner cs name \rangle}
```

\@glstrnewglslink

```
\newcommand*{\@glstrnewglslink}[4]{%
  \ifdef{#3}%
  {%
    \PackageError{glossaries-extra}{Command \string#3\space already
defined}{}%
  }%
  {%
```

Write information to the aux file for bib2gls.

```
  \glstridentifyglslink{#2}{#3}%
  \ifcsdef{@#4link@#2}%
  {%
    \advance\@glstrnewgls@inner by \@ne
    \def\@glstrnewgls@innercsname{@#4link\number\@glstrnewgls@inner @#2}%
  }%
  {\def\@glstrnewgls@innercsname{@#4link@#2}}%
  \expandafter\newrobustcmd\expandafter*\expandafter
  #3\expandafter{\expandafter\@glshyp@opt\csname\@glstrnewgls@innercsname\endcsname}%
  \ifstrempy{#1}%
  {%
    \expandafter\newcommand\expandafter*\csname\@glstrnewgls@innercsname\endcsname[2][]{%
      \csname #4\endcsname{##1}{#2##2}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\csname\@glstrnewgls@innercsname\endcsname[2][]{%
      \csname #4\endcsname{#1,##1}{#2##2}%
    }%
  }%
  }%
}
```

```
\glstrnewglslink[\langle options \rangle]{\langle prefix \rangle}{\langle cs \rangle}
```

\glstrnewglslink

The first argument prepends to the options and the second argument is the prefix.

```
\newrobustcmd*{\glstrnewglslink}[3][]{%
  \@glstrnewglslink{#1}{#2}{#3}{@glshyp@link}%
}
```

```
\glstrnewglsdisp[\langle options \rangle]{\langle prefix \rangle}{\langle cs \rangle}
```

\glstrnewglsdisp

The first argument prepends to the options and the second argument is the prefix.

```
\newrobustcmd*{\glxstrnewglsdisp}[3][{}%
  \glxstrnewglslink{#1}{#2}{#3}{\glsdisp}%
}
```

```
\@glxstrnewgls[<options>]{<prefix>}{<cs>}{<inner cs
name>}
```

\glxstrnewgls

```
\newcommand*{\@glxstrnewgls}[4]{%
  \ifdef{#3}%
  {%
    \PackageError{glossaries-extra}{Command \string#3\space already
defined}{}%
  }%
  {%
```

Write information to the aux file for bib2gls.

```
  \glxstridentifyglslike{#2}{#3}%
  \ifcsdef{@#4like@#2}%
  {%
    \advance\@glxstrnewgls@inner by \@ne
    \def\@glxstrnewgls@innercsname{@#4like\number\@glxstrnewgls@inner @#2}%
  }%
  {\def\@glxstrnewgls@innercsname{@#4like@#2}}%
  \expandafter\newrobustcmd\expandafter*\expandafter
  #3\expandafter{\expandafter\@gls@hyp@opt\csname\@glxstrnewgls@innercsname\endcsname}%
  \ifstrempy{#1}%
  {%
    \expandafter\newcommand\expandafter*\csname\@glxstrnewgls@innercsname\endcsname[2][{}%
    \new@ifnextchar[%
      {\csname @#4@\endcsname{##1}{#2##2}}%
      {\csname @#4@\endcsname{##1}{#2##2}[]}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\csname\@glxstrnewgls@innercsname\endcsname[2][{}%
    \new@ifnextchar[%
      {\csname @#4@\endcsname{#1,##1}{#2##2}}%
      {\csname @#4@\endcsname{#1,##1}{#2##2}[]}%
    }%
  }%
}
```

```
\glxstrnewgls[<options>]{<prefix>}{<cs>}
```

\glxstrnewgls

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}%
}
```

sXtrSetRecordCountAttribute

```
\newcommand*{\GlsXtrSetRecordCountAttribute}[2]{%
  \@for\@glstr@cat:=#1\do
  {%
    \ifdefempty{\@glstr@cat}{}%
    {%
      \glssetcategoryattribute{\@glstr@cat}{recordcount}{#2}%
    }%
  }%
}
```

\glstriferecordtrigger

`\glstriferecordtrigger{<label>}{<trigger format>}{<normal>}`

```
\newcommand*{\glstriferecordtrigger}[3]{%
  \glshasattribute{#1}{recordcount}%
  {%
    \ifnum\glstrrecordtriggervalue{#1}>\glsggetattribute{#1}{recordcount}\relax
    #3%
    \else
    #2%
    \fi
  }%
  {#3}%
}
```

\@glstr@rglstrigger@record Still need a record to ensure that bib2gls selects the entry.

```
\newcommand*{\@glstr@rglstrigger@record}[3]{%
  \protected@edef\glslabel{\glstetoklabel{#2}}%
  \let\@gls@link@label\glslabel
  \def\@glstr@thevalue{%
    \def\@glstr@theHvalue{\@glstr@thevalue}%
    \def\@glsnumberformat{glstriggerrecordformat}%
    \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
    \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
    \def\@glstr@thevalue{%
      \def\@glstr@theHvalue{\@glstr@thevalue}%
    }%
  }
```

Save local setting.

```
\@gls@save@glslocal
```

Initialise preunset, prereset and postunset

```
\glsinitreunsets
\glstrinitwrgloss
\glslinkpresetkeys
\setkeys{glslink}{#1}%
\glslinkpostsetkeys
\ifdefempty{\@glstr@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}%
}

```

`\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}

```

```

\@rglsp1
\newcommand*{\@rglsp1}[2][{}]{%
  \new@ifnextchar[{\@rglsp1@{#1}{#2}}{\@rglsp1@{#1}{#2}[]}%
}

\@rglsp1@
\def\@rglsp1@#1#2[#3]{%
  \glstriferecordtrigger{#2}%
  {%
    \glstr@rglstrigger@record{#1}{#2}{\rglsp1format{#2}{#3}}%
  }%
  {%
    \@rglsp1@{#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]{%
  \glstriferecordtrigger{#2}%
  {%
    \glstr@rglstrigger@record{#1}{#2}{\rglsformat{#2}{#3}}%
  }%
  {%
    \@rgls@{#1}{#2}[#3]%
  }%
}%

\rglsp1
\newrobustcmd*{\rglsp1}{\@gls@hyp@opt\@rglsp1}
\glsmfuaddmap{\rglsp1}{\rglsp1}

\@rglsp1
\newcommand*{\@rglsp1}[2][{}]{%
  \new@ifnextchar[{\@rglsp1@{#1}{#2}}{\@rglsp1@{#1}{#2}[]}%
}

\@rglsp1@
\def\@rglsp1@#1#2[#3]{%
  \glstriferecordtrigger{#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}{\glsentrylongplural{#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}{\Glsentrylongplural{#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`.)

`\@glstr@do@inc@linkcount` This performs the actual incrementing and counter definition. The counter is given by `\c@glstr@linkcount@label` where *label* is the entry's label. Since this is performed within `\@glsl@link` the label can be accessed with `\glslabel`.

```

\newcommand{\@glstr@do@inc@linkcount}{%
  Does this entry have the linkcount attribute set?
  \glslifattribute{\glslabel}{linkcount}{true}%
  {%
    Does the counter exist?
    \ifcsdef{c@glstr@linkcount@\glslabel}{}%
    {%
      Counter doesn't exist, so define it.
      \newcounter{glstr@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{glstr@linkcount@\glslabel}%
            {\glsggetattribute{\glslabel}{linkcountmaster}}}%
          \@glo@tmp
        }%
        {}%
      }%
      Increment counter:
      \glstrinlinkcounter{glstr@linkcount@\glslabel}%
    }%
    {}%
  }
}

```

`\glstrinlinkcounter` May be redefined to use `\refstepcounter` if required.

```

\newcommand*{\glstrinlinkcounter}[1]{\stepcounter{#1}}

```

`\GlsXtrLinkCounterValue` Expands to the associated link counter register or 0 if not defined.

```

\newcommand*{\GlsXtrLinkCounterValue}[1]{%
  \ifcsundef{c@glstr@linkcount@#1}{0}{\csname c@glstr@linkcount@#1\endcsname}%
}

```

`\GlsXtrTheLinkCounter` Expands to the display value of the associated link counter or 0 if not defined.

```

\newcommand*{\GlsXtrTheLinkCounter}[1]{%
  \ifcsundef{theglstr@linkcount@#1}{0}%
  {\csname theglstr@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]{glsextr@linkcount@#1}

```

`\GlsXtrEnableLinkCounting[<master counter>]{<categories>}`

`\GlsXtrEnableLinkCounting` Enable link counting for the given categories.

```

\newcommand*{\GlsXtrEnableLinkCounting}[2][{}]{%
\let\glsextr@inc@linkcount\@glsextr@do@inc@linkcount
\@for\@glsextr@label:=#2\do
{%
\glsssetcategoryattribute{\@glsextr@label}{linkcount}{true}%
\ifstrempy{#1}{}%
}%
\ifcsundef{c@#1}%
{\@nocounterr{#1}}%
{\glsssetcategoryattribute{\@glsextr@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
{%
\glstryname{#1}%
}%
}

```

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

```
\glsaccessfmtname{<insert>}{<cs>}{<label>}
```

\glsaccessfmtname

```
\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{<insert>}{<cs>}{<label>}
```

\Glsaccessfmtname

```
\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}\langle cs\rangle\langle label\rangle`

`\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
  {%
    \glentrytext{#1}%
  }%
  {#1}%
}
```

`\glsaccessfmtttext{\insert}\langle cs\rangle\langle label\rangle`

`\glsaccessfmtttext`

```
\newcommand*\glsaccessfmtttext}[3]{%
  \glstextaccessdisplay
  {%
    \glsfmtfield{#1}{#2}{#3}{text}%
  }%
  {#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}%
}
```

`\Glsaccessfmtttext{\insert}\langle cs\rangle\langle label\rangle`

`\Glsaccessfmtttext`

```
\newcommand*\Glsaccessfmtttext}[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
  {%
    \glssupercase{\glentrytext{#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
  {%
    \glentryplural{#1}%
  }%
  {#1}%
}

```

`\glsaccessfmtplural{<insert>}{<cs>}{<label>}`

`\glsaccessfmtplural`

```

\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{<insert>}{<cs>}{<label>}`

`\Glsaccessfmtplural`

```
\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>}`

`\GLSaccessfmtplural`

```
\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
  {%
```

```

\glentryfirst{#1}%
}%
{#1}%
}

```

```
\glsaccessfmtfirst{<insert>}{<cs>}{<label>}
```

\glsaccessfmtfirst

```

\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
{%
\glentryfirst{#1}%
}%
{#1}%
}

```

```
\Glsaccessfmtfirst{<insert>}{<cs>}{<label>}
```

\Glsaccessfmtfirst

```

\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
{%
\glsupercase{\glentryfirst{#1}}%
}%
{#1}%
}

```


\GLSaccessfmtfirst

\GLSaccessfmtfirst{<insert>}{<cs>}{<label>}

```
\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
  {%
    \glentryfirstplural{#1}%
  }%
  {#1}%
}
```

\glsaccessfmtfirstplural{<insert>}{<cs>}{<label>}

\glsaccessfmtfirstplural

```
\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{<insert>}{<cs>}{<label>}

\Glsaccessfmtfirstplural

```
\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
  {%
    \glssupercase{\glsentryfirstplural{#1}}%
  }%
  {#1}%
}

```

`\GLSaccessfmtfirstplural{<insert>}{<cs>}{<label>}`

`\GLSaccessfmtfirstplural`

```

\newcommand*{\GLSaccessfmtfirstplural}[3]{%
  \glsfirstpluralaccessdisplay
  {%
    \Glsfmtfield{#1}{#2}{#3}{firstpl}%
  }%
  {#3}%
}

```

`\glsaccesssymbol` Display the symbol value (no link and no check for existence).

```

\newcommand*{\glsaccesssymbol}[1]{%
  \glssymbolaccessdisplay
  {%
    \glsentrysymbol{#1}%
  }%
  {#1}%
}

```

`\glsaccessfmtymsymbol{<insert>}{<cs>}{<label>}`

`\glsaccessfmtymsymbol`

```

\newcommand*{\glsaccessfmtymsymbol}[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{<insert>}{<cs>}{<label>}`

`\Glsaccessfmtsymbol`

```
\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
  {%
    \glssupercase{\Glsentrysymbol{#1}}%
  }%
  {#1}%
}
```

`\GLSaccessfmtsymbol{<insert>}{<cs>}{<label>}`

`\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
  {%
```

```

\glentrysymbolplural{#1}%
}%
{#1}%
}

```

```

\glsaccessfmtsymbolplural{<insert>}{<cs>}{<label>}

```

\glsaccessfmtsymbolplural

```

\newcommand*{\glsaccessfmtsymbolplural}[3]{%
\glssymbolpluralaccessdisplay
{%
\glfmtfield{#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
{%
\glentrysymbolplural{#1}%
}%
{#1}%
}

```

```

\Glsaccessfmtsymbolplural{<insert>}{<cs>}{<label>}

```

\Glsaccessfmtsymbolplural

```

\newcommand*{\Glsaccessfmtsymbolplural}[3]{%
\glssymbolpluralaccessdisplay
{%
\glfmtfield{#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
{%
\glsupercase{\glentrysymbolplural{#1}}%
}%
{#1}%
}

```

`\GLSaccessfmtsymbolplural`

```
\GLSaccessfmtsymbolplural{<insert>}{<cs>}{<label>}
```

```
\newcommand*{\GLSaccessfmtsymbolplural}[3]{%
  \glsymbolpluralaccessdisplay
  {%
    \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`

```
\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) 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{<insert>}{<cs>}{<label>}`

`\glsaccessfmtdescplural`

```

\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{<insert>}{<cs>}{<label>}`

`\Glsaccessfmtdescplural`

```
\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{<insert>}{<cs>}{<label>}`

`\GLSaccessfmtdescplural`

```
\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
  {%
```

```

\glentryshort{#1}%
}%
{#1}%
}

```

```
\glsaccessfmtshort{<insert>}{<cs>}{<label>}
```

\glsaccessfmtshort

```

\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
{%
\glentryshort{#1}%
}%
{#1}%
}

```

```
\Glsaccessfmtshort{<insert>}{<cs>}{<label>}
```

\Glsaccessfmtshort

```

\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
{%
\glssupercase{\glentryshort{#1}}%
}%
{#1}%
}

```


`\GLSaccessfmtshort`

`\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
  {%
    \glstryshortpl{#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
  {%
    \Glsentryshortpl{#1}%
  }%
  {#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
  {%
    \glssuppercase{\glentryshortpl{#1}}%
  }%
  {#1}%
}

```

`\GLSaccessfmtshortpl{<insert>}{<cs>}{<label>}`

`\GLSaccessfmtshortpl`

```

\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{\glentrylong{#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

```
\Glsaccessfmtlong{<insert>}{<cs>}{<label>}
```

```
\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

```
\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

```
\glsaccessfmtlongpl{<insert>}{<cs>}{<label>}
```

```
\newcommand*\glsaccessfmtlongpl}[3]{%
  \glsaccesslongpl{#1}%
  {#2}%
  {#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>}

```

`\Glsaccessfmtlongpl`

```

\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{<insert>}{<cs>}{<label>}

```

`\GLSaccessfmtlongpl`

```

\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{<insert>}{<cs>}{<label>}`

`\glsaccessfmtuseri`

```
\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{<insert>}{<cs>}{<label>}
```

\Glsaccessfmtuseri

```

\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) converted to upper case.

```

\ifdef\glsuseriaccessdisplay
{
  \newcommand*{\GLSaccessuseri}[1]{%
    \glsuseriaccessdisplay
    {%
      \glsuppercase{\glsentryuseri{#1}}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\GLSaccessuseri}[1]{%
    \glsuppercase{\glsentryuseri{#1}}%
  }
}

```

`\GLSaccessfmtuseri{<insert>}{<cs>}{<label>}`

`\GLSaccessfmtuseri`

```
\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{<insert>}{<cs>}{<label>}`

`\glsaccessfmtuserii`

```
\ifdef\glsuseriiaccessdisplay
{
  \newcommand*{\glsaccessfmtuserii}[3]{%
    \glsuseriiaccessdisplay
    {%
      \glsfmtfield{#1}{#2}{#3}{userii}%
    }%
    {#3}%
  }
}
```

```

    }%
    {#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{<insert>}{<cs>}{<label>}`

`\Glsaccessfmtuserii`

```

\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}%
  }
}

```

```
\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) converted to upper case.

```

\ifdef\glsuseriiiaccessdisplay
{
  \newcommand*\GLSaccessuseriii}[1]{%
    \glsuseriiiaccessdisplay
    {%
      \glsuppercase{\glstentryuseriii{#1}}%
    }%
    {#1}%
  }
}
{
  \newcommand*\GLSaccessuseriii}[1]{%
    \glsuppercase{\glstentryuseriii{#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}%
  }
}

```

USER4

`\glsaccessuseriv` Display the `user4` value (no link and no check for existence).

```

\ifdef\glsuserivaccessdisplay
{
  \newcommand*{\glsaccessuseriv}[1]{%
    \glsuserivaccessdisplay
    {%
      \glstryuseriv{#1}%
    }%
    {#1}%
  }
}
{
  \newcommand*{\glsaccessuseriv}[1]{%
    \glstryuseriv{#1}%
  }
}

```

```
\glsaccessfmtuseriv{<insert>}{<cs>}{<label>}
```

`\glsaccessfmtuseriv`

```

\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{<insert>}{<cs>}{<label>}`

`\Glsaccessfmtuseriv`

```

\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{<insert>}{<cs>}{<label>}
```

\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{<insert>}{<cs>}{<label>}
```

\glsaccessfmtuserv

```

\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{<insert>}{<cs>}{<label>}

\Glsaccessfmtuserv

```

\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{<insert>}{<cs>}{<label>}

\GLSaccessfmtuserv

```
\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{<insert>}{<cs>}{<label>}
```

\glsaccessfmtuservi

```

\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{<insert>}{<cs>}{<label>}`

`\Glsaccessfmtuservi`

```
\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}}%
  }
}
```

`\GLSaccessfmtuservi{<insert>}{<cs>}{<label>}`

`\GLSaccessfmtuservi`

```
\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{glstrabbrv}{access}{%
    \def\@gls@nameaccess{#1}%
}
\define@key{glstrabbrv}{textaccess}{%
    \def\@gls@textaccess{#1}%
}
\define@key{glstrabbrv}{pluralaccess}{%
    \def\@gls@pluralaccess{#1}%
}
\define@key{glstrabbrv}{firstaccess}{%
    \def\@gls@firstaccess{#1}%
}
\define@key{glstrabbrv}{firstpluralaccess}{%
    \def\@gls@firstpluralaccess{#1}%
}
\define@key{glstrabbrv}{shortaccess}{%
    \def\@gls@shortaccess{#1}%
}
\define@key{glstrabbrv}{shortpluralaccess}{%
    \def\@gls@shortaccesspl{#1}%
}
\define@key{glstrabbrv}{longaccess}{%
    \def\@gls@longaccess{#1}%
}
\define@key{glstrabbrv}{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{<attribute>}{<true>}{<false>}
```

```
\@gls@ifaccessattribute@set
```

```

\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{<long>}{<short>}
```

```
\glsdefaultshortaccess
```

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
\glstrassignactualsetup
\protected@edef\@gls@tmp{\endgroup
\def\noexpand\@gls@actualshort{\glstrorgshort}%
\def\noexpand\@gls@actuallong{\glstrorglong}%
\def\noexpand\@gls@actualshortpl{\@gls@shortpl}%
\def\noexpand\@gls@actuallongpl{\@gls@longpl}%
}%
\@gls@tmp
}

```

\@setup@default@short@access Renamed \@gls@setup@default@access and removed argument since it can be obtained from \glstrorgshort.

\@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\@glstr@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}}%
}%
\ea\ppto\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'\glsextrabrvpluralsuffix}%
    }%
    {%
      \@gls@ifaccessattribute@set{noshortplural}%
      {%
        \let\@gls@shortaccesspl\@gls@shortaccess
      }%
      {%
        \let\@gls@shortaccesspl\@gls@actualshortpl
      }%
    }%
  }%
\ifdefempty\@gls@longaccesspl
{%
  \protected@edef\@gls@shortaccesspl{\glsdefaultshortaccess
    {\expandonce\@gls@actualallongpl}{\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}%
  {%
    \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.

`\glxtrprovideaccsuppcmd`

```
\newcommand*{\glxtrprovideaccsuppcmd}[2]{%
  \ifcsundef{glxtr#1#2accsup}%
  {\csdef{glxtr#1#2accsup}{\glsshortaccsup}}%
  {}%
}
```

`trAccSuppAbbrSetNoLongAttrs` For styles where the name, first and text are just the abbreviation.

```
\newcommand*{\glxtrAccSuppAbbrSetNoLongAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glssetcategoryattribute{#1}{firstshortaccess}{true}%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#1}{name}%
  \glxtrprovideaccsuppcmd{#1}{first}%
  \glxtrprovideaccsuppcmd{#1}{firstpl}%
  \glxtrprovideaccsuppcmd{#1}{text}%
  \glxtrprovideaccsuppcmd{#1}{plural}%
}
```

`ccSuppAbbrSetFirstLongAttrs` 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*{\glxtrAccSuppAbbrSetFirstLongAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#1}{name}%
  \glxtrprovideaccsuppcmd{#1}{text}%
  \glxtrprovideaccsuppcmd{#1}{plural}%
}
```

`ccSuppAbbrSetTextShortAttrs` 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*{\glxtrAccSuppAbbrSetTextShortAttrs}[1]{%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#1}{text}%
  \glxtrprovideaccsuppcmd{#1}{plural}%
}
```

`ccSuppAbbrSetNameShortAttrs` 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*{\glxtrAccSuppAbbrSetNameShortAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#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*{\glxtrAccSuppAbbrSetNameLongAttrs}[1]{%
```



```

\glsetcategoryattribute{#1}{firstshortaccess}{true}%
\glsetcategoryattribute{#1}{textshortaccess}{true}%
\glxtrprovideaccsuppcmd{#1}{first}%
\glxtrprovideaccsuppcmd{#1}{firstpl}%
\glxtrprovideaccsuppcmd{#1}{text}%
\glxtrprovideaccsuppcmd{#1}{plural}%
}

```

End of if accsupp part

```

}
{

```

No accessibility support. Just define these commands to do `\glentry<xxx>`

`\glsaccessname` Display the name value (no link and no check for existence).

```

\newcommand*{\glsaccessname}[1]{\glentryname{#1}}

```

`\glsaccessfmtname`

```

\glsaccessfmtname{<insert>}{<cs>}{<label>}

```

```

\newcommand*{\glsaccessfmtname}[3]{%
\glsfmtfield{#1}{#2}{#3}{name}%
}

```

`\Glsaccessname` Display the name value (no link and no check for existence) with the first letter converted to upper case.

```

\newcommand*{\Glsaccessname}[1]{\glentryname{#1}}

```

`\Glsaccessfmtname`

```

\Glsaccessfmtname{<insert>}{<cs>}{<label>}

```

```

\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{\glentryname{#1}}}

```

`\GLSaccessfmtname`

```

\GLSaccessfmtname{<insert>}{<cs>}{<label>}

```

```

\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{<insert>}{<cs>}{<label>}
```

```
\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{<insert>}{<cs>}{<label>}
```

```
\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`

```
\GLSaccessfmttext{<insert>}{<cs>}{<label>}
```

```
\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`

```
\glsaccessfmtplural{<insert>}{<cs>}{<label>}
```

```
\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`

```
\GLSaccessfmtfirst{<insert>}{<cs>}{<label>}
```

```
\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`

```
\glsaccessfmtfirstplural{<insert>}{<cs>}{<label>}
```

```
\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{<insert>}{<cs>}{<label>}
```

```
\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{<insert>}{<cs>}{<label>}
```

```
\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}}
```

`\glsaccessfmtsymb`

```
\glsaccessfmtsymb{\insert}{\cs}{\label}
```

```
\newcommand*{\glsaccessfmtsymb}[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}}
```

`\Glsaccessfmtsymb`

```
\Glsaccessfmtsymb{\insert}{\cs}{\label}
```

```
\newcommand*{\Glsaccessfmtsymb}[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}}}
```

`\GLSaccessfmtsymb`

```
\GLSaccessfmtsymb{\insert}{\cs}{\label}
```

```
\newcommand*{\GLSaccessfmtsymb}[3]{%  
  \GLSfmtfield{#1}{#2}{#3}{symbol}%  
}
```

`\glsaccesssymbolplural` Display the symbolplural value (no link and no check for existence).

```
\newcommand*{\glsaccesssymbolplural}[1]{\glsentrysymbolplural{#1}}
```

`\glsaccessfmtsymbplural`

```
\glsaccessfmtsymbplural{\insert}{\cs}{\label}
```

```
\newcommand*{\glsaccessfmtsymbplural}[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{<insert>}{<cs>}{<label>}
```

`\Glsaccessfmtsymbolplural`

```
\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{<insert>}{<cs>}{<label>}
```

`\GLSaccessfmtsymbolplural`

```
\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{<insert>}{<cs>}{<label>}
```

`\glsaccessfmtdesc`

```
\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{<insert>}{<cs>}{<label>}
```

`\Glsaccessfmtdesc`

```
\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{<insert>}{<cs>}{<label>}`

`\GLSaccessfmtdesc`

```
\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{<insert>}{<cs>}{<label>}`

`\glsaccessfmtdescplural`

```
\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{<insert>}{<cs>}{<label>}`

`\Glsaccessfmtdescplural`

```
\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{<insert>}{<cs>}{<label>}`

`\GLSaccessfmtdescplural`

```
\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`

```
\glsaccessfmtshort{<insert>}{<cs>}{<label>}
```

```
\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`

```
\Glsaccessfmtshort{<insert>}{<cs>}{<label>}
```

```
\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`

```
\GLSaccessfmtshort{<insert>}{<cs>}{<label>}
```

```
\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`

```
\glsaccessfmtshortpl{<insert>}{<cs>}{<label>}
```

```
\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{<insert>}{<cs>}{<label>}
```

`\Glsaccessfmtshortpl`

```
\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{<insert>}{<cs>}{<label>}
```

`\GLSaccessfmtshortpl`

```
\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{<insert>}{<cs>}{<label>}
```

`\glsaccessfmtlong`

```
\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{<insert>}{<cs>}{<label>}
```

`\Glsaccessfmtlong`

```
\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{<insert>}{<cs>}{<label>}
```

`\glsaccessfmtuseri`

```
\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{<insert>}{<cs>}{<label>}
```

`\Glsaccessfmtuseri`

```
\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{<insert>}{<cs>}{<label>}
```

`\GLSaccessfmtuseri`

```
\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{<insert>}{<cs>}{<label>}
```

`\glsaccessfmtuserii`

```
\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{<insert>}{<cs>}{<label>}
```

\Glsaccessfmtuserii

```
\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{<insert>}{<cs>}{<label>}
```

\GLSaccessfmtuserii

```
\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{<insert>}{<cs>}{<label>}
```

\glsaccessfmtuseriii

```
\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

```
\Glsaccessfmtuseriii{\insert}{\cs}{\label}
```

```
\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

```
\GLSaccessfmtuseriii{\insert}{\cs}{\label}
```

```
\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}}
```

\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` `{\insert}{\langle cs \rangle}{\langle label \rangle}`

`\GLSaccessfmtuseriv`

```
\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` `{\insert}{\langle cs \rangle}{\langle label \rangle}`

`\glsaccessfmtuserv`

```
\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` `{\insert}{\langle cs \rangle}{\langle label \rangle}`

`\Glsaccessfmtuserv`

```
\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` `{\insert}{\langle cs \rangle}{\langle label \rangle}`

`\GLSaccessfmtuserv`

```
\newcommand*{\GLSaccessfmtuservi}[3]{%
  \GLSfmtfield{#1}{#2}{#3}{uservi}%
}
```

USER6

\glsaccessuservi Display the user6 value (no link and no check for existence).

```
\newcommand*{\glsaccessuservi}[1]{\glsentryuservi{#1}}
```

```
\glsaccessfmtuservi{\insert}{\cs}{\label}}
```

\glsaccessfmtuservi

```
\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{\insert}{\cs}{\label}}
```

\Glsaccessfmtuservi

```
\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{\insert}{\cs}{\label}}
```

\GLSaccessfmtuservi

```
\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}{}

\extrAccSuppAbbrSetNoLongAttrs 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{\glsaccessstext}{\Glsaccessstext}
\glsmfuaddmap{\glsaccessfmtstext}{\Glsaccessfmtstext}
\glsmfublocker{\GLSaccessstext}
\glsmfublocker{\GLSaccessfmtstext}
\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{\GLSaccessfmtsymbollplural}
\glsmfuaddmap{\glaccessdesc}{\Glsaccessdesc}
\glsmfuaddmap{\glaccessfmdesc}{\Glsaccessfmdesc}
\glsmfublocker{\GLSaccessdesc}
\glsmfublocker{\GLSaccessfmdesc}
\glsmfuaddmap{\glaccessdescplural}{\Glsaccessdescplural}
\glsmfuaddmap{\glaccessfmdescplural}{\Glsaccessfmdescplural}
\glsmfublocker{\GLSaccessdescplural}
\glsmfublocker{\GLSaccessfmdescplural}
\glsmfuaddmap{\glaccessshort}{\Glsaccessshort}
\glsmfuaddmap{\glaccessfmdshort}{\Glsaccessfmdshort}
\glsmfublocker{\GLSaccessshort}
\glsmfublocker{\GLSaccessfmdshort}
\glsmfuaddmap{\glaccessshortpl}{\Glsaccessshortpl}
\glsmfuaddmap{\glaccessfmdshortpl}{\Glsaccessfmdshortpl}
\glsmfublocker{\GLSaccessshortpl}
\glsmfublocker{\GLSaccessfmdshortpl}
\glsmfuaddmap{\glaccesslong}{\Glsaccesslong}
\glsmfuaddmap{\glaccessfmdlong}{\Glsaccessfmdlong}
\glsmfublocker{\GLSaccesslong}
\glsmfublocker{\GLSaccessfmdlong}
\glsmfuaddmap{\glaccesslongpl}{\Glsaccesslongpl}
\glsmfuaddmap{\glaccessfmdlongpl}{\Glsaccessfmdlongpl}
\glsmfublocker{\GLSaccesslongpl}
\glsmfublocker{\GLSaccessfmdlongpl}
\glsmfuaddmap{\glaccessuseri}{\Glsaccessuseri}
\glsmfuaddmap{\glaccessfmduseri}{\Glsaccessfmduseri}
\glsmfublocker{\GLSaccessuseri}
\glsmfublocker{\GLSaccessfmduseri}
\glsmfuaddmap{\glaccessuserii}{\Glsaccessuserii}
\glsmfuaddmap{\glaccessfmduserii}{\Glsaccessfmduserii}
\glsmfublocker{\GLSaccessuserii}
\glsmfublocker{\GLSaccessfmduserii}
\glsmfuaddmap{\glaccessuseriii}{\Glsaccessuseriii}
\glsmfuaddmap{\glaccessfmduseriii}{\Glsaccessfmduseriii}
\glsmfublocker{\GLSaccessuseriii}
\glsmfublocker{\GLSaccessfmduseriii}
\glsmfuaddmap{\glaccessuseriv}{\Glsaccessuseriv}
\glsmfuaddmap{\glaccessfmduseriv}{\Glsaccessfmduseriv}
\glsmfublocker{\GLSaccessuseriv}
\glsmfublocker{\GLSaccessfmduseriv}
\glsmfuaddmap{\glaccessuserv}{\Glsaccessuserv}
\glsmfuaddmap{\glaccessfmduserv}{\Glsaccessfmduserv}
\glsmfublocker{\GLSaccessuserv}
\glsmfublocker{\GLSaccessfmduserv}
\glsmfuaddmap{\glaccessuservi}{\Glsaccessuservi}
\glsmfuaddmap{\glaccessfmduservi}{\Glsaccessfmduservi}
\glsmfublocker{\GLSaccessuservi}
\glsmfublocker{\GLSaccessfmduservi}

```

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}%
  }%
}%
}
```

`\glsetcategoryattributes`

$\glsetcategoryattributes\langle category \rangle\{\langle attribute\ list \rangle\}$
 $\{\langle value \rangle\}$

Similar to above, but globally apply to each attribute in the list to the given category.

```
\newcommand*\glsetcategoryattributes[3]{%
  \@for\@gls@thisattrlabel:=#2\do{%
    \csgdef{\glstr@categoryattr@#1\@gls@thisattrlabel}{#3}%
  }%
}
```

`\glgetcategoryattribute`

$\glgetcategoryattribute\langle category \rangle\{\langle attribute-label \rangle\}$

Get the value of the given attribute for the given category. Does nothing if the attribute isn't defined.

```
\newcommand*\glgetcategoryattribute[2]{%
  \csuse{\glstr@categoryattr@#1@#2}%
}
```

`\glunsetcategoryattribute`

$\glunsetcategoryattribute\langle category \rangle\{\langle attribute-label \rangle\}$

Unsets the given attribute for the given category.

```
\newcommand*\glunsetcategoryattribute[2]{%
  \csundef{\glstr@categoryattr@#1@#2}%
}
```

`\glshascategoryattribute`

$\glshascategoryattribute\langle category \rangle\{\langle attribute-label \rangle\}$
 $\{\langle true \rangle\}\{\langle false \rangle\}$

Tests if the category has the given attribute set.

```
\newcommand*\glshascategoryattribute[4]{%
  \ifcsvoid{\glstr@categoryattr@#1@#2}{#4}{#3}%
}
```

`\glsetattribute`

$\glsetattribute\langle entry\ label \rangle\{\langle attribute-label \rangle\}\{\langle value \rangle\}$

Short cut where the category label is obtained from the entry information.

```
\newcommand*\glsetattribute[3]{%
  \glsetcategoryattribute{\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]{%
  \ifglentryexists{#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.

```
\newcommand{\glsifcategoryattribute}[5]{%
  \ifcsundef{@glxtr@categoryattr@#1@#2}%
  {#5}%
  {\ifcsstring{@glxtr@categoryattr@#1@#2}{#3}{#4}{#5}}%
}
```

`\glsifattribute`

`\glsifattribute{<entry label>}{<attribute-label>}{<value>}{<true part>}{<false part>}`

Short cut to determine if the given entry has a category with the given attribute set.

```
\newcommand{\glsifattribute}[5]{%
  \ifglentryexists{#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{<category-label>}{<attribute>}
{<true>}{<false>}
```

`\glsifcategoryattributetrue`

Does *<false>* if the entry hasn't been defined.

```
\newcommand*{\glsifcategoryattributetrue}[4]{%
\ifcsequal{\@glsxtr@categoryattr@#1@#2}%
{\@glsxtr@truevalue}%
{#3}{#4}%
}
```

```
\glsifattributetrue{<label>}{<attribute>}{<true>}{<false>}
```

`\glsifattributetrue`

Does *<false>* if the entry hasn't been defined.

```
\newcommand*{\glsifattributetrue}[4]{%
\ifcsundef{glo@glsetoklabel{#1}@category}%
{#4}
{\ifcsequal
{\@glsxtr@categoryattr@%csname glo@glsetoklabel{#1}@category\endcsname @#2}%
{\@glsxtr@truevalue}%
{#3}{#4}%
}%
}
```

```
\glsifcategoryattributehasitem{<category>}
{<attribute-label>}{<item>}{<true
part>}{<false part>}
```

`\glsifcategoryattributehasitem`

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.

```
\glsssetcategoryattribute{acronym}{regular}{true}
```

`\glsssetregularcategory` Convenient shortcut to add the regular attribute.

```
\newcommand*\glsssetregularcategory[1]{%
  \glsssetcategoryattribute{#1}{regular}{true}%
}
```

`\glsifregularcategory`

```
\glsifregularcategory{<category>}{<true part>}{<false part>}
```

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>}
```

`\glsifregular`

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>}
```

`\glsifnotregular`

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>}
```

`\glsforeachincategory`

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}%
{%
\forallglsentries[#3]{#4}%
{%
\glsifcategory{#4}{#2}{#5}{}%
}%
}%
}
```

```
\glsforeachwithattribute[<glossary
labels>]{<attribute-label>}{<attribute-value>}
{<glossary-cs>}{<label-cs>}{<body>}
```

`\glsforeachwithattribute`

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}%
{%
\forallglsentries[#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 `\glstrpostdescription`.

```
\ifdef\newterm
{%
```

`\newterm`

```
\renewcommand*{\newterm}[2][{}]{%
\newglossaryentry{#2}%
{type={index},category=index,name={#2},%
```

```

        description={\glxtrpostdescription\nopostdesc},#1}%
    }

```

Indexed terms are regular by default.

```

\glsssetcategoryattribute{index}{regular}{true}

```

`\glxtrpostdescindex`

```

\newcommand*{\glxtrpostdescindex}{}

}

{}

```

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
{%

```

`\glxtrnewsymbol` Unlike `\newterm`, this has a separate argument for the label (since the symbol will likely contain commands).

```

\newcommand*{\glxtrnewsymbol}[3] []{%
    \newglossaryentry{#2}{name={#3},sort={#2},type=symbols,category=symbol,#1}%
}

```

Symbols are regular by default.

```

\glsssetcategoryattribute{symbol}{regular}{true}

```

`\glxtrpostdescsymbol`

```

\newcommand*{\glxtrpostdescsymbol}{}

}

{}

```

Similar for the `numbers` option.

```

\ifdef\printnumbers
{%

```

`\glxtrnewnumber`

```

\ifdef\printnumbers
\newcommand*{\glxtrnewnumber}[3] []{%
    \newglossaryentry{#2}{name={#3},sort={#2},type=numbers,category=number,#1}%
}

```

Numbers are regular by default.

```

\glsssetcategoryattribute{number}{regular}{true}

```

`\glxtrpostdescnumber`

```

\newcommand*{\glxtrpostdescnumber}{}

```



```
}
{}
```

`\glxtrsetcategory` 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*\glxtrsetcategory[2]{%
  \@for\@glxtr@label:=#1\do
  {%
    \glsfieldxddef{\@glxtr@label}{category}{#2}%
  }%
}
```

`\glxtrsetcategoryforall` 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*\glxtrsetcategoryforall[2]{%
  \forallglossaries[#1]{\@glxtr@type}{%
    \forglsentries[\@glxtr@type]{\@glxtr@label}%
    {%
      \glsfieldxddef{\@glxtr@label}{category}{#2}%
    }%
  }%
}
```

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

`\glxtrfieldtitlecase`

Apply title casing to the contents of the given field.

```
\newcommand*\glxtrfieldtitlecase[2]{%
  \expandafter\glxtrfieldtitlecasecs\expandafter
  {\csname glo@\glsdetoklabel{#1}0#2\endcsname}%
}
```

`\glxtrfieldtitlecasecs` The command used by `\glxtrfieldtitlecase`. May be redefined to use a different command, for example, `\xcapitalisefmtwords`. Check for `\glscapitalisewords`, which was added to glossaries v4.48.

```
\ifdef\glscapitalisewords
{
  \newcommand*\glxtrfieldtitlecasecs[1]{%
    \expandafter\glscapitalisewords\expandafter{#1}}
}
{
  \newcommand*\glxtrfieldtitlecasecs[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\@glstr@attrval{\glsgetattribute{#1}{glossdescfont}}%
      \ifcsdef{\@glstr@attrval}%
      {%
        \letcs{\@glstr@glossdescfont}{\@glstr@attrval}%
      }%
      {%
        \GlossariesExtraWarning{Unknown control sequence name
          '\@glstr@attrval' supplied in glossdescfont attribute
          for entry '#1'. Ignoring}%
        \let\@glstr@glossdescfont\@firstofone
      }%
    }%
    {\let\@glstr@glossdescfont\@firstofone}%
    \glsifattribute{#1}{glossdesc}{firstuc}%
    {%
      \@glstr@glossdescfont{\Glsaccessdesc{#1}}%
    }%
    {%
      \glsifattribute{#1}{glossdesc}{title}%
      {%
        \@glstr@do@titlecaps@warn
        \glsdescriptionaccessdisplay
        {%
          \@glstr@glossdescfont{\glstrfieldtitlecase{#1}{desc}}%
        }%
        {#1}%
      }%
      {%
        \@glstr@glossdescfont{\Glsaccessdesc{#1}}%
      }%
    }%
  }%
}
}
{
  \renewcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glssetabbrvfmt{\glscategory{#1}}}%

```

```

\glshasattribute{#1}{glossdescfont}%
{%
    \protected@edef\@glxtr@attrval{\glsggetattribute{#1}{glossdescfont}}%
    \ifcsdef{\@glxtr@attrval}%
    {%
        \letcs{\@glxtr@glossdescfont}{\@glxtr@attrval}%
    }%
    {%
        \GlossariesExtraWarning{Unknown control sequence name
            '\@glxtr@attrval' supplied in glossdescfont attribute
            for entry '#1'. Ignoring}%
        \let\@glxtr@glossdescfont\@firstofone
    }%
}%
{\let\@glxtr@glossdescfont\@firstofone}%
\gl@ifattribute{#1}{glossdesc}{firstuc}%
{%
    \@glxtr@glossdescfont{\Glsentrydesc{#1}}%
}%
{%
    \gl@ifattribute{#1}{glossdesc}{title}%
    {%
        \@glxtr@do@titlecaps@warn
        \@glxtr@glossdescfont{\glxtr@field@titlecase{#1}{desc}}%
    }%
    {%
        \@glxtr@glossdescfont{\glentrydesc{#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]{%
        \gl@ifexistsorwarn{#1}%
    }%
    \glsetabbrvfmt{\glscategory{#1}}%
}

```

As from version 1.04, allow the `glossnamefont` attribute to determine the font applied.

```

\glshasattribute{#1}{glossnamefont}%
{%
    \protected@edef\@glxtr@attrval{\glsggetattribute{#1}{glossnamefont}}%
    \ifcsdef{\@glxtr@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
{%

```

Hide the label from the upper-casing command.

```

\letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
\@glsxtr@glossnamefont{\glsupercase{\glo@name}}%
}%
{#1}%
}%
{%
\letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
\glsnameaccessdisplay
{%
\expandafter\@glsxtr@glossnamefont\expandafter{\glo@name}%
}%
{#1}%
}%
}%
}%

```

Do post-name hook:

```

\glxtrpostnamehook{#1}%
}%
}
}
{
\renewcommand*{\glossentryname}[1]{%
\@glsdoifexistsorwarn{#1}%
{%
\glsetabbrvfmt{\glscategory{#1}}%
\glshasattribute{#1}{glossnamefont}%
{%

\protected@edef\@glxtr@attrval{\glsetattribute{#1}{glossnamefont}}%
\ifcsdef{\@glxtr@attrval}%
{%
\letcs{\@glxtr@glossnamefont}{\@glxtr@attrval}%
}%
}%
\GlossariesExtraWarning{Unknown control sequence name
'\@glxtr@attrval' supplied in glossnamefont attribute
for entry '#1'. Reverting to default \string\glnamefont}%
\let\@glxtr@glossnamefont\glnamefont
}%
}%
{\let\@glxtr@glossnamefont\glnamefont}%
\gl@ifattribute{#1}{glossname}{firstuc}%
{%
\@glxtr@glossnamefont{\Glsentryname{#1}}%
}%
}%
\gl@ifattribute{#1}{glossname}{title}%
{%
\@glxtr@do@titlecaps@warn
\@glxtr@glossnamefont{\glxtrfieldtitlecase{#1}{name}}%
}%
}%
\gl@ifattribute{#1}{glossname}{uc}%
{%

```

Hide the label from the upper-casing command.

```

\letcs{\glo@name}{glo@\glsetoklabel{#1}@name}%
\@glxtr@glossnamefont{\gluppercase{\glo@name}}%
}%
{%

```

This little trick is used by glossaries to allow the user to redefine `\glnamefont` to use `\makefirstuc`. Support it even though they can now use the `firstuc` attribute.

```

\letcs{\glo@name}{glo@\glsetoklabel{#1}@name}%

```

```

        \expandafter\@glxtr@glossnamefont\expandafter{\glo@name}%
      }%
    }%
  }%
Do post-name hook.
  \glxtrpostnamehook{#1}%
}%
}
}

```

`\Glossentryname` Redefine to set the abbreviation format and accessibility support.

```

\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\Glossentryname}[1]{%
    \@glstdoifexistsorwarn{#1}%
    {%
      \glsetabbrvfmt{\glscategory{#1}}%

```

As from version 1.04, allow the `glossnamefont` attribute to determine the font applied.

```

    \glshasattribute{#1}{glossnamefont}%
    {%
      \protected@edef\@glxtr@attrval{\glsetattribute{#1}{glossnamefont}}%
      \ifcsdef{\@glxtr@attrval}%
      {%
        \letcs{\@glxtr@glossnamefont}{\@glxtr@attrval}%
      }%
      {%
        \GlossariesExtraWarning{Unknown control sequence name
          '\@glxtr@attrval' supplied in glossnamefont attribute
          for entry '#1'. Reverting to default \string\glnamefont}%
        \let\@glxtr@glossnamefont\glnamefont
      }%
    }%
    {\let\@glxtr@glossnamefont\glnamefont}%
    \glnameaccessdisplay
    {%
      \@glxtr@glossnamefont{\Glsentryname{#1}}%
    }%
    {#1}%

```

Do post-name hook:

```

    \glxtrpostnamehook{#1}%
  }%
}
}
{
  \renewcommand*{\Glossentryname}[1]{%
    \@glstdoifexistsorwarn{#1}%

```

```

{%
\glsetabbrvfmt{\glscategory{#1}}%
\glshasattribute{#1}{glossnamefont}%
{%
\protected@edef\@glxtr@attrval{\glsetattribute{#1}{glossnamefont}}%
\ifcsdef{\@glxtr@attrval}%
{%
\letcs{\@glxtr@glossnamefont}{\@glxtr@attrval}%
}%
{%
\GlossariesExtraWarning{Unknown control sequence name
'\@glxtr@attrval' supplied in glossnamefont attribute
for entry '#1'. Reverting to default \string\glnamefont}%
\let\@glxtr@glossnamefont\glnamefont
}%
}%
{\let\@glxtr@glossnamefont\glnamefont}%
\@glxtr@glossnamefont{\Glsentryname{#1}}%

```

Do post-name hook:

```

\glxtrpostnamehook{#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.

`\glxtrpostnamehook` Hook to append stuff after the name is displayed in the glossary. The argument is the entry's label.

```

\newcommand*{\glxtrpostnamehook}[1]{%
\let\@glnumberformat\@glxtr@defaultnumberformat
\glxtrdoautoindexname{#1}{indexname}%

```

Allow additional code regardless of category:

```
\glsextrapostnamehook{#1}%
```

Allow categories to hook in here.

```

\csuse{glxtrpostname\glscategory{#1}}%
}

```

`\glsextrapostnamehook`

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

`\gldefpostname` Provide a convenient command for defining the post-name hook for the given category.

```

\newcommand*{\gldefpostname}[2]{%
\csdef{glxtrpostname#1}{#2}%
}

```

`\glxtr@setaccessdisplay`

```
\@ifpackageloaded{glossaries-accsupp}
{
  \newcommand*{\glxtr@setaccessdisplay}[1]{%
    \ifcsdef{gls#1accessdisplay}%
      {\letcs\glxtr@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\glxtr@accessdisplay{gls\@gls@thisval accessdisplay}}%
      {\let\glxtr@accessdisplay\@firstoftwo}%
    }%
  }
}
{%
  \newcommand*{\glxtr@setaccessdisplay}[1]{%
    \let\glxtr@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:

```
\glxtr@setaccessdisplay{#2}%
```

Set the abbreviation format:

```
\glsssetabbrvfmt{\glscategory{#1}}%
\glshasattribute{#1}{glossnamefont}%
{%
  \protected@edef\@glxtr@attrval{\glsggetattribute{#1}{glossnamefont}}%
  \ifcsdef{\@glxtr@attrval}%
    {%
      \letcs{\@glxtr@glossnamefont}{\@glxtr@attrval}%
    }%
  }
```



```

    {%
      \GlossariesExtraWarning{Unknown control sequence name
        '\@glstr@attrval' supplied in glossnamefont attribute
        for entry '#1'. Reverting to default \string\glsnamefont}%
      \let\@glstr@glossnamefont\glsnamefont
    }%
  }%
  {\let\@glstr@glossnamefont\glsnamefont}%
  \glsifattribute{#1}{glossname}{firstuc}%
  {%
    \@glstr@accessdisplay
    {\@glstr@glossnamefont{\@Gls@entry@field{#1}{#2}}}%
    {#1}%
  }%
  {%
    \glsifattribute{#1}{glossname}{title}%
    {%
      \@glstr@do@titlecaps@warn
      \@glstr@accessdisplay
      {\@glstr@glossnamefont{\glstrfieldtitlecase{#1}{#2}}}%
      {#1}%
    }%
    {%
      \glsifattribute{#1}{glossname}{uc}%
      {%
        \letcs{\glo@name}{glo@\glstetoklabel{#1}0#2}%
        \@glstr@accessdisplay
        {\@glstr@glossnamefont{\glssupercase{\glo@name}}}%
        {#1}%
      }%
      {%
        \letcs{\glo@name}{glo@\glstetoklabel{#1}0#2}%
        \@glstr@accessdisplay
        {\expandafter\@glstr@glossnamefont\expandafter{\glo@name}}%
        {#1}%
      }%
    }%
  }%
  }%
  \glsxtrpostnamehook{#1}%
}

```

Do post-name hook.

```

\glsxtrpostnamehook{#1}%
}%
}

```

`\if@glstr@format@override` Determines if the format key should override the indexing attribute value.

```

\newif\if@glstr@format@override
\@glstr@format@overridefalse

```

If overriding is enabled, the `\glshypernumber` command will have to be redefined in the index to use `\hyperpage` instead.

trEnableIndexFormatOverride

```
\@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
```

\glsxtrdoautoindexname

```
\newcommand*{\glsxtrdoautoindexname}[2]{%
  \glsasattribute{#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}%
}%
```

```

    {}%
  }

\glxtrautoindex
\newcommand*\glxtrautoindex{\index}

\glxtrautoindexesc
\newcommand*\glxtrautoindexesc{%
  \@gls@checkmkidxchars\@glo@sort
  \@glxtr@autoindex@doextra@esc\@glo@sort
}

\@glxtr@autoindex@setname Assign \@glo@name for use with indexname attribute.
\newcommand*\@glxtr@autoindex@setname}[1]{%
  \protected@edef\@glo@name{\glxtrautoindexentry{#1}}%
  \glxtrautoindexassignsort{\@glo@sort}{#1}%
  \glxtrautoindexesc
  \epreto\@glo@name{\@glo@sort\@glxtr@autoindex@at}%
}

\glxtrautoindexentry Command used for the actual part when auto-indexing.
\newcommand*\glxtrautoindexentry}[1]{\string\glsentryname{#1}}

\glxtrautoindexassignsort Used to assign the sort value when auto-indexing.
\newcommand*\glxtrautoindexassignsort}[2]{%
  \glsletentryfield{#1}{#2}{sort}%
}

\glxtr@autoindex@doextra@esc
\newcommand*\@glxtr@autoindex@doextra@esc}[1]{%
Escape the escape character unless it has already been escaped.
  \ifx\@glxtr@autoindex@esc\@gls@quotechar
  \else
    \def\@gls@checkedmkidx{%
    \edef\@glxtr@checkspch{%
      \noexpand\@glxtr@autoindex@escquote\expandonce{#1}%
      \noexpand\@empty\@glxtr@autoindex@esc\noexpand\@nnil
      \@glxtr@autoindex@esc\noexpand\@empty\noexpand\@glxtr@endescspch}%
    \@glxtr@checkspch
    \let#1\@gls@checkedmkidx\relax
    \fi
  \fi
Escape actual character unless it has already been escaped.
  \ifx\@glxtr@autoindex@at\@gls@actualchar
  \else
    \def\@gls@checkedmkidx{%
    \edef\@glxtr@checkspch{%
      \noexpand\@glxtr@autoindex@escat\expandonce{#1}%
      \noexpand\@empty\@glxtr@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
        \@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##2##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\@glstr@autoindex@encap{#1}%
  \def\@glstr@autoindex@escencap##1##2##3\@glstr@endescspch{%
    \@glstr@autoindex@escspch{#1}{\@glstr@autoindex@escencap}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetEncapChar{}
\@onlypreamble\GlsXtrSetEncapChar

```

`\@glstr@autoindex@level` Level character for use with `\index`.

```

\newcommand*{\@glstr@autoindex@level}{}

```

`\GlsXtrSetLevelChar` Set the encap character.

```

\newcommand*{\GlsXtrSetLevelChar}[1]{%
  \gdef\@glstr@autoindex@level{#1}%
  \def\@glstr@autoindex@esclevel##1##2##3\@glstr@endescspch{%
    \@glstr@autoindex@escspch{#1}{\@glstr@autoindex@esclevel}{##1}{##2}{##3}%
  }%
}
\GlsXtrSetLevelChar{!}
\@onlypreamble\GlsXtrSetLevelChar

```

`\@glstr@autoindex@esc` Escape character for use with `\index`.

```

\newcommand*{\@glstr@autoindex@esc}{"}

```

`\GlsXtrSetEscChar` Set the escape character.

```

\newcommand*{\GlsXtrSetEscChar}[1]{%
  \gdef\@glstr@autoindex@esc{#1}%
  \def\@glstr@autoindex@escquote##1##2##3\@glstr@endescspch{%
    \@glstr@autoindex@escspch{#1}{\@glstr@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}}
{}
```

\@glxtr@gobbleto@endescspch

```
\def\@glxtr@gobbleto@endescspch#1\@glxtr@endescspch{}
```

```
\@@glxtr@autoindex@escspch{<char>}{<cs>}{<pre>}{<mid>}
{<post>}
```

\@@glxtr@autoindex@esc@spch

```
\newcommand*{\@glxtr@autoindex@escspch}[5]{%
\@glx@tmpb=\expandafter{\@glx@checkedmkidx}%
\toks@={#3}%
\ifx\@nnil#3\relax
\def\@glxtr@checkspch{\@glxtr@gobbleto@endescspch#5\@glxtr@endescspch}%
\else
\ifx\@nnil#4\relax
\edef\@glx@checkedmkidx{\the\@glx@tmpb\the\toks@}%
\def\@glxtr@checkspch{\@glxtr@gobbleto@endescspch
#4#5\@glxtr@endescspch}%
\else
\edef\@glx@checkedmkidx{\the\@glx@tmpb\the\toks@
\@glxtr@autoindex@esc#1}%
\def\@glxtr@checkspch{#2#5#1\@nnil#1\@glxtr@endescspch}%
\fi
\fi
\@glxtr@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]{%
\glstexpdfstring{\@glossentrysymbol{#1}}{\glstentrypdfsymbol{#1}}%
}
```

\glstentrypdfsymbol May be redefined to a field that expands to a value that's more suitable for PDF bookmarks.

```
\newcommand{\glstentrypdfsymbol}[1]{\glstentrysymbol{#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\@glxtr@attrval{\glsetattribute{#1}{glosssymbolfont}}%
        \ifcsdef{\@glxtr@attrval}%
        {%
          \letcs{\@glxtr@glosssymbolfont}{\@glxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
            '\@glxtr@attrval' supplied in glosssymbolfont attribute
            for entry '#1'. Ignoring}%
          \let\@glxtr@glosssymbolfont\@firstofone
        }%
      }%
      {\let\@glxtr@glosssymbolfont\@firstofone}%
      \@glxtr@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@glxtr@enabletagging\@glxtr@enabletagging
}
\@onlypreamble\GlsXtrEnableInitialTagging
```

`\@glxtr@enabletagging` Starred version undefines command.

```
\newcommand*{\s@glxtr@enabletagging}[2]{%
  \undef#2%
```

```

\@glstr@enabletagging{#1}{#2}%
}

```

`\@glstr@enabletagging` Internal command.

```

\newcommand*{\@glstr@enabletagging}[2]{%

```

Set attributes for categories given in the first argument.

```

\@for\@glstr@cat:=#1\do
{%
  \ifdefempty\@glstr@cat
  {}%
  {\glsssetcategoryattribute{\@glstr@cat}{tagging}{true}}%
}%
\newrobustcmd*#2[1]{##1}%
\def\@glstr@taggingcs{#2}%
\renewcommand*\@glstr@activate@initialtagging{%
  \let#2\@glstr@tag
}%
\ifundef\@glstr@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{\@glstr@do@titlecaps@warn}{%
    \GlossariesExtraWarning{mfirstuc.sty too old. Title Caps
      support not available}%
  }
}

```

One warning should suffice.

```

\let\@glstr@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

\@glxtr@do@titlecaps@warn Do warning if title case not supported.
\newcommand*{\@glxtr@do@titlecaps@warn}{}

\@glxtr@activate@initialtagging Used in \printglossary but at least v4.19 of glossaries required.
\newcommand*{\@glxtr@activate@initialtagging}{}

\@glxtr@tag Definition of tagging command when used in glossary.
\newrobustcmd*{\@glxtr@tag}[1]{%
  \glsifattribute{\glscurrententrylabel}{tagging}{true}%
  {\glxtrtagfont{#1}}{#1}%
}

\glxtrtagfont Used in the glossary.
\newcommand*{\glxtrtagfont}[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.
\ifdefined\@gls@preglossaryhook
{
  \renewcommand*{\@gls@preglossaryhook}{%
    \@glxtr@activate@initialtagging
  }
  Since the glossaries are automatically scoped, \@glxtr@org@postdescription
  shouldn't already be defined, but check anyway just as a precautionary measure.
  \ifundefined\@glxtr@org@postdescription
  {%
    \let\@glxtr@org@postdescription\glspostdescription
    \renewcommand*{\glspostdescription}{%
      \ifglentryexists{\glscurrententrylabel}%
      {%
        \glxtrpostdescription
        \@glxtr@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}{}%
```

`\glsxtrpostdescsterm` This is redundant as it doesn't match any common categories. \newterm sets the category to index.

```
\newcommand*{\glsxtrpostdescsterm}{}%
```

`\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}{%
  \ifglentryexists{\glslabel}{\glsxtrpostlinkhook}{}%
}
```

`\glsxtrpostlinkhook` The entry label should already be stored in \glslabel by \@gls@link.

```
\newcommand*{\glsxtrpostlinkhook}{%
  \glsxtrdiscardperiod{\glslabel}%
  {\glsxtrpostlinkendsentence}%
}
```

```

    {\glxstrifcustomdiscardperiod
     {\glxstrifperiod{\glxstrpostlinkendsentence}{\glxstrpostlink}}%
     {\glxstrpostlink}%
    }%
}

```

`\glxstrifcustomdiscardperiod` Allow user to provide a custom check. Should expand to #2 if no check is required otherwise expand to #1.

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

`\glxstrpostlink`

```

\newcommand*{\glxstrpostlink}{%
  \csuse{\glxstrpostlink\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 `\glxstrpostlink`.

```
\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{\glxstrpostlink#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{\glxstrpostlink#1}
        {\csdef{\glxstrpostlink#1}{#2}}%
        {\cspreto{\glxstrpostlink#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
}%
{%

```

Assume the full stop was discarded because the entry ends with a period, so adjust the spacefactor.

```

  \spacefactor\sfcode'\. \relax
}%
}

```

`trpostlinkAddDescOnFirstUse` 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{\glsaccessfmt{desc}}{\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}{%

```

```

\glxtrifwasfirstuse
{%
  \glxtrgenentrytextfmt{ }\glxtrparen
  {%
    \ifglshassymbol{\glslabel}%
    {\glssaccessfmtsymboll{\glxtrgenentrytextfmt}{\glslabel}%
    \expandafter\glxtrgenentrytextfmt\expandafter{\glxtrpostlinkSymbolDescSep}}%
    {}%
    \glssaccessfmdesc{\glxtrgenentrytextfmt}{\glslabel}%
  }%
}%
{}%
}

\glxtrpostlinkSymbolDescSep Separator used in the above
\newcommand*{\glxtrpostlinkSymbolDescSep}{, }

\discardperiodretainfirstuse
\newcommand*{\glxtrdiscardperiodretainfirstuse}[3]{%
\glxtrifwassubsequentorshort{\glxtrifperiod{#2}{#3}}{#3}%
}

\glxtrdiscardperiod 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*{\glxtrdiscardperiod}[3]{%
\glssifattribute{#1}{retainfirstuseperiod}{true}%
{\glxtrdiscardperiodretainfirstuse{#1}{#2}{#3}}%
{%
\glssifattribute{#1}{discardperiod}{true}%
{%
\glssifplural
{%
\glssifattribute{#1}{pluraldiscardperiod}{true}%
{\glxtrifperiod{#2}{#3}}%
{#3}%
}%
}%
\glxtrifperiod{#2}{#3}%
}%
}%
{#3}%
}

\glxtrifperiod 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*\glxstrifperiod}[1]{\new@ifnextchar.\{@firstoftwo{#1}}}
```

Sometimes it's useful to test if there's a punctuation character following the glossary entry.

`\glxstr@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 characters (such as `' '`).

```
\newcommand*\glxstr@punclist{.,:;?!}
```

`\glxstraddpunctuationmark` Add character to punctuation list.

```
\newcommand*\glxstraddpunctuationmark}[1]{\appto\glxstr@punclist{#1}}
```

`\glxstrsetpunctuationmarks` Reset the punctuation list.

```
\newcommand*\glxstrsetpunctuationmarks}[1]{\def\glxstr@punclist{#1}}
```

`\glxstrifnextpunc`

```
\glxstrifnextpunc{<true part>}{<false part>}
```

Test if this is followed by a punctuation mark. (Adapted from `\new@ifnextchar`.)

```
\newcommand*\glxstrifnextpunc}[2]{%
  \def\reserved@a{#1}%
  \def\reserved@b{#2}%
  \futurelet\@glspunc@token\glxstr@ifnextpunc
}
```

`\glxstr@ifnextpunc`

```
\newcommand*\glxstr@ifnextpunc{%
  \glxstr@ifpunctoken{\@glspunc@token}{\let\reserved@b\reserved@a}{}%
  \reserved@b
}
```

`\glxstr@ifpunctoken` Test if the token given in the first argument is in the punctuation list.

```
\newcommand*\glxstr@ifpunctoken}[1]{%
  \expandafter\@glxstr@ifpunctoken\expandafter#1\glxstr@punclist\@nnil
}
```

`\@glxstr@ifpunctoken`

```
\def\@glxstr@ifpunctoken#1#2{%
  \let\reserved@d=#2%
  \ifx\reserved@d\@nnil
    \let\glxstr@next\@glxstr@notfoundinlist
  \else
    \ifx#1\reserved@d
      \let\glxstr@next\@glxstr@foundinlist
    \else
      \let\glxstr@next\@glxstr@ifpunctoken
    \fi
}
```

```

\fi
\glxtr@next#1%
}

```

```

\@glxtr@foundinlist

```

```

\def\@glxtr@foundinlist#1\@nnil{\@firstoftwo}

```

```

\@glxtr@notfoundinlist

```

```

\def\@glxtr@notfoundinlist#1{\@secondoftwo}

```

```

\glxtrdopostpunc{<code>}

```

```

\glxtrdopostpunc

```

If this is followed by a punctuation character, do *<code>* after the character otherwise do *<code>* before whatever comes next.

```

\newrobustcmd{\glxtrdopostpunc}[1]{%
\glxtrifnextpunc{\@glxtr@swaptwo{#1}}{#1}%
}

```

```

\@glxtr@swaptwo

```

```

\newcommand{\@glxtr@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{glxtrabbrv}{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{glxtrabbrvpl}{shortplural}{%
\def\@glsc@shortpl{#1}%
}

```

Similarly for the long plural form.

```

\define@key{glxtrabbrvpl}{longplural}{%
\def\@glsc@longpl{#1}%
}

```

Token registers for the short plural and long plural, provided for use in the abbreviation style definitions.

```
\glsshortpltok
\newtoks\glsshortpltok
```

```
\glslongpltok
\newtoks\glslongpltok
```

`\@glsxtr@insertdots` Provided in case user wants to automatically insert dots between each letter of the abbreviation. This should be applied before defining the abbreviation to optimise the document build. (Otherwise, it would have to be done each time the short form is required, which is an unnecessary waste of time.) For this to work the short form must be expanded when passed to `\newabbreviation`. Note that explicitly using the `short` or `shortplural` keys will override this.

```
\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
}
```


`\@glxtr@mark@wordseps`

```
\def\@glxtr@mark@wordseps#1#2 #3{%
  \ifdefempty{#1}%
  {\def#1{\protect\glxtrword{#2}}}%
  {\appto#1{\protect\glxtrwordsep\protect\glxtrword{#2}}}%
  \ifx\@nnil#3\relax
    \let\@glxtr@mark@wordseps@next\relax
  \else
    \def\@glxtr@mark@wordseps@next{%
      \@glxtr@mark@wordseps#1#3}%
    \fi
  \@glxtr@mark@wordseps@next
}
```

`\newabbreviation` Define a new generic abbreviation.

```
\newcommand*{\newabbreviation}[4][[]]{%
  \glxtr@newabbreviation{#1}{#2}{#3}{#4}%
}
```

`\glxtr@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*{\glxtr@newabbreviation}[4]{%
  \glskeylisttok{#1}%
  \glslabeltok{#2}%
  \glsshorttok{#3}%
  \glslongtok{#4}%
}
```

Save the original short and long values (before attribute settings modify them).

```
\def\glxtrorgshort{#3}%
\def\glxtrorglong{#4}%

\def\glxtrorgkeylist{#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*{\glxtrabbrv}{#1}%
```

Save remaining keys, just in case any hook also uses `\setkeys`

```
\let\@glxtrabbrv@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@keywordseps\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@keywordseps\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{\gls@categorylabel}{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={\gls@categorylabel},%
  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.

```
\@glstr@addabbreviationlist{\glstrytype{\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
  }

```

`\glstrnewabbrevpresetkeyhook` Hook for extra stuff in `\newabbreviation`

```
\newcommand*{\glstrnewabbrevpresetkeyhook}[3]{}%
```

`\GlsXtrPostNewAbbreviation` Hook used by abbreviation styles.

```
\newcommand*{\GlsXtrPostNewAbbreviation}{}%
```

`\newabbreviationhook` Hook for use with `\newabbreviation`.

```
\newcommand*{\newabbreviationhook}{}%
```

`\CustomAbbreviationFields`

```
\newcommand*{\CustomAbbreviationFields}{}%
```

`\glstrparen` For the parenthetical styles.

```

\newcommand*{\glstrparen}[1]{%
  \glstrgenentrytextfmt{(#1\glstrgenentrytextfmt)}%
}

```

`\glstrfullformat` Full format without case change.

```

\newcommand*{\glstrfullformat}[2]{%
  \ifglstrinsertinside
    \glstrfirstlongfont{\glstraccessfmtlong{#2}{\glstrgenentrytextfmt}{#1}}%
  \else
    \glstrfirstlongfont{\glstraccessfmtlong{}{\glstrgenentrytextfmt}{#1}}%
    \glstrgenentrytextfmt{#2}%
  \fi
  \glstrfullsep{#1}%
  \glstrparen{\protect\glstrfirstabbrvfont
    {\glstraccessfmtshort{}{\glstrgenentrytextfmt}{#1}}}%
}

```

`\Glsxtrfullformat` Full format with case change.

```
\newcommand*{\Glsxtrfullformat}[2]{%}
```

```

\ifglstrinsertinside
\glstrfirstlongfont{\Glsaccessfmtlong{#2}{\glstrgenentrytextfmt}{#1}}%
\else
\glstrfirstlongfont{\Glsaccessfmtlong{}}{\glstrgenentrytextfmt}{#1}}%
\glstrgenentrytextfmt{#2}%
\fi
\glstrfullsep{#1}%
\glstrparen{\protect\glstrfirstabbrvfont
  {\glstraccessfmtshort{}}{\glstrgenentrytextfmt}{#1}}}%
}
\glsmfuaddmap{\glstrfullformat}{\Glsxtrfullformat}

```

`\Glsxtrfullformat` Full format with all caps.

```

\newcommand*{\Glsxtrfullformat}[2]{%
\ifglstrinsertinside
\glstrfirstlongfont{\Glsaccessfmtlong{#2}{\glstrgenentrytextfmt}{#1}}%
\else
\glstrfirstlongfont{\Glsaccessfmtlong{}}{\glstrgenentrytextfmt}{#1}}%
\glstruppercase{\glstrgenentrytextfmt{#2}}%
\fi
\glstrfullsep{#1}%
\glstrparen{\protect\glstrfirstabbrvfont
  {\Glsaccessfmtshort{}}{\glstrgenentrytextfmt}{#1}}}%
}
\glsmfublocker{\Glsxtrfullformat}

```

`\glstrfullplformat` Plural full format without case change.

```

\newcommand*{\glstrfullplformat}[2]{%
\ifglstrinsertinside
\glstrfirstlongfont{\glstraccessfmtlongpl{#2}{\glstrgenentrytextfmt}{#1}}%
\else
\glstrfirstlongfont{\glstraccessfmtlongpl{}}{\glstrgenentrytextfmt}{#1}}%
\glstrgenentrytextfmt{#2}%
\fi
\glstrfullsep{#1}%
\glstrparen{\protect\glstrfirstabbrvfont
  {\glstraccessfmtshortpl{}}{\glstrgenentrytextfmt}{#1}}}%
}

```

`\Glsxtrfullplformat` Plural full format with case change.

```

\newcommand*{\Glsxtrfullplformat}[2]{%
\ifglstrinsertinside
\glstrfirstlongfont{\Glsaccessfmtlongpl{#2}{\glstrgenentrytextfmt}{#1}}%
\else
\glstrfirstlongfont{\Glsaccessfmtlongpl{}}{\glstrgenentrytextfmt}{#1}}%
\glstrgenentrytextfmt{#2}%
\fi
\glstrfullsep{#1}%
\glstrparen{\protect\glstrfirstabbrvfont

```

```

        {\glsaccessfmtshortpl}{\glsxtrgenentrytextfmt}{#1}}%
    }
    \glsmfuaddmap{\glsxtrfullplformat}{\Glsxtrfullplformat}

\GLSxtrfullplformat Full format with all caps.
    \newcommand*{\GLSxtrfullplformat}[2]{%
        \ifglsxtrininsertinside
            \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 `\glentryfull` 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 `\glxtrfull` set of commands instead. If expandable sentence case is required, use `\MFUsentencecase` on the non-case-change version.

`\glentryfull`

```
\renewcommand*{\glentryfull}[1]{\glxtrinlinefullformat{#1}{}}
```

`\Glsentryfull`

```
\renewcommand*{\Glsentryfull}[1]{\Glsxtrinlinefullformat{#1}{}}
\glsmfuaddmap{\glentryfull}{\Glsentryfull}
```

`\glentryfullpl`

```
\renewcommand*{\glentryfullpl}[1]{\glxtrinlinefullplformat{#1}{}}
```

`\Glsentryfullpl`

```
\renewcommand*{\Glsentryfullpl}[1]{\Glsxtrinlinefullplformat{#1}{}}
\glsmfuaddmap{\glentryfullpl}{\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{\glxtrgenentrytextfmt{#1}}}%
}
```

`\glsfirstxppabbrvfont` Expand to appropriate formatting command.

```
\newcommand*{\glsfirstxppabbrvfont}[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{\glstrgenentrytextfmt{#1}}%
}

```

`\glspabbrvfont` Expand to appropriate formatting command.

```

\newcommand*{\glspabbrvfont}[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 `\glstrlong`.

```

\newcommand*{\glslongfont}[1]{\glslongdefaultfont{#1}}

```

`\glsinnerfmtlongfont` Include inner formatting command.

```

\newrobustcmd*{\glsinnerfmtlongfont}[1]{%
\glslongfont{\glstrgenentrytextfmt{#1}}%
}

```

`\glslongfont` Expand to appropriate formatting command.

```

\newcommand*{\glslongfont}[2]{%
\glsifcategoryattributetrue{#2}{markwords}%
{\protect\glslongfont{#1}}%
{\glsinnerfmtlongfont{#1}}%
}

```

`\glslongdefaultfont` Default font changing command used for the long form in commands like `\glstrlong`.

```

\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{\glstrgenentrytextfmt{#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{#3}%
```

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\glsapscase\@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
}

\glxtrsetupfulldefs
\newcommand*{\glxtrsetupfulldefs}{%
  \let\glxtrifwasfirsttuse\@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{\glxtrfull}{\Glsxtrfull}

\@Glsxtr@full Low-level macro:
\def\@Glsxtr@full#1#2[#3]{%
  \def\glxtrcurrentfield{%
    \glsoifexists{#2}%
    {%
      \glsssetabbrvfmt{\glscategory{#2}}%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \let\glxtrifwasglslike\@secondoftwo
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@secondofthree
      \glxtrfullsaveinsert{#2}{#3}%
    }
  }
  \def\glscustomtext{\Glsxtrinlinefullformat{#2}{#3}}%
  \glxtrsetupfulldefs
  \@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\glxtrcurrentfield{%
    \glsoifexists{#2}%
    {%

```

```

\glsetabbrvfmt{\glscategory{#2}}%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glxtrifwasglslike\@secondoftwo
\let\gl@ifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\glxtrfullsaveinsert{#2}{#3}%

```

The innertextformat support should be provided within the inline command.

```

\def\glscustomtext{\GLSxtrinlinefullformat{#2}{#3}}%
\glxtrsetupfulldefs
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\glsmfublocker{\GLSxtrfull}

```

`\glxtrfullpl` Plural full form (no case-change).

```

\newrobustcmd*{\glxtrfullpl}{\@gl@hyp@opt\ns@glxtrfullpl}
\newcommand*\ns@glxtrfullpl[2][]{%
  \new@ifnextchar[{\@glxtr@fullpl{#1}{#2}}%
    {\@glxtr@fullpl{#1}{#2}[]}%
}

```

`\@glxtr@fullpl` Low-level macro:

```

\def\@glxtr@fullpl#1#2[#3]{%
  \def\glxtrcurrentfield{%

```

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).

```

\@glxtr@record{#1}{#2}{glslink}%
\glsoifexists{#2}%
{%
  \glsetabbrvfmt{\glscategory{#2}}%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \let\glxtrifwasglslike\@secondoftwo
  \let\gl@ifplural\@firstoftwo
  \let\glscapscase\@firstofthree
  \glxtrfullsaveinsert{#2}{#3}%
}

```

The innertextformat support should be provided within the inline command.

```

\def\glscustomtext{\glxtrinlinefullplformat{#2}{#3}}%
\glxtrsetupfulldefs
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

`\Glsxtrfullpl` Plural full form (first letter uppercase).

```

\newrobustcmd*{\Glsxtrfullpl}{\@gl@hyp@opt\ns@Glsxtrfullpl}

```

```

\newcommand*\ns@Glsxtrfullpl[2][]{%
  \new@ifnextchar[{\@Glsxtr@fullpl{#1}{#2}}{%
    {\@Glsxtr@fullpl{#1}{#2}[]}%
  }
\glsmfuaddmap{\glxtrfullpl}{\Glsxtrfullpl}

```

`\@Glsxtr@fullpl` Low-level macro:

```

\def\@Glsxtr@fullpl#1#2[#3]{%
  \def\glxtrcurrentfield{%

```

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).

```

\@glxtr@record{#1}{#2}{glslink}%
\glsoifexists{#2}%
{%
  \glsssetabbrvfmt{\glscategory{#2}}%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \let\glxtrifwasglslike\@secondoftwo
  \let\glsifplural\@firstoftwo
  \let\glscapscase\@secondofthree
  \glxtrfullsaveinsert{#2}{#3}%

```

The `innertextformat` support should be provided within the inline command.

```

\def\glscustomtext{\Glsxtrinlinefullplformat{#2}{#3}}%
\glxtrsetupfulldefs
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

`\GLSxtrfullpl` Plural full form (all upper case).

```

\newrobustcmd*\@GLSxtrfullpl{\@gl@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\glxtrcurrentfield{%

```

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).

```

\@glxtr@record{#1}{#2}{glslink}%
\glsoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

```

```

\let\glxtrifwasglslike\@secondoftwo
\let\glsifplural\@firstoftwo
\let\glscapscase\@thirdofthree
\glxtrfullsaveinsert{#2}{#3}%

```

The innertextformat support should be provided within the inline command.

```

\def\glscustomtext{%
  \GLSxtrinlinefullplformat{#2}{#3}}%
\glxtrsetupfulldefs
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

The short and long forms work in a similar way to acronyms.

`\glxtrshort`

```

\newrobustcmd*{\glxtrshort}{\@gls@hyp@opt\@ns@glxtrshort}

```

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

```

\newcommand*{\ns@glxtrshort}[2][{}]{%
  \new@ifnextchar[{\@glxtrshort{#1}{#2}}{\@glxtrshort{#1}{#2}[]}%
}

```

Read in the final optional argument:

```

\def\@glxtrshort#1#2[#3]{%
  \def\glxtrcurrentfield{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).

```

\@glxtr@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\glxtrifwasglslike\@secondoftwo
\let\glxtrifwasfirstuse\@secondoftwo
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\glxtrsavesaveinsert{#2}{#3}%
\def\glscustomtext{%
  \glxtrshortformat{#2}{#3}{\glsabbrvfont}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

`\Glsxtrshort`

```
\newrobustcmd*{\Glsxtrshort}{\@gls@hyp@opt\@ns@Glsxtrshort}
\glsmfuaddmap{\glxtrshort}{\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\glxtrcurrentfield{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).

```
\@glxtr@record{#1}{#2}{\glslink}%
\glsoifexists{#2}%
{%
  \glsssetabbrvfmt{\glscategory{#2}}%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \let\glxtrifwasglslike\@secondoftwo
  \let\glxtrifwasfirsttuse\@secondoftwo
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@secondofthree
  \glxtrsaveinsert{#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\glxtrcurrentfield{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).

```

\@glxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
  \glssetabbrvfmt{\glscategory{#2}}%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \let\glxtrifwasglslike\@secondoftwo
  \let\glxtrifwasfirstuse\@secondoftwo
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@thirdofthree
  \glxtrsinsert{#2}{#3}%
  \def\glscustomtext{\GLSxtrshortformat{#2}{#3}{\glsabbrvfont}}%
  \@gl@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

`\glxtrsetlongfirstuse` Assigns `\glxtrifwasfirstuse` for the long commands. The argument is the entry label. This now makes commands such as `\glxtrlong` simulate first use.

```

\newcommand{\glxtrsetlongfirstuse}[1]{%
  \let\glxtrifwasfirstuse\@firstoftwo
}

```

`\glxtrlong`

```

\newrobustcmd*{\glxtrlong}{\@gl@hyp@opt\@ns@glxtrlong}

```

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

```

\newcommand*{\ns@glxtrlong}[2][{}]{%
  \new@ifnextchar[{\@glxtrlong{#1}{#2}}{\@glxtrlong{#1}{#2}[]}%
}

```

Read in the final optional argument:

```

\def\@glxtrlong#1#2[#3]{%
  \def\glxtrcurrentfield{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).

```

\@glxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \let\glxtrifwasglslike\@secondoftwo
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@firstofthree
  \glxtrsetlongfirstuse{#2}%
  \glxtrsinsert{#2}{#3}%
  \def\glscustomtext{%
    \glxtrlongformat{#2}{#3}{\glslongfont}%
  }
}

```



```

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

```

\Glsxtrlong

```

\newrobustcmd*{\Glsxtrlong}{\@gls@hyp@opt\@ns@Glsxtrlong}
\glsmfuaddmap{\glxtrlong}{\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\glxtrcurrentfield{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\glxtrifwasglslike\@secondoftwo
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \glxtrsetlongfirstuse{#2}%
    \glxtrsaveinsert{#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\glxtrcurrentfield{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).

```
\@glxtr@record{#1}{#2}{\glslink}%
\glsdofexists{#2}%
{%
\let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
\let\glxtrifwasglsl@like\@secondoftwo
\let\glslifplural\@secondoftwo
\let\glscaps@case\@thirdofthree
\glxtrsetlongfirstuse{#2}%
\glxtrsaveinsert{#2}{#3}%
\def\glscustomtext{%
\GLSxtrlongformat{#2}{#3}{\glslongfont}%
}%
\@glsl@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
```

Plural short forms:

`\glxtrshortpl`

```
\newrobustcmd*{\glxtrshortpl}{\@glsl@hyp@opt\@ns@glxtrshortpl}
```

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

```
\newcommand*{\ns@glxtrshortpl}[2][{}]{%
\new@ifnextchar[{\@glxtrshortpl{#1}{#2}}{\@glxtrshortpl{#1}{#2}[]}%
}
```

Read in the final optional argument:

```
\def\@glxtrshortpl#1#2[#3]{%
\def\glxtrcurrentfield{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).

```
\@glxtr@record{#1}{#2}{\glslink}%
\glsdofexists{#2}%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
\let\glxtrifwasglsl@like\@secondoftwo
\let\glxtrifwasfirstuse\@secondoftwo
\let\glslifplural\@firstoftwo
\let\glscaps@case\@firstofthree
```

```

\glxtrsaveinsert{#2}{#3}%
\def\glscustomtext{%
  \glxtrshortplformat{#2}{#3}{\glabbrvfont}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

\Glsxtrshortpl

```

\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\glxtrcurrentfield{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).

```

\@glxtr@record{#1}{#2}{\glslink}%
\glsoifexists{#2}%
{%
  \glsssetabbrvfmt{\glscategory{#2}}%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \let\glxtrifwasglslike\@secondoftwo
  \let\glxtrifwasfirstuse\@secondoftwo
  \let\glsifplural\@firstoftwo
  \let\glscapscase\@secondofthree
  \glxtrsaveinsert{#2}{#3}%
  \def\glscustomtext{%
    \Glsxtrshortplformat{#2}{#3}{\glabbrvfont}%
  }%
  \@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\glxtrcurrentfield{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).

```

\@glxtr@record{#1}{#2}{glslink}%
\glstoifexists{#2}%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
\let\glxtrifwasglsl@like\@secondoftwo
\let\glxtrifwasfirstuse\@secondoftwo
\let\glsl@plural\@firstoftwo
\let\glscaps@case\@thirdofthree
\glxtrsaveinsert{#2}{#3}%
\def\glscustomtext{%
\GLSxtrshortplformat{#2}{#3}{\glsl@abbrvfont}%
}%
\@glsl@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

Plural long forms:

`\glxtrlongpl`

```

\newrobustcmd*{\glxtrlongpl}{\@glsl@hyp@opt\@ns@glxtrlongpl}

```

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

```

\newcommand*{\@ns@glxtrlongpl}[2][{}]{%
\new@ifnextchar[{\@glxtrlongpl{#1}{#2}}{\@glxtrlongpl{#1}{#2}[]}%
}

```

Read in the final optional argument:

```

\def\@glxtrlongpl#1#2[#3]{%
\def\glxtrcurrentfield{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).

```

\@glxtr@record{#1}{#2}{glslink}%
\glstoifexists{#2}%
{%
\let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
\let\glxtrifwasglsl@like\@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
}

\Glsxtrlongpl
\newrobustcmd*{\Glsxtrlongpl}{\@gls@hyp@opt\ns@Glsxtrlongpl}
\glsmfuaddmap{\glsxtrlongpl}{\Glsxtrlongpl}
Define the un-starred form. Need to determine if there is a final optional argu-
ment
\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\glstrcurrentfield{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).

```
\@glstr@record{#1}{#2}{\glslink}%
\glsoifexists{#2}%
{%
  \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
  \let\glstrifwasglsl\@secondoftwo
  \let\glslplural\@firstoftwo
  \let\glscapscase\@thirdofthree
  \glstrsetlongfirstuse{#2}%
  \glstrsaveinsert{#2}{#3}%
  \def\glscustomtext{%
    \GLSxtrlongplformat{#2}{#3}{\glslongfont}%
  }%
  \@glsl@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
```

`\glissetabbrvfmt` Set the current format for the given category (or the abbreviation category if unset).

```
\newcommand*{\glissetabbrvfmt}[1]{%
  \ifcsdef{@glssabbrv@current@#1}%
  {\glstr@applyabbrvfmt{\csname @glssabbrv@current@#1\endcsname}}%
  {\glstr@applyabbrvfmt{\@glssabbrv@current@abbreviation}}%
}
```

`\glseuseabbrvfont` Provide a way to use the abbreviation font for a given category for arbitrary text.

```
\newrobustcmd*{\glseuseabbrvfont}[2][{\glissetabbrvfmt{#2}\glssabbrvfont{#1}}]
```

`\glsuselongfont` Provide a way to use the long font for a given category for arbitrary text.

```
\newrobustcmd*{\glsuselongfont}[2][{\glissetabbrvfmt{#2}\glslongfont{#1}}]
```

`\glstrgenabbrvfmt` Similar to `\glsgenacfmt`, but for abbreviations. The expansion is to ensure that `\glinsert` 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*{\glxtrgenabbrvfmt}{%
\ifdefempty\glscustomtext
{%
\ifglused\glslabel
{%

```

Subsequent use:

```

\glcifplural
{%

```

Subsequent plural form:

```

\glscapscase
{%

```

Subsequent plural form, don't adjust case:

```

\expandafter\glxtrsubsequentplfmt\expandafter\glslabel
\expandafter{\glinsert}%
}%
{%

```

Subsequent plural form, make first letter upper case:

```

\expandafter\Glxtrsubsequentplfmt\expandafter\glslabel
\expandafter{\glinsert}%
}%
{%

```

Subsequent plural form, all caps:

```

\expandafter\GLSxtrsubsequentplfmt\expandafter\glslabel
\expandafter{\glinsert}%
}%
}%
{%

```

Subsequent singular form

```

\glscapscase
{%

```

Subsequent singular form, don't adjust case:

```

\expandafter\glxtrsubsequentfmt\expandafter\glslabel
\expandafter{\glinsert}%
}%
{%

```

Subsequent singular form, make first letter upper case:

```

\expandafter\Glxtrsubsequentfmt\expandafter
\glslabel\expandafter{\glinsert}%
}%
{%

```

Subsequent singular form, all caps:

```

\expandafter\GLSxtrsubsequentfmt\expandafter
\glslabel\expandafter{\glinsert}%
}%

```

```
}%
}%
{%
```

First use:

```
\glsifplural
{%
```

First use plural form:

```
\glsapscase
{%
```

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

```
\glsapscase
{%
```

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}%
{%
\ifglsxtrininsertinside
\glsabbrvfont{\glsaccessshort{#1}\glsxtrgenentrytextfmt{#2}}%
\else
\glsabbrvfont{\glsaccessshort{#1}}\glsxtrgenentrytextfmt{#2}%
\fi
}%
}%
{%
\ifglsxtrininsertinside
\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}%
{%
\ifglsxtrininsertinside
\glsabbrvfont{\glsaccessshortpl{#1}\glsxtrgenentrytextfmt{#2}}%
\else
\glsabbrvfont{\glsaccessshortpl{#1}}\glsxtrgenentrytextfmt{#2}%
\fi
}%
}%
{%
\ifglsxtrininsertinside
\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}%
  {%
    \ifglxtrinsertinside
      \glsabbrvfont{\Glsaccessshort{#1}\glxtrgenentrytextfmt{#2}}%
    \else
      \glsabbrvfont{\Glsaccessshort{#1}}\glxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglxtrinsertinside
      \glsabbrvfont{\Glsaccessfmtshort{#2}{\glxtrgenentrytextfmt}{#1}}%
    \else
      \glsabbrvfont{\Glsaccessfmtshort{}}{\glxtrgenentrytextfmt}{#1}%
      \glxtrgenentrytextfmt{#2}%
    \fi
  }%
}
```

`\let\Glsxtrdefaultsubsequentfmt\Glsxtrsubsequentfmt`
`\glsmfuaddmap{\glxtrsubsequentfmt}{\Glsxtrsubsequentfmt}`

`\Glsxtrsubsequentplfmt` Subsequent use format (plural, first letter uppercase).

```
\newcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglxtrinsertinside
      \glsabbrvfont{\Glsaccessshortpl{#1}\glxtrgenentrytextfmt{#2}}%
    \else
      \glsabbrvfont{\Glsaccessshortpl{#1}}\glxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglxtrinsertinside
      \glsabbrvfont{\Glsaccessfmtshortpl{#2}{\glxtrgenentrytextfmt}{#1}}%
    \else
      \glsabbrvfont{\Glsaccessfmtshortpl{}}{\glxtrgenentrytextfmt}{#1}%
      \glxtrgenentrytextfmt{#2}%
    \fi
  }%
}
```

`\let\Glsxtrdefaultsubsequentplfmt\Glsxtrsubsequentplfmt`
`\glsmfuaddmap{\glxtrsubsequentplfmt}{\Glsxtrsubsequentplfmt}`

`\GLSxtrsubsequentfmt` Subsequent use format (singular, all caps).

```
\newcommand*{\GLSxtrsubsequentfmt}[2]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglxtrinsertinside
```

```

        \glsabbrvfont{\GLSaccessshort{#1}}%
        \glsuppercase{\glsxrigenentrytextfmt{#2}}}%
    \else
        \glsabbrvfont{\GLSaccessshort{#1}}%
        \glsuppercase{\glsxrigenentrytextfmt{#2}}}%
    \fi
}%
{%
    \ifglsxtrinsertinside
        \glsabbrvfont{\GLSaccessfmtshort{#2}{\glsxrigenentrytextfmt}{#1}}%
    \else
        \glsabbrvfont{\GLSaccessfmtshort{}{\glsxrigenentrytextfmt}{#1}}%
        \glsuppercase{\glsxrigenentrytextfmt{#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{\glsxrigenentrytextfmt{#2}}}%
        \else
            \glsabbrvfont{\GLSaccessshortpl{#1}}%
            \glsuppercase{\glsxrigenentrytextfmt{#2}}}%
        \fi
    }%
    {%
        \ifglsxtrinsertinside
            \glsabbrvfont{\GLSaccessfmtshortpl{#2}{\glsxrigenentrytextfmt}{#1}}%
        \else
            \glsabbrvfont{\GLSaccessfmtshortpl{}{\glsxrigenentrytextfmt}{#1}}%
            \glsuppercase{\glsxrigenentrytextfmt{#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}%
    }

```

`\do@ifcomplexstyle@allcaps` Do second argument if entry identified by first argument has a problem with all caps. Does nothing otherwise.

```
\newcommand*{\glxtr@do@ifcomplexstyle@allcaps}[2]{%
  \ifcsdef{@glxtr@has@complexstyle@#1}%
  {%
    \ifnum\csuse{@glxtr@has@complexstyle@#1}<1
    \else
    \ifnum\csuse{@glxtr@has@complexstyle@#1}<3
      #2%
    \fi
    \fi
  }%
}%
}
```

`\do@ifcomplexstyle@insert` Do second argument if entry identified by first argument has a problem with the insert argument. Does nothing otherwise.

```
\newcommand*{\glxtr@do@ifcomplexstyle@insert}[2]{%
  \ifcsdef{@glxtr@has@complexstyle@#1}%
  {%
    \ifnum\csuse{@glxtr@has@complexstyle@#1}<2
    \else
      #2%
    \fi
  }%
}%
}
```

`\sAbbrStyleTooComplexWarning`

```
\newcommand*{\GlossariesAbbrStyleTooComplexWarning}[2]{%
  \GlossariesExtraWarning{Abbreviation style used by ‘#1’ too complex #2}%
}
```

`\glxtr@check@complexstyle` The first argument is the label the second is the insert.

```
\newcommand*{\glxtr@check@complexstyle}[2]{%
  \ifx\glscapscase\@thirdofthree
  \glxtr@do@ifcomplexstyle@allcaps{#1}%
  {%
    \glxtrifwasfirstuse
  {%
    \glisifplural
  {%
    \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\GLSxtrshort.
    Use \string\GLS{#1} or \string\GLSxtrshort{#1} instead.
    Switching off all-caps%
}%
}%
}%
\let\glscapscase\@firstofthree
}%
\fi
\ifstrempy{#2}{}%
{%
    \glxtr@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\glxtrshort\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 abbreviation 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 `\glxtrsubsequentfmt` etc in case a style changes this.

```
\let\glxtrsubsequentfmt\glxtrdefaultsubsequentfmt
\let\glxtrsubsequentplfmt\glxtrdefaultsubsequentplfmt
\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{@glabbrv@dispstyle@setup@#1}{@glabbrv@dispstyle@setup@#2}%
  \csletcs{@glabbrv@dispstyle@fmts@#1}{@glabbrv@dispstyle@fmts@#2}%
}
```

```
@glxtr@deprecated@abbrstyle{<old-name>}{<new-name>}
```

`@glxtr@deprecated@abbrstyle`

Define a synonym for a deprecated abbreviation style.

```
\newcommand*{\@glxtr@deprecated@abbrstyle}[2]{%
  \csdef{@glabbrv@dispstyle@setup@#1}{%
    \GlsXtrWarnDeprecatedAbbrStyle{#1}{#2}%
    \csuse{@glabbrv@dispstyle@setup@#2}%
  }%
  \csletcs{@glabbrv@dispstyle@fmts@#1}{@glabbrv@dispstyle@fmts@#2}%
}
```

`sXtrWarnDeprecatedAbbrStyle` Generate warning for deprecated style use.

```
\newcommand*{\GlsXtrWarnDeprecatedAbbrStyle}[2]{%
  \GlossariesExtraWarning{Deprecated abbreviation style name ‘#1’,
    use ‘#2’ instead}%
}
```

`\GlsXtrUseAbbrStyleSetup`

```
\newcommand*{\GlsXtrUseAbbrStyleSetup}[1]{%
  \ifcsundef{@glabbrv@dispstyle@setup@#1}%
  {%
    \PackageError{glossaries-extra}%
      {Unknown abbreviation style definitions ‘#1’}{}%
  }%
  {%
    \csname @glabbrv@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` 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 too 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\glxtrifintoc\@firstoftwo
\glxtrmarkhook
\glxtrinmark
\@glxtr@org\@starttoc{#1}%
\glxtrnotinmark
\glxtrrestoremarkhook
\let\glxtrifintoc\@secondoftwo
}
```

If this causes a problem provide a simple way of switching back to the original definitions:

```
\glxtrRevertMarks
\newcommand*{\glxtrRevertMarks}{%
\let\markright\@glxtr@org\markright
\let\markboth\@glxtr@org\markboth
\let\@starttoc\@glxtr@org\@starttoc
}
```

```
\glxtrRevertTocMarks Just restores \@starttoc.
\newcommand*{\glxtrRevertTocMarks}{%
\let\@starttoc\@glxtr@org\@starttoc
}
```

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

```
\@glxtrinmark
\newrobustcmd*{\@glxtrinmark}{%
\let\glxtrifinmark\@firstoftwo
}
```

```
\@glxtrnotinmark
\newrobustcmd*{\@glxtrnotinmark}{%
\let\glxtrifinmark\@secondoftwo
}
```

```
\glxtrtitleorpdforheading
\newcommand*{\glxtrtitleorpdforheading}[3]{%
\glxtrifinmark{#3}{\glstexorpdfstring{#1}{#2}}}
```

This will require `\GetTitleStringSetup{expand}` to work.

```
\ifdef\GetTitleStringDisableCommands
{\GetTitleStringDisableCommands{\let\glxtrtitleorpdforheading\@thirdoftwo
\let\glxtrifinmark\@firstoftwo}}
{}}
```

`\glxtrmarkhook` Hook used in new definition of `\markboth` and `\markright` to make some changes to apply to the marks:

```
\newcommand*{\glxtrmarkhook}{%
```

Save current definitions:

```
\@glxtr@saveMakeUppercase
\let\@glxtr@org@glxtrtitleorpdforheading\glxtrtitleorpdforheading
\let\@glxtr@org@glxtrtitleshort\glxtrtitleshort
\let\@glxtr@org@glxtrtitleshortpl\glxtrtitleshortpl
\let\@glxtr@org@GLsxtrtitleshort\GLsxtrtitleshort
\let\@glxtr@org@GLsxtrtitleshortpl\GLsxtrtitleshortpl
\let\@glxtr@org@GLSxtrtitleshort\GLSxtrtitleshort
\let\@glxtr@org@GLSxtrtitleshortpl\GLSxtrtitleshortpl
\let\@glxtr@org@glxtrtitlename\glxtrtitlename
\let\@glxtr@org@GLsxtrtitlename\GLsxtrtitlename
\let\@glxtr@org@glxtrtitletext\glxtrtitletext
\let\@glxtr@org@GLsxtrtitletext\GLsxtrtitletext
\let\@glxtr@org@GLSxtrtitletext\GLSxtrtitletext
\let\@glxtr@org@glxtrtitleplural\glxtrtitleplural
\let\@glxtr@org@GLsxtrtitleplural\GLsxtrtitleplural
\let\@glxtr@org@GLSxtrtitleplural\GLSxtrtitleplural
\let\@glxtr@org@glxtrtitlefirst\glxtrtitlefirst
\let\@glxtr@org@GLsxtrtitlefirst\GLsxtrtitlefirst
\let\@glxtr@org@glxtrtitlefirstplural\glxtrtitlefirstplural
\let\@glxtr@org@GLsxtrtitlefirstplural\GLsxtrtitlefirstplural
\let\@glxtr@org@GLSxtrtitlefirstplural\GLSxtrtitlefirstplural
\let\@glxtr@org@glxtrtitlelong\glxtrtitlelong
\let\@glxtr@org@glxtrtitlelongpl\glxtrtitlelongpl
\let\@glxtr@org@GLsxtrtitlelong\GLsxtrtitlelong
\let\@glxtr@org@GLsxtrtitlelongpl\GLsxtrtitlelongpl
\let\@glxtr@org@glxtrtitlefull\glxtrtitlefull
\let\@glxtr@org@glxtrtitlefullpl\glxtrtitlefullpl
\let\@glxtr@org@GLsxtrtitlefull\GLsxtrtitlefull
\let\@glxtr@org@GLsxtrtitlefullpl\GLsxtrtitlefullpl
\let\@glxtr@org@GLSxtrtitlefull\GLSxtrtitlefull
\let\@glxtr@org@GLSxtrtitlefullpl\GLSxtrtitlefullpl
```

New definitions

```
\let\glxtrifinmark\@firstoftwo
\@glxtr@assignMakeUppercase
\let\glxtrtitleorpdforheading\@thirdofthree
\let\glxtrtitleshort\glxtrheadshort
\let\glxtrtitleshortpl\glxtrheadshortpl
\let\GLsxtrtitleshort\GLsxtrheadshort
\let\GLsxtrtitleshortpl\GLsxtrheadshortpl
\let\GLSxtrtitleshort\GLSxtrheadshort
\let\GLSxtrtitleshortpl\GLSxtrheadshortpl
\let\glxtrtitlename\glxtrheadname
\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
  \let\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\@glxstr@org@Glsxtrtitlefirst
\let\GLSxtrtitlefirst\@glxstr@org@GLSxtrtitlefirst
\let\glxstrtitlefirstplural\@glxstr@org@glxstrtitlefirstplural
\let\Glsxtrtitlefirstplural\@glxstr@org@Glsxtrtitlefirstplural
\let\GLSxtrtitlefirstplural\@glxstr@org@GLSxtrtitlefirstplural
\let\glxstrtitlelong\@glxstr@org@glxstrtitlelong
\let\glxstrtitlelongpl\@glxstr@org@glxstrtitlelongpl
\let\Glsxtrtitlelong\@glxstr@org@Glsxtrtitlelong
\let\GLSxtrtitlelongpl\@glxstr@org@GLSxtrtitlelongpl
\let\glxstrtitlefull\@glxstr@org@glxstrtitlefull
\let\glxstrtitlefullpl\@glxstr@org@glxstrtitlefullpl
\let\Glsxtrtitlefull\@glxstr@org@Glsxtrtitlefull
\let\GLSxtrtitlefullpl\@glxstr@org@GLSxtrtitlefullpl
}

```

Instead of using one document-wide conditional, use `headuc` attribute to determine whether or not to use the all upper case form.

`\glxstrtitleopts` Make it possible to change the default options within the title (but not the page header or table of contents).

```
\newcommand*{\glxstrtitleopts}{noindex,hyper=false}
```

```
\glxstr@title@field{<cs>}{<label>}
```

`\glxstr@title@field`

Used by all the `\glxstrtitle<field>` commands for consistency.

```

\newcommand*{\glxstr@title@field}[2]{%
  \expandafter#1\expandafter[\glxstrtitleopts]{#2}[]%
}

```

`\glxstrheadshort` Command used to display short form in the page header.

```

\newcommand*{\glxstrheadshort}[1]{%
  \protect\NoCaseChange
  {%
    \glxstrifheaduc{#1}%
    {%
      \GLSxtrshort[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glxstrshort[noindex,hyper=false]{#1}[]%
    }%
  }%
}

```

`\glxstrtitleshort` Command to display short form of abbreviation in section title.

```

\newrobustcmd*{\glxstrtitleshort}[1]{%
  \glxstr@title@field\glxstrshort{#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]{%
        \glxtr@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
        {%
            \glxtrifheaduc{#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]{%
        \glxtr@title@field\GLSxtrshortpl{#1}%
    }

\GLSxtrtitleshortpl Command to display plural short form of abbreviation in section title in all
upper case.
    \newrobustcmd*{\GLSxtrtitleshortpl}[1]{%
        \glxtr@title@field\GLSxtrshortpl{#1}%
    }

\glxtrheadname As above but for the name value.
    \newcommand*{\glxtrheadname}[1]{%
        \protect\NoCaseChange
        {%
            \glxtrifheaduc{#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
  {%

```

```

\glxstrifheaduc{#1}%
{%
  \GLStext[noindex,hyper=false]{#1}[]%
}%
{%
  \glstext[noindex,hyper=false]{#1}[]%
}%
}%
}

\glxstrtitletext Command to display text value in section title.
\newrobustcmd*{\glxstrtitletext}[1]{%
  \glxstr@title@field\glstext{#1}%
}

\GLxtrheadtext First letter converted to upper case
\newcommand*{\GLxtrheadtext}[1]{%
  \protect\NoCaseChange
  {%
    \glxstrifheaduc{#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]{%
  \glxstr@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]{%
  \glxstr@title@field\GLStext{#1}%
}

```

`\glxtrheadplural` As above but for the plural value.

```
\newcommand*{\glxtrheadplural}[1]{%
\protect\NoCaseChange
{%
\glxtrifheaduc{#1}%
{%
\GLSplural[noindex,hyper=false]{#1}[]%
}%
}%
\glsplural[noindex,hyper=false]{#1}[]%
}%
}
```

`\glxtrtitleplural` Command to display plural value in section title.

```
\newrobustcmd*{\glxtrtitleplural}[1]{%
\glxtr@title@field\glsplural{#1}%
}
```

`\Glsxtrheadplural` Convert first letter to upper case.

```
\newcommand*{\Glsxtrheadplural}[1]{%
\protect\NoCaseChange
{%
\glxtrifheaduc{#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]{%
\glxtr@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}%
}
```

```

        \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]{%
  \glxtr@title@field\GLSfirst{#1}%
}

```

`\glxtrheadfirstplural` As above but for the firstplural value.

```

\newcommand*{\glxtrheadfirstplural}[1]{%
  \protect\NoCaseChange
  {%
    \glxtrifheaduc{#1}%
    {%
      \GLSfirstplural[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glSfirstplural[noindex,hyper=false]{#1}[]%
    }%
  }%
}

```

`\glxtrtitlefirstplural` Command to display firstplural value in section title.

```

\newrobustcmd*{\glxtrtitlefirstplural}[1]{%
  \glxtr@title@field\glSfirstplural{#1}%
}

```

`\Glsxtrheadfirstplural` First letter converted to upper case

```

\newcommand*{\Glsxtrheadfirstplural}[1]{%
  \protect\NoCaseChange
  {%
    \glxtrifheaduc{#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]{%
  \glxtr@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]{%
  \glxtr@title@field\GLSfirstplural{#1}%
}
```

`\glxtrheadlong` Command used to display long form in the page header.

```
\newcommand*{\glxtrheadlong}[1]{%
  \protect\NoCaseChange
  {%
    \glxtrifheaduc{#1}%
    {%
      \GLSxtrlong[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glxtrlong[noindex,hyper=false]{#1}[]%
    }%
  }%
}
```

`\glxtrtitlelong` Command to display long form of abbreviation in section title.

```
\newrobustcmd*{\glxtrtitlelong}[1]{%
  \glxtr@title@field\glxtrlong{#1}%
}
```

`\glxtrheadlongpl` 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*{\glxtrheadlongpl}[1]{%
  \protect\NoCaseChange
  {%
    \glxtrifheaduc{#1}%
    {%
      \GLSxtrlongpl[noindex,hyper=false]{#1}[]%
    }%
    {%
      \glxtrlongpl[noindex,hyper=false]{#1}[]%
    }%
  }%
}
```

`\glxtrtitlelongpl` Command to display plural long form of abbreviation in section title.

```
\newrobustcmd*{\glxtrtitlelongpl}[1]{%
  \glxtr@title@field\glxtrlongpl{#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]{%
  \glxtr@title@field\Glsxtrlongpl{#1}%
}
```

`\GLSxtrtitlelongpl` Command to display plural long form of abbreviation in section title in all upper case.

```
\newrobustcmd*{\GLSxtrtitlelongpl}[1]{%
  \glxtr@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}[]%
  }%
}
```

`\glxtrheadfull` Command used to display full form in the page header.

```
\newcommand*{\glxtrheadfull}[1]{%
  \protect\NoCaseChange
  {%
    \glxtrifheaduc{#1}%
  {%
    \GLSxtrfull[noindex,hyper=false]{#1}[]%
  }%
  {%
    \glxtrfull[noindex,hyper=false]{#1}[]%
  }%
}
```

`\glxtrtitlefull` Command to display full form of abbreviation in section title.

```
\newrobustcmd*{\glxtrtitlefull}[1]{%
  \glxtr@title@field\glxtrfull{#1}%
}
```

`\glxtrheadfullpl` 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*{\glxtrheadfullpl}[1]{%
  \protect\NoCaseChange
  {%
    \glxtrifheaduc{#1}%
  {%
    \GLSxtrfullpl[noindex,hyper=false]{#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}[]%
}%
{%
\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
{%
\glxstrifheaduc{#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]{%
\glxstr@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]{%
\glxstr@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
{\glxstrtitleshort{#1}}%
{\glsentryshort{#1}}%
}
```

Similarly for the plural version.

`\glsfmtshortpl`

```
\newcommand*{\glsfmtshortpl}[1]{%
\glstexorpdfstring
{\glxstrtitleshortpl{#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{\glstryname{#1}}}%
    }
    \glsmfuaddmap{\glsfmtname}{\Glsfmdname}

\GLSfmdname All upper case.
    \newcommand*{\GLSfmdname}[1]{%
        \glstexorpdfstring
        {\GLSxtrtitle{#1}}%
        {\GLSxtrusefield{#1}{name}}%
    }
    \glsmfublocker{\GLSfmdname}

\glsfmdtext As above but for the text value.
    \newcommand*{\glsfmdtext}[1]{%
        \glstexorpdfstring
        {\glstxttitle{#1}}%
        {\glstrytext{#1}}%
    }

\Glsfmdtext First letter converted to upper case.
    \newcommand*{\Glsfmdtext}[1]{%
        \glstexorpdfstring
        {\glstxttitle{#1}}%
        {\MFUsentencecase{\glstrytext{#1}}}%
    }
    \glsmfuaddmap{\glsfmdtext}{\Glsfmdtext}

\GLSfmdtext All upper case.
    \newcommand*{\GLSfmdtext}[1]{%
        \glstexorpdfstring
        {\GLSxtrtitle{#1}}%
        {\GLSxtrusefield{#1}{text}}%
    }
    \glsmfublocker{\GLSfmdtext}

\glsfmdplural As above but for the plural value.
    \newcommand*{\glsfmdplural}[1]{%
        \glstexorpdfstring
        {\glstxttitleplural{#1}}%
        {\glstryplural{#1}}%
    }

\Glsfmdplural First letter converted to upper case.
    \newcommand*{\Glsfmdplural}[1]{%
        \glstexorpdfstring
        {\glstxttitleplural{#1}}%
        {\MFUsentencecase{\glstryplural{#1}}}%
    }
    \glsmfuaddmap{\glsfmdplural}{\Glsfmdplural}

```

`\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}}%
        {\MFUsentencecase{\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}}%
        {\MFUsentencecase{\glsentrylongpl{#1}}}%
    }
    \glsmfuaddmap{\glsfmtlongpl}{\Glsfmtlongpl}

\GLSfmtlongpl All upper case.
    \newcommand*{\GLSfmtlongpl}[1]{%
        \glstexorpdfstring
        {\GLSxtrtitlelongpl{#1}}%
        {\GLSxtrusefield{#1}{longpl}}%
    }
    \glsmfublocker{\GLSfmtlongpl}

```

```

\glspdffmtfull Can't use \glxtrinlinefullformat 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]{\glentrylong{#1} (\glentryshort{#1})}%

\glspdffmtfullpl Likewise for plural.
    \newcommand*{\glspdffmtfullpl}[1]{\glentrylongpl{#1} (\glentryshortpl{#1})}%

\glsfmtfull In-line full format.
    \newcommand*{\glsfmtfull}[1]{%
        \glstexorpdfstring
        {\glxtrtitlefull{#1}}%
        {\glspdffmtfull{#1}}%
    }

\Glsfmtfull First letter converted to upper case.
    \newcommand*{\Glsfmtfull}[1]{%
        \glstexorpdfstring
        {\Glsxtrtitlefull{#1}}%
        {\MFUsentencecase{\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
        {\glxtrtitlefullpl{#1}}%
        {\glspdffmtfullpl{#1}}%
    }

\Glsfmtfullpl First letter converted to upper case.
    \newcommand*{\Glsfmtfullpl}[1]{%
        \glstexorpdfstring
        {\Glsxtrtitlefullpl{#1}}%
        {\MFUsentencecase{\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{\glspdfmtfullpl{#1}{}}}%
}
\ExplSyntaxOff
\glsmfublocker{\GLSfmtfullpl}
```

1.9 Prefixes

Provide support for glossaries-prefix.

`\pglsprefix`

`\pglsprefix{<entry-label>}{<prefix-field>}`

A shortcut way of inserting the prefix and separator if they are required. If this needs to be redefined, use `\ifglsfldvoid` for an expandable test.

```
\newcommand{\pglsprefix}[2]{%
  \ifcempty{glo@\glsdetoklabel{#1}@#2}{}%
  {\csuse{glo@\glsdetoklabel{#1}@#2}\glsprefixsep}%
}
```

`\Pglsprefix`

`\Pglsprefix{<entry-label>}{<prefix-field>}`

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`

`\PGLSprefix{<entry-label>}{<prefix-field>}`

As `\pglsprefix` but all caps.

```
\newcommand{\PGLSprefix}[2]{%
  \ifcempty{glo@\glsdetoklabel{#1}@#2}{}%
  {\glssupercase{\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}%
  \@glxtrshort{#1}{#2}[#3]%
}

```

\Pglxtrshort Sentence case.

```

\newrobustcmd*{\Pglxtrshort}{\@glshyp@opt\ns@Pglxtrshort}
\newcommand*{\ns@Pglxtrshort}[2][]{%
  \new@ifnextchar[{\@Pglxtrshort{#1}{#2}}{\@Pglxtrshort{#1}{#2}[]}%
}
\def\@Pglxtrshort#1#2[#3]{%
  \ifglshasprefix{#2}%
  {%
    \Pglsprefix{#2}{prefix}%
    \@glxtrshort{#1}{#2}[#3]%
  }%
  {\@Glsxtrshort{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglsxtrshort}{\Pglxtrshort}

```

\PGLSxtrshort All-caps is also fairly simple.

```

\newrobustcmd*{\PGLSxtrshort}{\@glshyp@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}{\@glshyp@opt\ns@pglsxtrshortpl}
\newcommand*{\ns@pglsxtrshortpl}[2][]{%
  \new@ifnextchar[{\@pglsxtrshortpl{#1}{#2}}{\@pglsxtrshortpl{#1}{#2}[]}%
}
\def\@pglsxtrshortpl#1#2[#3]{%
  \pglsprefix{#2}{prefixplural}%
  \@glxtrshortpl{#1}{#2}[#3]%
}

```

\Pglxtrshortpl

```

\newrobustcmd*{\Pglxtrshortpl}{\@glshyp@opt\ns@Pglxtrshortpl}
\newcommand*{\ns@Pglxtrshortpl}[2][]{%
  \new@ifnextchar[{\@Pglxtrshortpl{#1}{#2}}{\@Pglxtrshortpl{#1}{#2}[]}%
}

```

```

}
\def\@Pglstrshortpl#1#2[#3]{%
  \ifglshasprefixplural{#2}%
  {%
    \Pglsprefix{#2}{prefixplural}%
    \@glstrshortpl{#1}{#2}[#3]%
  }%
  {\@Glsstrshortpl{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglstrshortpl}{\Pglstrshortpl}

```

\PGLSxtrshortpl All-caps is also fairly simple.

```

\newrobustcmd*{\PGLSxtrshortpl}{\@glshyp@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}{\@glshyp@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]%
}

```

\Pglxtrlong Sentence case.

```

\newrobustcmd*{\Pglxtrlong}{\@glshyp@opt\ns@Pglxtrlong}
\newcommand*{\ns@Pglxtrlong}[2][]{%
  \new@ifnextchar[{\@Pglxtrlong{#1}{#2}}{\@Pglxtrlong{#1}{#2}[]}%
}
\def\@Pglxtrlong#1#2[#3]{%
  \ifglshasprefixfirst{#2}%
  {%
    \Pglsprefix{#2}{prefixfirst}%
    \@glxtrlong{#1}{#2}[#3]%
  }%
  {\@Glsxtrlong{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglxtrlong}{\Pglxtrlong}

```

\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}%
  \@glxtrlongpl{#1}{#2}[#3]%
}
}
```

\Pglxtrlongpl

```
\newrobustcmd*{\Pglxtrlongpl}{\@gls@hyp@opt\ns@Pglxtrlongpl}
\newcommand*{\ns@Pglxtrlongpl}[2][]{%
  \new@ifnextchar[{\@Pglxtrlongpl{#1}{#2}}{\@Pglxtrlongpl{#1}{#2}[]}%
}
\def\@Pglxtrlongpl#1#2[#3]{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Pglsprefix{#2}{prefixfirstplural}%
    \@glxtrlongpl{#1}{#2}[#3]%
  }%
  {\@GLSxtrlongpl{#1}{#2}[#3]}%
}
\glsmfuaddmap{\pglsxtrlongpl}{\Pglxtrlongpl}
```

\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 \glsmfshort etc).

```

\pglsfmtshort
\newcommand*{\pglsfmtshort}[1]{%
  \pglsprefix{#1}{prefix}%
  \glsfmtshort{#1}%
}

\Pglsfmtshort
\newcommand*{\Pglsfmtshort}[1]{%
  \glstexpdfstring
  {\Pglsxtrtitleshort{#1}}%
  {\MFUsentencecase{%
    \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]{%
  \glstexpdfstring
  {\Pglsxtrtitleshortpl{#1}}%
  {\MFUsentencecase
    {%
      \pglsprefix{#1}{prefixplural}%
      \glsentryshortpl{#1}%
    }%
  }%
}
\glsmfuaddmap{\pglsfmtshortpl}{\Pglsfmtshortpl}

```

```

\Glsxtrtitleshortpl
\newrobustcmd*{\Glsxtrtitleshortpl}[1]{%
  \glsxtr@title@field\Glsxtrshortpl{#1}%
}

\GLSfmtshortpl
\newcommand*{\GLSfmtshortpl}[1]{%
  \GLSPrefix{#1}{prefixplural}%
  \GLSfmtshortpl{#1}%
}
\glsmfublocker{\GLSfmtshortpl}

\pglsfmtlong
\newcommand*{\pglsfmtlong}[1]{%
  \pglsprefix{#1}{prefixfirst}%
  \glsfmtlong{#1}%
}

\Pglsfmtlong
\newcommand*{\Pglsfmtlong}[1]{%
  \glstexorpdfstring
  {\Glsxtrtitlelong{#1}}%
  {\MFUsentencecase%
    \pglsprefix{#1}{prefixfirst}%
    \glstentrylong{#1}}%
  }%
}
\glsmfuaddmap{\pglsfmtlong}{\Pglsfmtlong}

\Glsxtrtitlelong
\newrobustcmd*{\Glsxtrtitlelong}[1]{%
  \glsxtr@title@field\Glsxtrlong{#1}%
}

\GLSfmtlong
\newcommand*{\GLSfmtlong}[1]{%
  \GLSPrefix{#1}{prefixfirst}%
  \GLSfmtlong{#1}%
}
\glsmfublocker{\GLSfmtlong}

\pglsfmtlongpl
\newcommand*{\pglsfmtlongpl}[1]{%
  \pglsprefix{#1}{prefixfirstplural}%
  \glsfmtlongpl{#1}%
}

```

```

\Glsfmtlongpl
\newcommand*\Glsfmtlongpl[1]{%
  \glstexorpdfstring
  {\Pglstrtitlelongpl{#1}}%
  {\MFUsentencecase
   {%
    \pglsprefix{#1}{prefixfirstplural}%
    \glentrylongpl{#1}%
   }}%
}%
\glsmfuaddmap{\pglsfmtlongpl}{\Glsfmtlongpl}

\Pglstrtitlelongpl
\newrobustcmd*\Pglstrtitlelongpl[1]{%
  \glstr@title@field\Pglstrlongpl{#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 `\cgl` 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{clostridium}
%\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 con-
tent 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 \mgl.
\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}

\mgl@disable@mglsopts
\newcommand*{\mgl@disable@mglsopts}{%
\let\@gls@combined@mglsopts@do\@gls@combined@mglsopts@do@not
}

\mgl@enable@mglsopts
\newcommand*{\mgl@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@glxtrcombined@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@glxtrcombined@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@glxtrcombined@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@glxtrcombined@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}}

```

`\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

```

```

\multiglossaryentry[\langle options \rangle]{\langle multi-label \rangle}[\langle main label \rangle]
{\langle label
list \rangle}

```

`\multiglossaryentry`

Defines the label *multi-label* that can be used in `\mgl`s.

```
\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}%
{}}%
}%
{%
\@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
    {\@glstr@docdefs@multilist}{\expandonce\@gl@combined@current@label}%
  }%
}%
\@def@multi@glossaryentry@do{#1}{#2}%
}

```

`\@multi@glossary@doifexists`

```
\newcommand*{\@multi@glossary@doifexists}{\gl@doifexists}
```

```

\providemultiglossaryentry[\langle options \rangle]{\langle multi-label \rangle}[\langle main
label \rangle]{\langle label
list \rangle}

```

`\providemultiglossaryentry`

Defines a multi-entry unless it has already been defined.

```

\newrobustcmd{\providemultiglossaryentry}[2] [] {%
  \def\@gl@combined@current@opts{#1}%
  \def\@gl@combined@current@label{#2}%
  \ifcsdef{\@gl@combined@\@gl@combined@current@label @main}%
  {\def\@multi@glossentry@donext{\@provide@multi@glossaryentry@noop}}%
  {%
    \ifnum\@glstr@docdefval=1\relax
    \def\@multi@glossentry@donext{\@def@multi@glossaryentry}%
    \else
    \def\@multi@glossentry@donext{\@multi@glossaryentry}%
    \fi
  }%
  \@multi@glossentry@donext
}

```

`\@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\@gl@combined@current@main{#1}%

```

Fully expand list.

```
\protected@edef\@gl@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}%
{%
\ea\pto\@multi@glossaryentry@list{,\@gls@combined@current@label}%
}%
\else
\ifdefempty\@multi@glossaryentry@list
{\global\let\@multi@glossaryentry@list\@gls@combined@current@label}%
{%
\xappto\@multi@glossaryentry@list{,\@gls@combined@current@label}%
}%
\fi
}
```

```
\@glsxtr@multientry{<options>}{<multilabel>}{<main>}{<list>}
```

\@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}%
  }%

\mglsSetName{<multi-label>}{<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
}%
}

```

```
\mglSetOptions{<multi-label>}{<new options>}
```

\mglSetOptions

Allow the options to be changed (local). No expansion is applied.

```

\newrobustcmd{\mglSetOptions}[2]{%
  \ifcsundef{\@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label ‘#1’ not defined}{}}%
  {%
    \csdef{\@gls@combined@#1@options}{#2}%
  }%
}

```

```
}%
}
```

```
\mglAddOptions{<multi-label>}{<extra options>}
```

\mglAddOptions

Allow the options to be changed (local). No expansion is applied.

```
\newrobustcmd{\mglAddOptions}[2]{%
  \ifcsundef{@gls@combined@#1@main}%
  {\PackageError{glossaries-extra}{Multi-entry label ‘#1’ not defined}{}}%
  {%
    \ifcempty{@gls@combined@#1@options}%
    {\csdef{@gls@combined@#1@options}{#2}}%
    {\csappto{@gls@combined@#1@options}{, #2}}%
  }%
}
```

Options for \mgl:

\@mgl@all Options to apply to all elements.

```
\newcommand*{\@mgl@all}{}
\define@key{mgl}{all}{\renewcommand*{\@mgl@all}{#1}}
```

\@mgl@main Options to apply to the main element only.

```
\newcommand*{\@mgl@main}{}
\define@key{mgl}{main}{\renewcommand*{\@mgl@main}{#1}}
```

\@mgl@others Options to apply to the other (no main) elements.

```
\newcommand*{\@mgl@others}{}
\define@key{mgl}{others}{\renewcommand*{\@mgl@others}{#1}}
```

\@mgl@setup Options to apply to \multiglossaryentrysetup.

```
\newcommand*{\@mgl@setup}{}
\define@key{mgl}{setup}{%
  \@mgl@setup@do{\renewcommand*{\@mgl@setup}{#1}}%
}
```

\@mgl@setup@do

```
\newcommand*{\@mgl@setup@do}[1]{#1}
```

\@mgl@setup@do@not

```
\newcommand*{\@mgl@setup@do@not}[1]{%
  \PackageError{glossaries-extra}{‘setup’ key not permitted inside
  ‘mglsopts’ value}{}}%
}
```

\mgl@disable@setup

```
\newcommand*{\mgl@disable@setup}{%
  \let\@mgl@setup@do\@mgl@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{\glslinkprefix#1}{#2}%
        \fi
    }

\@mgls@hyperlink
    \let\@mgls@hyperlink\@@mgls@hyperlink

\mglsforelements
    \mglsforelements{\langle multi-label \rangle}{\langle cs \rangle}{\langle body \rangle}

    \newcommand*\mglsforelements}[3]{%
        \expandafter\@for\expandafter#2\expandafter:\expandafter
            =csname @gls@combined@#1@list@endcsname\do{#3}%
    }

\mglsforotherelements
    \mglsforotherelements{\langle multi-label \rangle}{\langle cs \rangle}{\langle body \rangle}

```

```

\newcommand*{\mglsotherelements}[3]{%
  \expandafter\@for\expandafter#2\expandafter:\expandafter
    =\csname @gls@combined@#1@list\endcsname\do
    {\expandafter\ifdefequal\csname @gls@combined@#1@main\endcsname{#2}-{#3}}%
}

\mglunsetothers

\newcommand*{\mglunsetothers}[1]{%
  \mglsotherelements{#1}{\@gls@tmp}{\glsunset{\@gls@tmp}}%
}

\mgllocalunsetothers

\newcommand*{\mgllocalunsetothers}[1]{%
  \mglsotherelements{#1}{\@gls@tmp}{\glslocalunset{\@gls@tmp}}%
}

\mglselementreset

\newcommand*{\mglselementreset}[1]{%
  \ifKV@mgl@presetlocal
    \glslocalreset{#1}%
  \else
    \glsreset{#1}%
  \fi
}

\mglselementunset

\newcommand*{\mglselementunset}[1]{%
  \ifKV@mgl@presetlocal
    \glslocalunset{#1}%
  \else
    \glsunset{#1}%
  \fi
}

\@mgl@resetall

\newcommand*{\@mgl@resetall}{}
\define@choicekey{mgl}{resetall}%
[\@mgl@resetall@val\@mgl@resetall@nr]{false,true}[true]%
{%
  \ifcase\@mgl@resetall@nr\relax
    \renewcommand*{\@mgl@resetall}{}%
  \or
    \renewcommand*{\@mgl@resetall}{%
      \@for\@gls@resetlabel:=\mglcurrentlist\do{\mglselementreset\@gls@resetlabel}}%
    \renewcommand*{\@mgl@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{mgl}{unsetmain}
[ \@mgl{unsetmain@val\@mgl{unsetmain@nr}{false,true}[true]}%
{%
\ifcase\@mgl{unsetmain@nr}\relax
\renewcommand*{\@mgl{unsetmain}}{}%
\or
\renewcommand*{\@mgl{unsetmain}}{\mglselementunset\mglcurrentmainlabel}%
\renewcommand*{\@mgl{resetmain}}{}%
\fi
}

```

\@mgl{unsetothers}

```

\newcommand*{\@mgl{unsetothers}}{}
\define@choicekey{mgl}{unsetothers}
[ \@mgl{unsetothers@val\@mgl{unsetothers@nr}{false,true}[true]}%
{%
\ifcase\@mgl{unsetothers@nr}\relax
\renewcommand*{\@mgl{unsetothers}}{}%
\or
\renewcommand*{\@mgl{unsetothers}}{%
\@for\@gl{unsetlabel:=\mglcurrentlist\do{%
\ifx\@gl{unsetlabel\mglcurrentmainlabel
\else
\mglselementunset\@gl{unsetlabel
\fi
}%
}%
}%
\renewcommand*{\@mgl{resetothers}}{}%
\fi
}

```

\glxtr@setup@docurrent Set up the commands to determine whether or not to do the current element.

```
\newcommand{\glxtr@setup@docurrent}{%
```

\mglcurrentlabel expands to the label of the current element. Should this element be skipped?

```
\ifx\mglcurrentlabel\mglcurrentmainlabel
```

Main element. Should it be skipped?

```

\mgl{isfirstuse}
{%
\ifKV@glxtrcombined@firstskipmain
\let\@mgl{do@current@element}\@gobble
\else
\let\@mgl{do@current@element}\@firstofone
\fi
}%
{%
\ifKV@glxtrcombined@usedskipmain
\let\@mgl{do@current@element}\@gobble

```

```

        \else
        \let\@mgls@do@current@element\@firstofone
        \fi
    }%
\else
Other element. Should it be skipped?
\mglsisfirstuse
{%
    \ifKV@glstrcombined@firstskipothers
    \let\@mgls@do@current@element\@gobble
    \else
    \let\@mgls@do@current@element\@firstofone
    \fi
}%
{%
    \ifKV@glstrcombined@usedskipothers
    \let\@mgls@do@current@element\@gobble
    \else
    \let\@mgls@do@current@element\@firstofone
    \fi
}%
\fi
}

```

`\glstr@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*{\glstr@mgls@checklastelement}[2]{%
    \ifbool{KV@glstrcombined@#1skipmain}%
    {%
        \ifbool{KV@glstrcombined@#1skipothers}%
        {%

```

This condition has already been checked for.

```

    }%
    {%

```

Main skipped. The last item will be the last other element.

```

        \ifnum\mglsindex=\glstrmultilastotherindex{#2}\relax
        \let\mglsiflast\@firstoftwo
        \else
        \let\mglsiflast\@secondoftwo
        \fi
    }%
}%
{%

```

Main not skipped.

```

    \ifbool{KV@glstrcombined@#1skipothers}%
    {%

```


Others skipped. The main element is the only item.

```
\ifnum\mglsselementindex=\glxstrmultimainindex{#2}\relax
\let\mglsiflast\@firstoftwo
\else
\let\mglsiflast\@secondoftwo
\fi
}%
{%
```

None skipped. This isn't the last element.

```
\let\mglsiflast\@secondoftwo
}%
}%
}
```

`\glxstrmglsWarnAllSkipped` 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{\glxstrmglsWarnAllSkipped}[3]{%
\GlossariesExtraWarning{#1}%
#3{#2}%
}
```

`\glxstr@mgls@applyopts`

```
\newcommand*{\glxstr@mgls@applyopts}[1]{%
\edef\@mgls@doptions{\noexpand\setkeys*{mgls}{\expandonce#1}}%
\@mgls@doptions}
```

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@doptions{%
\noexpand\setkeys*{glxstrcombinedpreopts}{\expandonce\@mgls@setup}}%
\mgls@disable@mgls@opts
\@mgls@doptions
\mgls@enable@mgls@opts}
```

Save remaining setup options.

```
\ifx\@mgls@setuptoptions\empty
\let\@mgls@setuptoptions\XKV@rm
\else
\eappto\@mgls@setuptoptions{\expandonce\XKV@rm}%
\fi
}%
```

Apply gls unset/reset options.

```
\@mgls@resetall
```

```

\@mgl@unsetall
\@mgl@resetmain
\@mgl@unsetmain
\@mgl@resetothers
\@mgl@unsetothers

```

Disable.

```

\let\@mgl@resetall\@empty
\let\@mgl@resetmain\@empty
\let\@mgl@resetothers\@empty
\let\@mgl@unsetall\@empty
\let\@mgl@unsetmain\@empty
\let\@mgl@unsetothers\@empty

```

First use flags.

```

\ifmgl@used\mgl@currentmultilabel
{\let\mgl@isfirstuse\@secondoftwo}%
{\let\mgl@isfirstuse\@firstoftwo}%
}

```

\@firstofthree

```

\providecommand{\@firstofthree}[3]{#1}

```

\@secondofthree

```

\providecommand{\@secondofthree}[3]{#2}

```

\@thirdofthree

```

\providecommand{\@thirdofthree}[3]{#3}

```

The main internal command for referencing multi-entries:

```

\glxtr@mgl@inner{<options>}{<label>}{<insert>}{<first
cs>}{<not first cs>}{<main first cs>}{<main other cs>}

```

\glxtr@mgl@inner

```

\newcommand*\glxtr@mgl@inner[7]{%
\let\mgl@lastmainlabel\@empty
\let\mgl@siflastmainwasfirstuse\@firstoftwo
\let\mgl@siflastmainwasplural\@secondoftwo
\let\mgl@siflastmaincapscase\@firstofthree
\let\mgl@siflastmainskipped\@firstoftwo
\bgroup
\ifcsundef{@gls@combined@#2@main}%
{%
\glxtrundefaction{Multi entry ‘#2’ hasn’t been defined}%
{You need to define ‘#2’ with \string\multiglossaryentry}%
\gdef\@mgl@post@hookdefs{%
\protected@edef\mgl@lastmultilabel{#2}%
\let\mgl@wasfirstuse\@firstoftwo
\let\mgl@lastcategory\@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\glstrifwasfirstuse\@firstoftwo
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree

```

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@doptions{%
  \noexpand\setkeys*{glstrcombinedpreopts}{\expandonce\mglscurrentoptions}}%
\@mgls@doptions

```

Save remaining setup options.

```

\let\@mgls@setuptoptions\XKV@rm

```

Apply \mgls options.

```

\mgls@disable@setup
\ifdefvoid{@gls@combined@mglsopts}
{}%
{\glstr@mgls@applyopts{@gls@combined@mglsopts}%
\mgls@enable@setup

```

Apply options provided in #1.

```

\ifstrempy{#1}{-}{\def\@mgls@options{#1}\glstr@mgls@applyopts\@mgls@options}%

```

Check for attribute settings.

```

\ifx{@gls@combined@category}\empty

```

No category

```

\else

```

Attribute options:

```

\glshascategoryattribute{\@gls@combined@category}{multioptions}%

```

```
{%
\letcs\@mgl@attroptions{\@glxtr@categoryattr@@\@gl@combined@category
@multioptions}%
```

Only obtain pre-option keys:

```
\let\@gl@combined@mglsopts\@empty
\edef\@mgl@doptions{%
\noexpand\setkeys*{\glxtrcombinedpreopts}{\expandonce\@mgl@attroptions}}%
\@mgl@doptions
```

Append remaining options:

```
\eappto\@mgl@setoptions{,\expandonce\XKV@rm}%
\ifx\@gl@combined@mglsopts\@empty
\else
```

mgl options found:

```
\let\@mgl@setup\@empty
\mgl@disable@setup
\glxtr@mgl@applyopts\@gl@combined@mglsopts
\mgl@enable@setup
\fi
}%
{}%
\fi
```

Apply setup options.

```
\edef\@mgl@doptions{%
\noexpand\setkeys{\glxtrcombined}{\expandonce\@mgl@setoptions}}%
\@mgl@doptions
```

Provide local user-level access to category:

```
\let\mglcurrentcategory\@gl@combined@category
```

Should the entire content be a hyperlink?

```
\ifnum\@gl@combined@hyper=1\relax
\def\@mgl@combinedlink{\@mgl@hyperlink{\mglcurrentmainlabel}}%
\else
\def\@mgl@combinedlink{\@firstofone}%
\fi
```

Entire content encapsulator.

```
\def\@gl@combined@encapsulator##1{%
\@mgl@combinedlink{\csuse{\@gl@combined@textformat}{##1}}}%
```

Initialise.

```
\let\@mgl@do@current@element\@firstofone
```

Check if all elements are being skipped.

```
\mglsisfirstuse
{%
\ifKV@glxtrcombined@firstskipmain
\ifKV@glxtrcombined@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{%
  \glstrmglWarnAllSkipped{All elements skipped for
    first use of multi-entry '#2'#{#3}%
    {\@gls@org@combined@encapsulator}%
  }%
  \let\@mgl@do@current@element\@gobble
\fi
\fi
}%
{%
  \ifKV@glstrcombined@usedskipmain
  \ifKV@glstrcombined@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{%
  \glstrmglWarnAllSkipped{All elements skipped for
    subsequent use of multi-entry '#2'#{#3}%
    {\@gls@org@combined@encapsulator}%
  }%
  \let\@mgl@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\@mgl@post@hookdefs{%
  \noexpand\def\noexpand\mglslastmultilabel{\expandonce\mglscurrentmultilabel}%
  \noexpand\def\noexpand\mglslastcategory{\mglscurrentcategory}%
}%
\ifx\@mgl@do@current@element\@gobble
  \gappto\@mgl@post@hookdefs{%
    \let\mglslastelementskipped\@firstoftwo
    \let\mglslastelementlabel\@empty
    \let\mglslastelementwasfirstuse\@firstoftwo
    \let\mglslastelementwasplural\@secondoftwo
    \let\mglslastelementcapscase\@firstofthree
  }%

```

```

\fi
\mglsisfirstuse
{%
  \gappto\@mglspost@hookdefs{\let\mglswasfirstuse\@firstoftwo}%
Determine if the multi-entry post-link hook should be applied.
  \ifcase\@gls@combined@mpostlink@nr\relax
mpostlink=false.
  \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\relax}%
  \or
mpostlink=true.
  \ifcase\@gls@combined@mpostlinkelement@nr\relax
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglslastelementpostlinkhook}%
  \or
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglslastmainpostlinkhook}%
  \or
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglscustompostlinkhook}%
  \fi
  \or
mpostlink=firstonly.
  \ifcase\@gls@combined@mpostlinkelement@nr\relax
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglslastelementpostlinkhook}%
  \or
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglslastmainpostlinkhook}%
  \or
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglscustompostlinkhook}%
  \fi
  \or
mpostlink=usedonly.
  \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\relax}%
  \fi
}%
{%
  \gappto\@mglspost@hookdefs{\let\mglswasfirstuse\@secondoftwo}%
Determine if the multi-entry post-link hook should be applied.
  \ifcase\@gls@combined@mpostlink@nr\relax
mpostlink=false.
  \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\relax}%
  \or
mpostlink=true.
  \ifcase\@gls@combined@mpostlinkelement@nr\relax
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglslastelementpostlinkhook}%
  \or
    \gappto\@mglspost@hookdefs{\let\mgl@do@postlinkhook\mglslastmainpostlinkhook}%
  \or
    \gappto\@mglspost@hookdefs{\let\mgl@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
\glxtr@setup@docurrent
Is this the last element?
\ifx\@xfor@nextelement\@nnil
\let\mglsiflast\@firstoftwo
\else
\let\mglsiflast\@secondoftwo
Are any elements being skipped?
\mglsisfirstuse
{%
\glxtr@mgls@checklastelement{first}{#2}%
}%
{%
\glxtr@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\mglorg@postlinkhook
\or
postlinks=notlast
    \mgliflast
    {%
        \let\glspostlinkhook\relax
    }%
    {%
        \let\glspostlinkhook\mglorg@postlinkhook
    }%
\or
postlinks=mainnotlast
    \ifx\mglcurrentlabel\mglcurrentmainlabel
        \mgliflast
        {%
            \let\glspostlinkhook\relax
        }%
        {%
            \let\glspostlinkhook\mglorg@postlinkhook
        }%
    \else
        \let\glspostlinkhook\relax
    \fi
\or
postlinks=mainonly
    \ifx\mglcurrentlabel\mglcurrentmainlabel
        \let\glspostlinkhook\mglorg@postlinkhook
    \else
        \let\glspostlinkhook\relax
    \fi
\or
postlinks=othernotlast
    \ifx\mglcurrentlabel\mglcurrentmainlabel
        \let\glspostlinkhook\relax
    \else
        \mgliflast
        {%
            \let\glspostlinkhook\relax
        }%
        {%
            \let\glspostlinkhook\mglorg@postlinkhook
        }%
    \fi
\or

```



```

\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

```

```

\appto\@mgl@current@options{,\@mgl@hyper}% mainly
\or
\appto\@mgl@current@options{,\@mgl@hyper}% individual
\or
\appto\@mgl@current@options{,hyper=false}% otheronly
\or
\mgl@isfirstuse
{%
\appto\@mgl@current@options{,hyper=false}% notmainfirst
}%
{%
\appto\@mgl@current@options{,\@mgl@hyper}% notmainfirst
}%
\or
\appto\@mgl@current@options{,\@mgl@hyper}% nototherfirst
\or
\mgl@isfirstuse
{%
\appto\@mgl@current@options{,hyper=false}% notfirst
}%
{%
\appto\@mgl@current@options{,\@mgl@hyper}% notfirst
}%
\fi

```

Append all and then main:

```

\appto\@mgl@current@options{,\@mgl@all,\@mgl@main}%
\else

```

Other element. Location encaps option:

```

\edef\@mgl@current@options{format=\@gls@combined@encapothers}%

```

Indexing option:

```

\ifcase\@gls@combined@indexothers\relax
\appto\@mgl@current@options{,noindex}%
\or
\appto\@mgl@current@options{,noindex=false}%
\or
\@mgl@current@iffirstuse
{\appto\@mgl@current@options{,noindex=false}}%
{\appto\@mgl@current@options{,noindex}}%
\fi

```

Hyperlink option:

```

\ifcase\@gls@combined@hyper\relax
\appto\@mgl@current@options{,hyper=false}% none
\or
\appto\@mgl@current@options{,hyper=false}% allmain
\or
\appto\@mgl@current@options{,hyper=false}% mainly
\or

```

```

\eaopto\@mglscurrentoptions{,\@mgls@hyper}% individual
\or
\eaopto\@mglscurrentoptions{,\@mgls@hyper}% otheronly
\or
\eaopto\@mglscurrentoptions{,\@mgls@hyper}% notmainfirst
\or
\mglsisfirstuse
{%
\eaopto\@mglscurrentoptions{,hyper=false}% nototherfirst
}%
{%
\eaopto\@mglscurrentoptions{,\@mgls@hyper}% nototherfirst
}%
\or
\mglsisfirstuse
{%
\eaopto\@mglscurrentoptions{,hyper=false}% notfirst
}%
{%
\eaopto\@mglscurrentoptions{,\@mgls@hyper}% notfirst
}%
\fi

```

Append all and then others:

```

\eaopto\@mglscurrentoptions{,\@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
\gappto\@mgls@post@hookdefs{%
\noexpand\def\noexpand\mglslastmainlabel
{\expandonce\mglscurrentmainlabel}%
}%
\glstrifwasfirstuse
{%
\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\mglsiflastmainsskipped\@firstoftwo}%
\else
\gappto\@mgls@post@hookdefs{\let\mglsiflastmainsskipped\@secondoftwo}%
\fi

```

Encapsulator may introduce grouping so check here.

```

\ifx\@mgls@do@current@element\@gobble
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementsskipped\@firstoftwo}%
\else
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementsskipped\@secondoftwo}%
\fi
\glstrifwasfirstuse
{%
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@firstoftwo}%
}%
{%
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@secondoftwo}%
}%
\glssifplural
{%
\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
\zappto\@mgls@post@hookdefs{%
\noexpand\mglsunset{\expandonce\mglscurrentmultilabel}}%
\or
\zappto\@mgls@post@hookdefs{%
\noexpand\mglslocalunset{\expandonce\mglscurrentmultilabel}}%
\fi
}%
\glstrmglswrite{#2}%
\egroup
\@mgls@post@hookdefs
\mgls@do@postlinkhook
}

\mglscustompostlinkhook

\newcommand*{\mglscustompostlinkhook}{}

\mglslastelementpostlinkhook

\newcommand*{\mglslastelementpostlinkhook}{%
\let\glstrifwasfirstuse\mglsiflastelementwasfirstuse
\let\glstifplural\mglsiflastelementwasplural
\let\glscapscase\mglsiflastelementcapscase
\let\glslabel\mglslastelementlabel
\glspostlinkhook
}

\mglslastmainpostlinkhook

\newcommand*{\mglslastmainpostlinkhook}{%
\let\glstrifwasfirstuse\mglsiflastmainwasfirstuse
\let\glstifplural\mglsiflastmainwasplural
\let\glscapscase\mglsiflastmaincapscase
\let\glslabel\mglslastmainlabel
\glspostlinkhook
}

\mglsdefcategoryprefix

\newcommand*{\mglsdefcategoryprefix}[2]{%
\csdef{mglsprefix@#1}{#2}%
}

\mglsdescategoryprefix

\newcommand*{\mglsdescategoryprefix}[3]{%
\ifcsdef{mglsprefix@#1}{#2}{#3}%
}

\mglsusecategoryprefix

\newcommand*{\mglsusecategoryprefix}[1]{%
\csuse{mglsprefix@#1}%
}

```

`\mglsprefix`

```
\newcommand*{\mglsprefix}{%
\ifdefempty\mglscurrentcategory
{\mglscurrentprefix}%
{%
\mglshascategoryprefix{\mglscurrentcategory}%
{\mglsecategoryprefix{\mglscurrentcategory}}%
{\mglscurrentprefix}%
}%
}
```

`\mgldefcategorysuffix`

```
\newcommand*{\mgldefcategorysuffix}[2]{%
\csdef{mglssuffix@#1}{#2}%
}
```

`\mglshascategorysuffix`

```
\newcommand*{\mglshascategorysuffix}[3]{%
\ifcsdef{mglssuffix@#1}{#2}{#3}%
}
```

`\mglsecategorysuffix`

```
\newcommand*{\mglsecategorysuffix}[1]{%
\csuse{mglssuffix@#1}%
}
```

`\mglssuffix`

```
\newcommand*{\mglssuffix}{%
\ifdefempty\mglscurrentcategory
{\ifdefempty{\mglscurrentsuffix}{ }\space(\mglscurrentsuffix)}}%
{%
\mglshascategorysuffix{\mglscurrentcategory}%
{\mglsecategorysuffix{\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]{%
\glsattribute{#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}}%
}

```

`\glssetcombinedsepabbrvnbs` Provide shortcut for using non-breakable space following an abbreviation that has already been used.

```

\newcommand*{\glssetcombinedsepabbrvnbs}{%
  \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}{ }\ifglshasshort{##2}{ }{ } }%
}%
\renewcommand*\glscombinedfirstsep}[2]{%
\glshasattribute{##1}{combinedfirstsep}%
{\glsgetattribute{##1}{combinedfirstsep}}%
{\ifglshasshort{##2}{ }\ifglshasshort{##3}{ }{ } }%
}%
\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}%
{\settowidth{\dimen@}{\glsentryshort{##1}}}%
{\settowidth{\dimen@}{\glsentrytext{##1}}}%
\ifdim\dimen@<#1\relax
#2%
\else
\ifglshasshort{##2}%
{\settowidth{\dimen@}{\glsentryshort{##2}}}%
{\settowidth{\dimen@}{\glsentrytext{##2}}}%
\ifdim\dimen@<#1\relax
#2%
\else
\space
\fi
\fi
}%
}%
\renewcommand*\glscombinedsepfirst}[2]{%
```

```

\glshasattribute{##1}{combinedsepfirst}%
{\glsggetattribute{##1}{combinedsepfirst}}%
{%
  \ifglshasshort{##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*{\glscmbinedfirstsep}[2]{%
\glshasattribute{##1}{combinedfirstsep}%
{\glsggetattribute{##1}{combinedfirstsep}}%
{%
  \ifhaslong{##1}%
  {\settowidth{\dimen@}{\glsentrylong{##1}}}%
  {\settowidth{\dimen@}{\glsentryfirst{##1}}}%
  \ifdim\dimen@<#1\relax
    #2%
  \else
    \ifglshasshort{##2}%
    {\settowidth{\dimen@}{\glsentryshort{##2}}}%
    {\settowidth{\dimen@}{\glsentrytext{##2}}}%
    \ifdim\dimen@<#1\relax
      #2%
    \else
      \space
    \fi
  \fi
}%
}%
\renewcommand*{\glscmbinedfirstsepfirst}[2]{%
\glshasattribute{##1}{combinedfirstsepfirst}%
{\glsggetattribute{##1}{combinedfirstsepfirst}}%
{%
  \ifhaslong{##1}%
  {\settowidth{\dimen@}{\glsentrylong{##1}}}%
  {\settowidth{\dimen@}{\glsentryfirst{##1}}}%
  \ifdim\dimen@<#1\relax
    #2%

```

```

\else
  \ifhaslong{##2}%
    {\settowidth{\dimen@}{\glentrylong{##2}}}%
    {\settowidth{\dimen@}{\glentryfirst{##2}}}%
    \ifdim\dimen@<#1\relax
      #2%
    \else
      \space
    \fi
  \fi
}%
}%
}

```

`\@glxtr@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{\glxtr@mglswrite}[1]{%
  \ifx\@glxtr@record@setting\@glxtr@record@setting@off
  \else
    \protected@edef\@glxtr@mglslabel{#1}%
    \ifdef\@glxtr@mglssreflist
      {%
        \expandafter\DTLifinlist\expandafter{\@glxtr@mglslabel}%
        {\@glxtr@mglssreflist}{}%
        {%
          \xappto\@glxtr@mglssreflist{,\expandonce\@glxtr@mglslabel}%
          \if@mglswriteseparaterefs
            \protected@write\@auxout{}{\string\@glxtr@mglssrefs{#1}}%
          \fi
        }%
      }%
    {%
      \global\let\@glxtr@mglssreflist\@glxtr@mglslabel
      \if@mglswriteseparaterefs
        \protected@write\@auxout{}{\string\@glxtr@mglssrefs{#1}}%
      \else
        \AtEndDocument{\immediate\protected@write\@auxout{}%
          {\string\@glxtr@mglssrefs{\@glxtr@mglssreflist}}}%
      \fi
      \@mglswriteseparateref@cond
    }%
  \fi
}

```

`\@glxtr@mglssrefs`

```

\newcommand{\@glxtr@mglssrefs}[1]{%

```

`\if@mglswriteseparaterefs` If this conditional is changed, it must be done before the first instance of any `\mglswriteseparateref`-like command.

```

\newif\if@mglswritesepraterefs \@mglswritesepraterefsfalse

\mglswriteSeparateRefsTrue
\newcommand{\mglswriteSeparateRefsTrue}{\global\@mglswritesepraterefstrue}

\mglswriteSeparateRefsFalse
\newcommand{\mglswriteSeparateRefsFalse}{\global\@mglswritesepraterefsfalse}

\@mglswriteseprateref@cond
\newcommand*{\@mglswriteseprateref@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\mglswrite-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\mglswrite-like command}}%
}

\glxtr@newmglswrite
\newcommand{\glxtr@newmglswrite}[5]{%
\edef\glxtr@newmglswrite@do{%
\noexpand\newrobustcmd*{\expandonce{\csname #1\endcsname}}%
{\noexpand\@glshyp@opt\expandonce{\csname ns@glxtr@#1\endcsname}}%
\noexpand\newcommand*{\expandonce{\csname ns@glxtr@#1\endcsname}}[2][{%
\noexpand\new@ifnextchar[%
{\expandonce{\csname glxtr@#1\endcsname}{####1}{####2}}%
{\expandonce{\csname glxtr@#1\endcsname}{####1}{####2}}[]}%
}%
\noexpand\def\expandonce{\csname glxtr@#1\endcsname}####1####2[####3]{%
\noexpand\def\noexpand\glxtrcurrentmglscsname{#1}%
\noexpand\glxtr@mglswrite@inner{####1}{####2}{####3}%
{\noexpand#2}{\noexpand#3}{\noexpand#4}{\noexpand#5}%
}%
}%
\glxtr@newmglswrite@do
\ifx\@glxtr@record@setting\@glxtr@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\@glxtr@mglswrite@list
{\xappto\@glxtr@mglswrite@list{, #1}}%
{%
\gdef\@glxtr@mglswrite@list{#1}%
\AtEndDocument{\immediate\protected@write\@auxout{%
{\string\@glxtr@mglswrite@list{\@glxtr@mglswrite@list}}}%
}%
\fi

```

```

}

\@glsxtr@mglslslike
\newcommand*{\@glsxtr@mglslslike}[1]{%

\GlsXtrMglsOrGls{\mglsls cs}{\gls cs}{\modifier}[\options]
{\label}{\insert]

\newcommand*{\GlsXtrMglsOrGls}[2]{%
\def\@glsxtr@mglsls@or@gls@mcs{#1}%
\def\@glsxtr@mglsls@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@mglsls@or@gls@mcs[1]{#2}}%
{\@glsxtr@mglsls@or@gls@gcs[1]{#2}}%
}

\s@GlsXtrMglsOrGls
\newcommand*{\s@GlsXtrMglsOrGls}[2][ ]{%
\glsxtrifmulti{#2}%
{\@glsxtr@mglsls@or@gls@mcs*[1]{#2}}%
{\@glsxtr@mglsls@or@gls@gcs*[1]{#2}}%
}

\p@GlsXtrMglsOrGls
\newcommand*{\p@GlsXtrMglsOrGls}[2][ ]{%
\glsxtrifmulti{#2}%
{\@glsxtr@mglsls@or@gls@mcs+[1]{#2}}%
{\@glsxtr@mglsls@or@gls@gcs+[1]{#2}}%
}

```

`\@alt@GlsXtrMglsOrGls`

```
\newcommand*{\@alt@GlsXtrMglsOrGls}[2][\%
\glstrifmulti{#2}%
{\expandafter\@glstr@mgl@or@gl@mcs\@gl@alt@hyp@opt@char[#1]{#2}}%
{\expandafter\@glstr@mgl@or@gl@gcs\@gl@alt@hyp@opt@char[#1]{#2}}%
}
```

`\mgls`

`\mgls[<options>]{<label>}[<insert>]`

Use `\gls` for all elements.

```
\glstr@newmgls{mgls}{\@gls@}{\@gls@}{\@gls@}{\@gls@}%
```

`\mglspl`

`\mglspl[<options>]{<label>}[<insert>]`

Use `\glspl` for all elements.

```
\glstr@newmgls{mglspl}{\@glspl@}{\@glspl@}{\@glspl@}{\@glspl@}%
```

`\mglsmainpl`

`\mglsmainpl[<options>]{<label>}[<insert>]`

Only use `\glspl` for the main element, otherwise use `\gls`.

```
\glstr@newmgls{mglsmainpl}{\@gls@}{\@gls@}{\@glspl@}{\@glspl@}%
```

`\Mgls`

`\Mgls[<options>]{<label>}[<insert>]`

Use `\Gls` for first element and `\gls` for others.

```
\glstr@newmgls{Mgls}{\@Gls@}{\@gls@}{\@Gls@}{\@gls@}%
\glsmfuaddmap{mgls}{Mgls}
```

`\Mglspl`

`\Mglspl[<options>]{<label>}[<insert>]`

Use `\Glspl` for first element and `\glspl` for others.

```
\glstr@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.

```
\glstr@newmgls{Mglsmainpl}{\@Gls@}{\@gls@}{\@Glspl@}{\@glspl@}%
\glsmfuaddmap{mglsmainpl}{Mglsmainpl}
```

`\MGLs[<options>]{<label>}[<insert>]`

`\MGLs`

Use \Gls for all elements.

```
\glstr@newmgl{s}{MGLs}{\@Gls@}{\@Gls@}{\@Gls@}{\@Gls@}%
\glsmfublocker{\MGLs}
```

`\MGLspl[<options>]{<label>}[<insert>]`

`\MGLspl`

Use \Glspl for all elements.

```
\glstr@newmgl{s}{MGLspl}{\@Glspl@}{\@Glspl@}{\@Glspl@}{\@Glspl@}%
\glsmfublocker{\MGLspl}
```

`\MGLsmainpl[<options>]{<label>}[<insert>]`

`\MGLsmainpl`

Start all elements with upper case. Only use plural for main element.

```
\glstr@newmgl{s}{MGLsmainpl}{\@Gls@}{\@Gls@}{\@Glspl@}{\@Glspl@}%
\glsmfublocker{\MGLsmainpl}
```

`\MGLS[<options>]{<label>}[<insert>]`

`\MGLS`

Use \GLS for all elements.

```
\glstr@newmgl{s}{MGLS}{\@GLS@}{\@GLS@}{\@GLS@}{\@GLS@}%
\glsmfublocker{\MGLS}
```

`\MGLSpl[<options>]{<label>}[<insert>]`

`\MGLSpl`

Use \GLSpl for all elements.

```
\glstr@newmgl{s}{MGLSpl}{\@GLSpl@}{\@GLSpl@}{\@GLSpl@}{\@GLSpl@}%
\glsmfublocker{\MGLSpl}
```

`\MGLSmainpl[<options>]{<label>}[<insert>]`

`\MGLSmainpl`

Upper case all elements. Only use plural for main element.

```
\glstr@newmgl{s}{MGLSmainpl}{\@GLS@}{\@GLS@}{\@GLSpl@}{\@GLSpl@}%
\glsmfublocker{\MGLSmainpl}
```

`\@glslongortext@`

```
\def\@glslongortext#1#2[#3]{%
  \ifglshaslong{#2}{\@glstrlong{#1}{#2}[#3]}{\@glstext@{#1}{#2}[#3]}%
}
```

```

\@glsshortortext@
\def\@glsshortortext#1#2[#3]{%
  \ifglshasshort{#2}{\@glxtrshort{#1}{#2}[#3]}\@glstext@{#1}{#2}[#3]}%
}

\@glsfullorfirst@
\def\@glsfullorfirst#1#2[#3]{%
  \ifglshasshort{#2}{\@glxtr@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.

```

\glxtr@newmgls{mglsshort}%
{\@glsshortortext}\@glsshortortext{\@glsshortortext}\@glsshortortext}%

```

\mglslong

`\mglslong[<options>]{<label>}[<insert>]`

Use long or text for all elements.

```

\glxtr@newmgls{mglslong}%
{\@glslongortext}\@glslongortext{\@glslongortext}\@glslongortext}%

```

\mglsfull

`\mglsfull[<options>]{<label>}[<insert>]`

Use full or first for all elements.

```

\glxtr@newmgls{mglsfull}%
{\@glsfullorfirst}\@glsfullorfirst{\@glsfullorfirst}\@glsfullorfirst}%

```


<code>\Mglsshort</code>	<div><code>\Mglsshort[<i><options></i>]{<i><label></i>}[<i><insert></i>]</code></div> <p>Use short or text for all elements with initial cap on first element.</p> <pre> \glxtr@newmgl{s}{Mglsshort}% {\@Glsshortortext}{\@glsshortortext}{\@Glsshortortext}{\@glsshortortext}% \glsmfuaddmap{\mglsshort}{\Mglsshort} </pre>
<code>\Mglslong</code>	<div><code>\Mglslong[<i><options></i>]{<i><label></i>}[<i><insert></i>]</code></div> <p>Use long or text for all elements with initial cap on first element.</p> <pre> \glxtr@newmgl{s}{Mglslong}% {\@Glslongortext}{\@glslongortext}{\@Glslongortext}{\@glslongortext}% \glsmfuaddmap{\mglslong}{\Mglslong} </pre>
<code>\Mglsofull</code>	<div><code>\Mglsofull[<i><options></i>]{<i><label></i>}[<i><insert></i>]</code></div> <p>Use full or first for all elements with initial cap on first element.</p> <pre> \glxtr@newmgl{s}{Mglsofull}% {\@Glsfullorfirst}{\@glsofullorfirst}{\@Glsfullorfirst}{\@glsofullorfirst}% \glsmfuaddmap{\mglsofull}{\Mglsofull} </pre>
<code>\mglsname</code>	<div><code>\mglsname[<i><options></i>]{<i><label></i>}[<i><insert></i>]</code></div> <p>Use name for all elements.</p> <pre> \glxtr@newmgl{s}{mglsname}% {\@Glsname@}{\@glname@}{\@Glsname@}{\@glname@}% </pre>
<code>\Mglsoiname</code>	<div><code>\Mglsoiname[<i><options></i>]{<i><label></i>}[<i><insert></i>]</code></div> <p>Use name for all elements with initial cap on first element.</p> <pre> \glxtr@newmgl{s}{Mglsoiname}% {\@Glsiname@}{\@glsoiname@}{\@Glsiname@}{\@glsoiname@}% \glsmfuaddmap{\mglsoiname}{\Mglsoiname} </pre>
<code>\MGlsiname</code>	<div><code>\MGlsiname[<i><options></i>]{<i><label></i>}[<i><insert></i>]</code></div> <p>Use name for all elements with initial cap on all elements.</p> <pre> \glxtr@newmgl{s}{MGlsiname}% {\@Glsiname@}{\@MGlsiname@}{\@Glsiname@}{\@MGlsiname@}% \glsmfublocker{\MGlsiname} </pre>

```
\@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[<options>]{<label>}[<insert>]`

`\mglssymbol`

Use `\glsymbol` if the symbol key is set otherwise use `\gls`.

```
\glstr@newmgl{s}{mglssymbol}%
{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
```

`\Mglssymbol[<options>]{<label>}[<insert>]`

`\Mglssymbol`

As above but initial the first element if it's not a symbol.

```
\glstr@newmgl{s}{Mglssymbol}%
{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
\glsmfublockmap{\mglssymbol}{\Mglssymbol}
```

`\MGlssymbol[<options>]{<label>}[<insert>]`

`\MGlssymbol`

As above but initial each element if it's not a symbol.

```
\glstr@newmgl{s}{MGlssymbol}%
{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
\glsmfublockmap{\Mglssymbol}{\MGlssymbol}
```

`\mglsfield`

```
\newcommand{\mglsfield}{useri}
```

`\@glsfieldorgls`

```
\def\@glsfieldorgls#1#2[#3]{%
  \glstrifhasfield{\mglsfield}{#2}%
  {\@glsdisp[#1]{#2}{\glscurrentfieldvalue#3}}%
  {\@gls@{#1}{#2}[#3]}%
}
```

`\@Glsfieldorgls`

```
\def\@Glsfieldorgls#1#2[#3]{%
  \glstrifhasfield{\mglsfield}{#2}%
  {\@glsdisp[#1]{#2}{%

```

```

\expandafter\glsentencecase\expandafter{\glscurrentfieldvalue#3}}}%
{\@Gls@{#1}{#2}[#3]]}%
}

```

\mglsusefield

`\mglsusefield[⟨options⟩]{⟨label⟩}[⟨insert⟩]`

Use the field given by \mglsfield.

```

\glsxtr@newmgls{mglsusefield}%
{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}%

```

\Mglsusefield

`\Mglsusefield[⟨options⟩]{⟨label⟩}[⟨insert⟩]`

As above but use initial cap for first element only.

```

\glsxtr@newmgls{Mglsusefield}%
{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}%
\glsmfuaddmap{mglsusefield}{Mglsusefield}

```

\MGlsusefield

`\MGlsusefield[⟨options⟩]{⟨label⟩}[⟨insert⟩]`

As above but use initial cap for all elements.

```

\glsxtr@newmgls{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}%
}

```

\@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@}%
```

```
\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@}%
```

```
\Mppls \Mppls[<options>]{<label>}[<insert>]
```

Use \Pgls for the first element and \gls for the remainder.

```
\glsxtr@newmgls{Mppls}{\@Pglsorgls@}{\@gls@}{\@Pglsorgls@}{\@gls@}%
\glsmfuaddmap{mpgls}{\Mppls}
```

\Mpglsp1

\Mpglsp1[*<options>*]{*<label>*}[*<insert>*]

Use \Pglsp1 for the first element and \glsp1 for the remainder.

\glxstr@newmgls{Mpglsp1}{\@Pglorglsp1@}{\@glsp1@}{\@Pglorglsp1@}{\@glsp1@}%
\glsmfuaddmap{\mpglsp1}{\Mpglsp1}

\Mpglsmainpl

\Mpglsmainpl[*<options>*]{*<label>*}[*<insert>*]

Only use plural for main element and only use first letter uppercase prefixing command for first element.

\glxstr@newmgls{Mpglsmainpl}{\@Pglorglsp1@}{\@glsp1@}{\@Pglorglsp1@}{\@glsp1@}%
\glsmfuaddmap{\mpglsmainpl}{\Mpglsmainpl}

\MPGls

\MPGls[*<options>*]{*<label>*}[*<insert>*]

Use \Pglsp1 for the first element and \Gls for the remainder.

\glxstr@newmgls{MPGls}{\@Pglorglsp1@}{\@Gls@}{\@Pglorglsp1@}{\@Gls@}%
\glsmfublocker{\MPGls}

\MPGlspl

\MPGlspl[*<options>*]{*<label>*}[*<insert>*]

Use \Pglsp1 for the first element and \Glspl for the remainder.

\glxstr@newmgls{MPGlspl}{\@Pglorglsp1@}{\@Glspl@}{\@Pglorglsp1@}{\@Glspl@}%
\glsmfublocker{\MPGlspl}

\MPGlsmainpl

\MPGlsmainpl[*<options>*]{*<label>*}[*<insert>*]

Only use plural for main element and first letter uppercase all elements.

\glxstr@newmgls{MPGlsmainpl}{\@Pglorglsp1@}{\@Gls@}{\@Pglorglsp1@}{\@Glspl@}%
\glsmfublocker{\MPGlsmainpl}

\MPGLS

\MPGLS[*<options>*]{*<label>*}[*<insert>*]

Use \PGLS for the first element and \GLS for the remainder.

\glxstr@newmgls{MPGLS}{\@PGLSorgls@}{\@GLS@}{\@PGLSorgls@}{\@GLS@}%
\glsmfublocker{\MPGLS}

\MPGLSpl

\MPGLSpl[*<options>*]{*<label>*}[*<insert>*]

Use `\PGLSpl` for the first element and `\GLSpl` for the remainder.

```
\glxtr@newmgl{s}{MPGLSpl}{\@PGLSorglspl@}{\@GLSpl@}{\@PGLSorglspl@}{\@GLSpl@}%
\glsmfublocker{\MPGLSpl}
```

```
\MPGLSmainpl[<options>]{<label>}[<insert>]
```

`\MPGLSmainpl`

Only use plural for main element and uppercase all elements.

```
\glxtr@newmgl{s}{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.

`\RequireGlossariesExtraLang`

```
\newcommand*\RequireGlossariesExtraLang[1]{%
  \ifundefined{ver@glossariesxtr-#1.1df}{\input{glossariesxtr-#1.1df}}{}}%
```

`\ProvidesGlossariesExtraLang`

```
\newcommand*\ProvidesGlossariesExtraLang[1]{%
  \ProvidesFile{glossariesxtr-#1.1df}%
}
```

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.)

`\glxtr@loaddialect` The dialect label should be stored in `\this@dialect` before using this command.

```
\newcommand{\glxtr@loaddialect}{%
  \IfTrackedLanguageFileExists{\this@dialect}%
  {glossariesxtr-}% prefix
  {.1df}%
  {%
    \RequireGlossariesExtraLang{\CurrentTrackedTag}%
  }%
  {}% not found
}
```

If `glossaries-extra-bib2gls` has been loaded, `\@glxtrdialecthook` will check for the associated script, otherwise it will do nothing.

```
\@glxtrdialecthook
}
```

```

\@ifpackageloaded{tracklang} {%
  \AnyTrackedLanguages
  {%
    \ForEachTrackedDialect{\this@dialect}{\glxtr@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.

```

\@glxtr@redefstyles
and set the style:
\@glxtr@do@style

```

2 Predefined Abbreviation Styles (glossaries-extra-abbrstyles.def)

```

\ProvidesFile{glossaries-extra-abbrstyles.def}[2022/06/16 v1.48b (NLCT)]

```

This file contains the predefined abbreviation styles. Some helper commands first.

```

\glxtrlongshortname
\newcommand*{\glxtrlongshortname}{%
  \glxpabbrrvfont{\the\glsshorttok}{\glscategorylabel}%
}

```

Provide convenient wrappers for common formats.

`\glxtrlongformat`

```

\glxtrlongformat{\label}{\insert}{\longfmtcs}

```

```

\newcommand*{\glxtrlongformat}[3]{%

```

Don't add inner formatting if `markwords` attribute set as the inner formatting is implemented within `\glxtrword` and `\glxtrwordsep`.

```

\gl@ifattribute{#1}{markwords}{true}%
{%
  \ifglxtrininsertinside
    #3{\glssaccesslong{#1}\glxtrgenentrytextfmt{#2}}%
  \else
    #3{\glssaccesslong{#1}}\glxtrgenentrytextfmt{#2}%
  \fi
}%
{%
  \ifglxtrininsertinside
    #3{\glssaccessfmtlong{#2}{\glxtrgenentrytextfmt}{#1}}%

```

```

\else
  #3{\glsaccessfmtlong}{\glsxtrgenentrytextfmt}{#1}}%
  \glsxtrgenentrytextfmt{#2}}%
\fi
}%
}%

```

\glsxtrlongplformat

```
\glsxtrlongplformat{<label>}{<insert>}{<longfmtcs>}
```

```

\newcommand*{\glsxtrlongplformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrininsertinside
      #3{\glsaccesslongpl{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\glsaccesslongpl{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrininsertinside
      #3{\glsaccessfmtlongpl{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\glsaccessfmtlongpl{}{\glsxtrgenentrytextfmt}{#1}}%
      \glsxtrgenentrytextfmt{#2}%
    \fi
  }%
}%

```

\Glsxtrlongformat

```
\Glsxtrlongformat{<label>}{<insert>}{<longfmtcs>}
```

```

\newcommand*{\Glsxtrlongformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglsxtrininsertinside
      #3{\Glsaccesslong{#1}\glsxtrgenentrytextfmt{#2}}%
    \else
      #3{\Glsaccesslong{#1}}\glsxtrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglsxtrininsertinside
      #3{\Glsaccessfmtlong{#2}{\glsxtrgenentrytextfmt}{#1}}%
    \else
      #3{\Glsaccessfmtlong{}{\glsxtrgenentrytextfmt}{#1}}%

```



```

\glstrgenentrytextfmt{#2}%
\fi
}%
}%

```

`\Glsxtrlongplformat{<label>}{<insert>}{<longfmtcs>}`

`\Glsxtrlongplformat`

```

\newcommand*{\Glsxtrlongplformat}[3]{%
\glsifattribute{#1}{markwords}{true}%
{%
\ifglstrinsertinside
#3{\Glsaccesslongpl{#1}\glstrgenentrytextfmt{#2}}%
\else
#3{\Glsaccesslongpl{#1}}\glstrgenentrytextfmt{#2}%
\fi
}%
}%
\ifglstrinsertinside
#3{\Glsaccessfmlongpl{#2}{\glstrgenentrytextfmt}{#1}}%
\else
#3{\Glsaccessfmlongpl{}{\glstrgenentrytextfmt}{#1}}%
\glstrgenentrytextfmt{#2}%
\fi
}%
}%

```

`\GLSxtrlongformat{<label>}{<insert>}{<longfmtcs>}`

`\GLSxtrlongformat`

```

\newcommand*{\GLSxtrlongformat}[3]{%
\glsifattribute{#1}{markwords}{true}%
{%
\ifglstrinsertinside
#3{\GLSaccesslong{#1}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}%
\else
#3{\GLSaccesslong{#1}}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}%
\fi
}%
}%
\ifglstrinsertinside
#3{\GLSaccessfmlong{#2}{\glstrgenentrytextfmt}{#1}}%
\else
#3{\GLSaccessfmlong{}{\glstrgenentrytextfmt}{#1}}%
\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}%
\fi

```

```
}%
}%
```

```
\GLSxtrlongplformat{<label>}{<insert>}{<longfmtcs>}
```

\GLSxtrlongplformat

```
\newcommand*{\GLSxtrlongplformat}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglstrinsertinside
      #3{\GLSaccesslongpl{#1}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}%
    \else
      #3{\GLSaccesslongpl{#1}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    \ifglstrinsertinside
      #3{\GLSaccessfmtlongpl{#2}{\glstrgenentrytextfmt}{#1}}%
    \else
      #3{\GLSaccessfmtlongpl{}{\glstrgenentrytextfmt}{#1}}%
      \mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

```
\glstrlongformatgrp{<label>}{<insert>}{<longfmtcs>}
```

\glstrlongformatgrp

Add grouping around insert.

```
\newcommand*{\glstrlongformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglstrinsertinside
      #3{\glsaccesslong{#1}{\glstrgenentrytextfmt{#2}}}%
    \else
      #3{\glsaccesslong{#1}{\glstrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    #3{\glsaccessfmtlong{}{\glstrgenentrytextfmt}{#1}}%
    \ifglstrinsertinside
      {#3{\glstrgenentrytextfmt{#2}}}%
    \else
      {\glstrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\glxstrlongplformatgrp

\glxstrlongformatplgrp{<label>}{<insert>}{<longfmtcs>}

Add grouping around insert.

```
\newcommand*{\glxstrlongplformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglxstrinsertinside
      #3{\glsaccesslongpl{#1}{\glxstrgenentrytextfmt{#2}}}%
    \else
      #3{\glsaccesslongpl{#1}{\glxstrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    #3{\glsaccessfmtlongpl{}{\glxstrgenentrytextfmt}{#1}}%
    \ifglxstrinsertinside
      {#3{\glxstrgenentrytextfmt{#2}}}%
    \else
      {\glxstrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\Glsxtrlongformatgrp

\Glsxtrlongformatgrp{<label>}{<insert>}{<longfmtcs>}

Add grouping around insert.

```
\newcommand*{\Glsxtrlongformatgrp}[3]{%
  \glsifattribute{#1}{markwords}{true}%
  {%
    \ifglxstrinsertinside
      #3{\Glsaccesslong{#1}{\glxstrgenentrytextfmt{#2}}}%
    \else
      #3{\Glsaccesslong{#1}{\glxstrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    #3{\Glsaccessfmtlong{}{\glxstrgenentrytextfmt}{#1}}%
    \ifglxstrinsertinside
      {#3{\glxstrgenentrytextfmt{#2}}}%
    \else
      {\glxstrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

\Glsxtrlongplformatgrp

\Glsxtrlongformatplgrp{<label>}{<insert>}{<longfmtcs>}

Add grouping around insert.

```
\newcommand*{\GLSxtrlongplformatgrp}[3]{%
  \glsifattribute{#1}{keywords}{true}%
  {%
    \ifglstrinsertinside
      #3{\GLSaccesslongpl{#1}{\glstrgenentrytextfmt{#2}}}%
    \else
      #3{\GLSaccesslongpl{#1}{\glstrgenentrytextfmt{#2}}}%
    \fi
  }%
  {%
    #3{\GLSaccessfmtlongpl{}{\glstrgenentrytextfmt}{#1}}%
    \ifglstrinsertinside
      {#3{\glstrgenentrytextfmt{#2}}}%
    \else
      {\glstrgenentrytextfmt{#2}}%
    \fi
  }%
}%
```

`\GLSxtrlongformatgrp{<label>}{<insert>}{<longfmtcs>}`

`\GLSxtrlongformatgrp`

Add grouping around insert.

```
\newcommand*{\GLSxtrlongformatgrp}[3]{%
  \glsifattribute{#1}{keywords}{true}%
  {%
    \ifglstrinsertinside
      #3{\GLSaccesslong{#1}{\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}}%
    \else
      #3{\GLSaccesslong{#1}{\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}}%
    \fi
  }%
  {%
    #3{\GLSaccessfmtlong{}{\glstrgenentrytextfmt}{#1}}%
    \ifglstrinsertinside
      {\mfirstucMakeUppercase{#3{\glstrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}%
    \fi
  }%
}%
```

`\GLSxtrlongformatplgrp{<label>}{<insert>}{<longfmtcs>}`

`\GLSxtrlongplformatgrp`

Add grouping around insert.

```
\newcommand*{\GLSxtrlongplformatgrp}[3]{%
```

```

\glsifattribute{#1}{markwords}{true}%
{%
  \ifglstrinsertinside
    #3{\GLSaccesslongpl{#1}{\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}}%
  \else
    #3{\GLSaccesslongpl{#1}{\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}}%
  \fi
}%
{%
  #3{\GLSaccessfmtlongpl{#1}{\glstrgenentrytextfmt{#1}}}%
  \ifglstrinsertinside
    {\mfirstucMakeUppercase{#3{\glstrgenentrytextfmt{#2}}}}%
  \else
    {\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}%
  \fi
}%
}%

```

\glstrshortformat

```
\glstrshortformat{<label>}{<insert>}{<shortfmtcs>}
```

```

\newcommand*{\glstrshortformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglstrinsertinside
      #3{\glsaccessshort{#1}\glstrgenentrytextfmt{#2}}%
    \else
      #3{\glsaccessshort{#1}\glstrgenentrytextfmt{#2}}%
    \fi
  }%
  {%
    \ifglstrinsertinside
      #3{\glsaccessfmtshort{#2}{\glstrgenentrytextfmt{#1}}}%
    \else
      #3{\glsaccessfmtshort{#2}{\glstrgenentrytextfmt{#1}}}%
      \glstrgenentrytextfmt{#2}%
    \fi
  }%
}%

```

\glstrshortplformat

```
\glstrshortplformat{<label>}{<insert>}{<shortfmtcs>}
```

```

\newcommand*{\glstrshortplformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%

```

```

\ifglstrinsertinside
  #3{\glsaccesssshortpl{#1}\glstrgenentrytextfmt{#2}}%
\else
  #3{\glsaccesssshortpl{#1}}\glstrgenentrytextfmt{#2}%
\fi
}%
{%
\ifglstrinsertinside
  #3{\glsaccessfmtshortpl{#2}{\glstrgenentrytextfmt}{#1}}%
\else
  #3{\glsaccessfmtshortpl{}{\glstrgenentrytextfmt}{#1}}%
\glstrgenentrytextfmt{#2}%
\fi
}%
}%

```

`\Glsxtrshortformat{<label>}{<insert>}{<shortfmtcs>}`

`\Glsxtrshortformat`

```

\newcommand*{\Glsxtrshortformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglstrinsertinside
      #3{\Glsaccesssshort{#1}\glstrgenentrytextfmt{#2}}%
    \else
      #3{\Glsaccesssshort{#1}}\glstrgenentrytextfmt{#2}%
    \fi
  }%
  {%
    \ifglstrinsertinside
      #3{\Glsaccessfmtshort{#2}{\glstrgenentrytextfmt}{#1}}%
    \else
      #3{\Glsaccessfmtshort{}{\glstrgenentrytextfmt}{#1}}%
      \glstrgenentrytextfmt{#2}%
    \fi
  }%
}%

```

`\Glsxtrshortplformat{<label>}{<insert>}{<shortfmtcs>}`

`\Glsxtrshortplformat`

```

\newcommand*{\Glsxtrshortplformat}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglstrinsertinside
      #3{\Glsaccesssshortpl{#1}\glstrgenentrytextfmt{#2}}%

```

```

\else
  #3{\Glsaccesssshortpl{#1}}\glstrgenentrytextfmt{#2}%
\fi
}%
{%
\ifglstrinsertinside
  #3{\Glsaccessfmtshortpl{#2}}{\glstrgenentrytextfmt}{#1}}%
\else
  #3{\Glsaccessfmtshortpl{}{\glstrgenentrytextfmt}{#1}}%
\glstrgenentrytextfmt{#2}%
\fi
}%
}%
}%

```

\GLSxtrshortformat

`\GLSxtrshortformat{<label>}{<insert>}{<shortfmtcs>}`

```

\newcommand*{\GLSxtrshortformat}[3]{%
\glsifattribute{#1}{markshortwords}{true}%
{%
\ifglstrinsertinside
  #3{\Glsaccesssshort{#1}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}%
\else
  #3{\Glsaccesssshort{#1}}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}%
\fi
}%
{%
\ifglstrinsertinside
  #3{\Glsaccessfmtshort{#2}}{\glstrgenentrytextfmt}{#1}}%
\else
  #3{\Glsaccessfmtshort{}{\glstrgenentrytextfmt}{#1}}%
\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}%
\fi
}%
}%
}%

```

\GLSxtrshortplformat

`\GLSxtrshortplformat{<label>}{<insert>}{<shortfmtcs>}`

```

\newcommand*{\GLSxtrshortplformat}[3]{%
\glsifattribute{#1}{markshortwords}{true}%
{%
\ifglstrinsertinside
  #3{\Glsaccesssshortpl{#1}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}}%
\else
  #3{\Glsaccesssshortpl{#1}}\mfirstucMakeUppercase{\glstrgenentrytextfmt{#2}}%

```

```

\fi
}%
{%
\ifglxtrinsertinside
#3{\GLSaccessfmtshortpl{#2}{\glxtrgenentrytextfmt}{#1}}%
\else
#3{\GLSaccessfmtshortpl{}{\glxtrgenentrytextfmt}{#1}}%
\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}%
\fi
}%
}%

```

\glxtrshortformatgrp

\glxtrshortformatgrp{<label>}{<insert>}{<shortfmtcs>}

Add grouping around insert.

```

\newcommand*{\glxtrshortformatgrp}[3]{%
\glsifattribute{#1}{markshortwords}{true}%
{%
\ifglxtrinsertinside
#3{\glsaccesssshort{#1}{\glxtrgenentrytextfmt{#2}}}%
\else
#3{\glsaccesssshort{#1}{\glxtrgenentrytextfmt{#2}}}%
\fi
}%
{%
#3{\glsaccessfmtshort{}{\glxtrgenentrytextfmt}{#1}}%
\ifglxtrinsertinside
{#3{\glxtrgenentrytextfmt{#2}}}%
\else
{\glxtrgenentrytextfmt{#2}}%
\fi
}%
}%

```

\glxtrshorttplformatgrp

\glxtrshorttplformatgrp{<label>}{<insert>}{<shortfmtcs>}

Add grouping around insert.

```

\newcommand*{\glxtrshorttplformatgrp}[3]{%
\glsifattribute{#1}{markshortwords}{true}%
{%
\ifglxtrinsertinside
#3{\glsaccesssshorttpl{#1}{\glxtrgenentrytextfmt{#2}}}%
\else
#3{\glsaccesssshorttpl{#1}{\glxtrgenentrytextfmt{#2}}}%
\fi
}%

```



```

{%
#3{\Glsaccessfmtshortpl}{\Glsxtrgenentrytextfmt}{#1}}%
\ifGlsxtrininsertinside
  {#3{\Glsxtrgenentrytextfmt{#2}}}%
\else
  {\Glsxtrgenentrytextfmt{#2}}%
\fi
}%
}%

```

`\Glsxtrshortformatgrp{<label>}{<insert>}{<shortfmtcs>}`

`\Glsxtrshortformatgrp`

Add grouping around insert.

```

\newcommand*{\Glsxtrshortformatgrp}[3]{%
\Glsifattribute{#1}{markshortwords}{true}%
{%
\ifGlsxtrininsertinside
#3{\Glsaccesssshort{#1}{\Glsxtrgenentrytextfmt{#2}}}%
\else
#3{\Glsaccesssshort{#1}{\Glsxtrgenentrytextfmt{#2}}}%
\fi
}%
}%
#3{\Glsaccessfmtshort}{\Glsxtrgenentrytextfmt}{#1}}%
\ifGlsxtrininsertinside
  {#3{\Glsxtrgenentrytextfmt{#2}}}%
\else
  {\Glsxtrgenentrytextfmt{#2}}%
\fi
}%
}%

```

`\Glsxtrshorttplformatgrp{<label>}{<insert>}{<shortfmtcs>}`

`\Glsxtrshorttplformatgrp`

Add grouping around insert.

```

\newcommand*{\Glsxtrshorttplformatgrp}[3]{%
\Glsifattribute{#1}{markshortwords}{true}%
{%
\ifGlsxtrininsertinside
#3{\Glsaccesssshortpl{#1}{\Glsxtrgenentrytextfmt{#2}}}%
\else
#3{\Glsaccesssshortpl{#1}{\Glsxtrgenentrytextfmt{#2}}}%
\fi
}%
}%
#3{\Glsaccessfmtshortpl}{\Glsxtrgenentrytextfmt}{#1}}%

```

```

\ifglxtrinsertinside
  {#3{\glxtrgenentrytextfmt{#2}}}%
\else
  {\glxtrgenentrytextfmt{#2}}%
\fi
}%
}%

```

`\GLSxtrshortformatgrp{<label>}{<insert>}{<shortfmtcs>}`

`\GLSxtrshortformatgrp`

Add grouping around insert.

```

\newcommand*{\GLSxtrshortformatgrp}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglxtrinsertinside
      #3{\GLSaccessshort{#1}{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}}}%
    \else
      #3{\GLSaccessshort{#1}{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}}}%
    \fi
  }%
  {%
    #3{\GLSaccessfmtshort{#1}{\glxtrgenentrytextfmt{#1}}}%
    \ifglxtrinsertinside
      {\mfirstucMakeUppercase{#3{\glxtrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}}%
    \fi
  }%
}%

```

`\GLSxtrshorttplformatgrp{<label>}{<insert>}{<shortfmtcs>}`

`\GLSxtrshorttplformatgrp`

Add grouping around insert.

```

\newcommand*{\GLSxtrshorttplformatgrp}[3]{%
  \glsifattribute{#1}{markshortwords}{true}%
  {%
    \ifglxtrinsertinside
      #3{\GLSaccessshorttpl{#1}{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}}}%
    \else
      #3{\GLSaccessshorttpl{#1}{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}}}%
    \fi
  }%
  {%
    #3{\GLSaccessfmtshorttpl{#1}{\glxtrgenentrytextfmt{#1}}}%
    \ifglxtrinsertinside
      {\mfirstucMakeUppercase{#3{\glxtrgenentrytextfmt{#2}}}}%
    \fi
  }%
}%

```

```

\else
  {\mfirstucMakeUppercase{\glxstrgenentrytextfmt{#2}}}%
\fi
}%
}%

```

`\glxstrlongshortformat`

```

\glxstrlongshortformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}

```

```

\newcommand*{\glxstrlongshortformat}[4]{%
  \glxstrlongformat{#1}{#2}{#3}%
  \glxstrfullsep{#1}%
  \glxstrparen{\glxstrshortformat{#1}{#4}}%
}%

```

`\glxstrlongshortplformat`

```

\glxstrlongshortplformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}

```

```

\newcommand*{\glxstrlongshortplformat}[4]{%
  \glxstrlongplformat{#1}{#2}{#3}%
  \glxstrfullsep{#1}%
  \glxstrparen{\glxstrshortplformat{#1}{#4}}%
}%

```

`\Glsxtrlongshortformat`

```

\Glsxtrlongshortformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}

```

```

\newcommand*{\Glsxtrlongshortformat}[4]{%
  \Glsxtrlongformat{#1}{#2}{#3}%
  \glxstrfullsep{#1}%
  \glxstrparen{\glxstrshortformat{#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

```

\glsxtrshortlongplformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}

```

```

\newcommand*{\glsxtrshortlongplformat}[4]{%
  \glsxtrshortplformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrlongplformat{#1}{#3}}%
}%

```

\Glsxtrshortlongformat

```
\Glsxtrshortlongformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

```
\newcommand*{\Glsxtrshortlongformat}[4]{%
  \Glsxtrshortformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrlongformat{#1}{#3}}%
}%
```

\Glsxtrshortlongplformat

```
\Glsxtrshortlongplformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

```
\newcommand*{\Glsxtrshortlongplformat}[4]{%
  \Glsxtrshortplformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\glsxtrlongplformat{#1}{#3}}%
}%
```

\GLSxtrshortlongformat

```
\GLSxtrshortlongformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

```
\newcommand*{\GLSxtrshortlongformat}[4]{%
  \GLSxtrshortformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\GLSxtrlongformat{#1}{#3}}%
}%
```

\GLSxtrshortlongplformat

```
\GLSxtrshortlongplformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

```
\newcommand*{\GLSxtrshortlongplformat}[4]{%
  \GLSxtrshortplformat{#1}{#2}{#4}%
  \glsxtrfullsep{#1}%
  \glsxtrparen{\GLSxtrlongplformat{#1}{#3}}%
}%
```

\glsxtrfootnotelongformat

```
\glsxtrfootnotelongformat{<label>}{<longfmtcs>}
```

```
\newcommand*{\glxtrfootnotelongformat}[2]{%
  \glxtrlongformat{#1}{#2}%
}%
```

\glxtrfootnotelongplformat

```
\glxtrfootnotelongplformat{<label>}{<longfmtcs>}
```

```
\newcommand*{\glxtrfootnotelongplformat}[2]{%
  \glxtrlongplformat{#1}{#2}%
}%
```

\glxtrpostfootnotelongformat

```
\glxtrpostfootnotelongformat{<label>}{<longfmtcs>}
```

```
\newcommand*{\glxtrpostfootnotelongformat}{%
  \glxtrfootnotelongformat
}%
```

\glxtrpostusersshortformat

```
\glxtruserpostshortformat{<label>}{<shortfmtcs>}
```

```
\newcommand*{\glxtrpostusersshortformat}[2]{%
  \glxtrifallcaps
  {\GLSxtrusersshortformat{#1}{#2}}%
  {\glxtrusersshortformat{#1}{#2}}%
}%
```

\glxtrusersshortformat

```
\glxtrusersshortformat{<label>}{<shortfmtcs>}
```

```
\newcommand*{\glxtrusersshortformat}[2]{%
  \glxtruserparen{\glxtrshortformat{#1}{#2}}{#1}%
}%
```

\glxtrusersshorttplformat

```
\glxtrusersshorttplformat{<label>}{<shortfmtcs>}
```

```
\newcommand*{\glxtrusersshorttplformat}[2]{%
  \glxtruserparen{\glxtrshorttplformat{#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*\glxtruserlongplformat}[2]{%
  \glxtruserparen{\glxtrlongplformat{#1}{#2}}{#1}%
}%
```

$\backslash\text{GLSxtruserlongplformat}\{\langle label \rangle\}\{\langle longfmtcs \rangle\}$

$\backslash\text{GLSxtruserlongplformat}$

```
\newcommand*\GLSxtruserlongplformat}[2]{%
  \GLSxtruserparen{\GLSxtrlongplformat{#1}{#2}}{#1}%
}%
```

$\backslash\text{glxtruserlongshortformat}\{\langle label \rangle\}\{\langle insert \rangle\}\{\langle longfmtcs \rangle\}$
 $\{\langle shortfmtcs \rangle\}$

$\backslash\text{glxtruserlongshortformat}$

```
\newcommand*\glxtruserlongshortformat}[4]{%
  \glxtrlongformat{#1}{#2}{#3}%
  \glxtrusershortformat{#1}{#4}%
}%
```

$\backslash\text{glxtruserlongshorttplformat}\{\langle label \rangle\}\{\langle insert \rangle\}\{\langle longfmtcs \rangle\}$
 $\{\langle shortfmtcs \rangle\}$

$\backslash\text{glxtruserlongshorttplformat}$

```
\newcommand*\glxtruserlongshorttplformat}[4]{%
  \glxtrlongplformat{#1}{#2}{#3}%
  \glxtrusershorttplformat{#1}{#4}%
}%
```

$\backslash\text{Glsxtruserlongshortformat}\{\langle label \rangle\}\{\langle insert \rangle\}\{\langle longfmtcs \rangle\}$
 $\{\langle shortfmtcs \rangle\}$

$\backslash\text{Glsxtruserlongshortformat}$

```
\newcommand*\Glsxtruserlongshortformat}[4]{%
  \Glsxtrlongformat{#1}{#2}{#3}%
  \glxtrusershortformat{#1}{#4}%
}%
```

$\backslash\text{Glsxtruserlongshorttplformat}\{\langle label \rangle\}\{\langle insert \rangle\}\{\langle longfmtcs \rangle\}$
 $\{\langle shortfmtcs \rangle\}$

$\backslash\text{Glsxtruserlongshorttplformat}$


```
\newcommand*\GLSxtruserlongshortplformat}[4]{%
  \GLSxtrlongplformat{#1}{#2}{#3}%
  \GLSxtrusershortplformat{#1}{#4}%
}%
```

```
\GLSxtruserlongshortformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

\GLSxtruserlongshortformat

```
\newcommand*\GLSxtruserlongshortformat}[4]{%
  \GLSxtrlongformat{#1}{#2}{#3}%
  \GLSxtrusershortformat{#1}{#4}%
}%
```

```
\GLSxtruserlongshortplformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

\GLSxtruserlongshortplformat

```
\newcommand*\GLSxtruserlongshortplformat}[4]{%
  \GLSxtrlongplformat{#1}{#2}{#3}%
  \GLSxtrusershortplformat{#1}{#4}%
}%
```

```
\GLSxtrusershortlongformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

\GLSxtrusershortlongformat

```
\newcommand*\GLSxtrusershortlongformat}[4]{%
  \GLSxtrshortformat{#1}{#2}{#3}%
  \GLSxtruserlongformat{#1}{#4}%
}%
```

```
\GLSxtrusershortlongplformat{<label>}{<insert>}{<longfmtcs>}
{<shortfmtcs>}
```

\GLSxtrusershortlongplformat

```
\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

```
\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.

```
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrlongshortname},
  sort={\the\glsshorttok},
  first={\glstrfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\glstrfirstxabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glstrfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\glstrfirstxabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
  plural={\glstrxabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glstrxabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  description={\the\glslongtok}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glstrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
\renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvdefaultfont{##1}}%
\renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlongdefaultfont{##1}}%
\renewcommand*{\glstrlongfont}[1]{\glstrlongdefaultfont{##1}}%
\renewcommand*{\glstrrevert}[1]{\glstrdefaultrevert{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*{\glstrfullformat}[2]{%
  \glstrlongshortformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrvfont}%
}%
\renewcommand*{\glstrfullplformat}[2]{%
  \glstrlongshortplformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrvfont}%
}%
```

```

\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}{%
  \glstxplongfont{\the\glslongtok}{\glscategorylabel}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\glstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}%
}

```

`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={\glstxplongfont{\the\glslongtok}{\glscategorylabel}%
    \protect\glsxtrfullsep{\the\glslabeltok}%
    \protect\glsxtrparen{\glstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glstxplongfont{\the\glslongpltok}{\glscategorylabel}%
    \protect\glsxtrfullsep{\the\glslabeltok}%
    \protect\glsxtrparen{\glstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%

```

The text key should only have the short form.

```
text={\glstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
```

```
plural={\glxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glxtrsetcomplexstyle{\the\glslabeltok}{3}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
\GlsXtrUseAbbrStyleFmts{long-short}%
}
```

`\glxtrshortlongname`

```
\newcommand*{\glxtrshortlongname}{%
\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
}
```

`short-long` Short form followed by long form in parenthesis on first use.

```
\newabbreviationstyle{short-long}%
{%
```

Set accessibility attributes if enabled.

```
\glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortlongname},
sort={\the\glsshorttok},
description={\the\glslongtok},%
first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
\protect\glxtrfullsep{\the\glslabeltok}%
\protect\glxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
\protect\glxtrfullsep{\the\glslabeltok}%
\protect\glxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%

text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
```

```
}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
\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}{%
  \glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
  \protect\glsxtrfullsep{\the\glslabeltok}%
  \protect\glsxtrparen{\glxplongfont{\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.

```
\glxstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxstrshortlongdescname},
  sort={\glxstrshortlongdescsort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \protect\glxstrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \protect\glxstrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%

  text={\glxspabbrvfont{\the\glsshorttok}{\glscategorylabel}},%

  plural={\glxspabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glxstrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\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}}%
```

`\glxtrabbrvfootnote`

```
\glxtrabbrvfootnote{<label>}{<long>}
```

Command used by footnote abbreviation styles. The default definition ignores the first argument. The second argument *<long>* includes the font changing command and may be the singular or plural form, depending on the command that was used (for example, `\gls` or `\glspl`).

```
\newcommand*{\glxtrabbrvfootnote}[2]{\footnote{#2}}
```

`\glxtrpostabrvfootnote` Used by post-footnote style to include formatting.

```
\newrobustcmd*{\glxtrpostabrvfootnote}[2]{%
  \glxtrabrvfootnote{#1}%
  {#2\glxtrpostfootnotelongformat{#1}{\glslongfootnotefont}}%
}
```

`\xpglxtrpostabrvfootnote` Perform all the appropriate expansions to ensure `\glslabel` and `\glxtrassignlinktextfmt` are expanded as they may be lost by the time the footnote occurs.

```
\newcommand*{\xpglxtrpostabrvfootnote}{%
  \expandafter\expandafter\expandafter
  \glxtrpostabrvfootnote
  \expandafter\expandafter\expandafter
  {\expandafter\glslabel\expandafter}\expandafter
  {\glxtrassignlinktextfmt}%
}
```

`\glxtrfootnotename`

```
\newcommand*{\glxtrfootnotename}{%
  \glxpabrvfont{\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 time the footnote text is processed. (Neither is the outer formatting applied.)

```
\glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%

  first={\glslongfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glxtrabrvfootnote{\the\glslabeltok}%
    {\protect\glslongfootnotefont{\the\glslongtok}}},%
  firstplural={\glslongfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glxtrabrvfootnote{\the\glslabeltok}%
    {\protect\glslongfootnotefont{\the\glslongpltok}}},%

  text={\glxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glxpabrvfont{\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}%
  \glsattribute{\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*{\glxtrinlinefullformat}[2]{%
  \glxtrshortlongformat{##1}{##2}%
  {\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glxtrshortlongplformat{##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}
```

\glxtrfootnotedesname

```

\newcommand*{\glxtrfootnotedesname}{%
  \glxspabbrvfont{\the\glsshorttok}{\glscategorylabel}%
  \protect\glxtrfullsep{\the\glslabeltok}%
  \protect\glxtrparen{\glxplongfont{\the\glslongtok}{\glscategorylabel}}%
}

```

\glxtrfootnotedesort

```
\newcommand*{\glxtrfootnotedesort}{\the\glsshorttok}
```

short-footnote-desc Like short-footnote but with user supplied description.

```

\newabbreviationstyle{short-footnote-desc}{%
  {%

```

Set accessibility attributes if enabled

```
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrfootnotedesname},
  sort={\glxtrfootnotedesort},
}

```

```

first={\glfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}%
\protect\glstrabbrvfootnote{\the\glslabeltok}%
{\protect\glfirstlongfootnote{\the\glslongtok}}},%
firstplural={\glfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
\protect\glstrabbrvfootnote{\the\glslabeltok}%
{\protect\glfirstlongfootnote{\the\glslongpltok}}},%
text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\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}{%
\glsetattribute{\the\glslabeltok}{nohyperfirst}{true}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glsetattribute{\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 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.)

```
\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
name={\glstrfootnotename},
sort={\the\glsshorttok},
description={\the\glslongtok},%
first={\glfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
firstplural={\glfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%

text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\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{glstrpostlink\glscategorylabel}{%
```

The footnote needs to be suppressed in the inline form. Previously this was done by redefining `glstrsetupfulldefs` but that interferes with other styles. Instead, this now uses `\glstrifwasglslike`

```
\glstrifwasglslikeandfirstuse
{%
```

Ensure `\glslabel` and `\glstrassignlinktextfmt` are expanded as they may be lost by the time the footnote occurs.

```
\glstrdopostpunc{\expandafter\expandafter\expandafter
\glstrpostabbrvfootnote
\expandafter\expandafter\expandafter
{\expandafter\glslabel\expandafter}\expandafter
{\glstrassignlinktextfmt}}}%
}%}%
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
\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*{\glstrfullformat}[2]{%
\glstrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glstrfullplformat}[2]{%
\glstrshortplformat{##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={\glsxtrfootnotedesname},
sort={\glsxtrfootnotedesort},
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}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
{}%
}%
}%
{}%
\GlsXtrUseAbbrStyleFmts{postfootnote}%
}
```

postfootnote-desc

```
\letabbreviationstyle{postfootnote-desc}{short-postfootnote-desc}
```

`\glsxtrshortnolongname`

```
\newcommand*{\glsxtrshortnolongname}{%
\glspabrvfont{\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={\glstrshortnolongname},
  sort={\the\glsshorttok},
  first={\glstxfirstpabrvfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glstxfirstpabrvfont{\the\glsshortptok}{\glscategorylabel}},
  text={\glstxpabrvfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glstxpabrvfont{\the\glsshortptok}{\glscategorylabel}},
  description={\the\glslongtok}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
\renewcommand*{\glabbrvfont}[1]{\glabbrvdefaultfont{##1}}%
\renewcommand*{\glstfirstabbrvfont}[1]{\glstfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glstfirstlongfont}[1]{\glstfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short form followed by the long form in parentheses.

```
\renewcommand*{\glstxtrinlinefullformat}[2]{%
  \glstxtrshortlongformat{##1}{##2}%
  {\glstfirstlongfont}{\glstfirstabbrvfont}}%
}%
\renewcommand*{\glstxtrinlinefullplformat}[2]{%
  \glstxtrshortlongplformat{##1}{##2}%
  {\glstfirstlongfont}{\glstfirstabbrvfont}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
  {\glstfirstlongfont}{\glstfirstabbrvfont}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
  {\glstfirstlongfont}{\glstfirstabbrvfont}}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
  {\glstfirstlongfont}{\glstfirstabbrvfont}}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
  {\glstfirstlongfont}{\glstfirstabbrvfont}}%
}%
```

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*{\glxtrfullformat}[2]{%
  \glxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrshorttplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshorttplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshorttplformat{##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}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong}%
}

```

\glxtrshortdescname

```

\newcommand*{\glxtrshortdescname}{%
  \glxtpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
  \protect\glxtrfullsep{\the\glslabeltok}%
  \protect\glxtrparen{\glxplongfont{\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.

```
\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrshortdescname},
  sort={\the\glsshorttok},
  first={\glstrfirstabbrfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glstrfirstabbrfont{\the\glsshortpltok}{\glscategorylabel}},
  text={\glstrabbrfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glstrabbrfont{\the\glsshortpltok}{\glscategorylabel}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
\renewcommand*{\glstrabbrfont[1]{\glstrabbrdefaultfont{##1}}}%
\renewcommand*{\glstrfirstabbrfont}[1]{\glstrfirstabbrdefaultfont{##1}}%
\renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlongdefaultfont{##1}}%
\renewcommand*{\glstrlongfont}[1]{\glstrlongdefaultfont{##1}}%
```

The inline full form displays the short format followed by the long form in parentheses.

```
\renewcommand*{\glstrinlinefullformat}[2]{%
  \glstrshortlongformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrfont}%
}%
\renewcommand*{\glstrinlinefullplformat}[2]{%
  \glstrshortlongplformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
  {\glstrfirstlongfont}{\glstrfirstabbrfont}%
}%
```

```

\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]{%
  \glsxtrshorttplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshorttplformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshorttplformat{##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*{\glxtrinlinefullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glxtrlongshortplformat{##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}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%

```

```

\GlsXtrUseAbbrStyleFmts{no long-short}%
}

```

`\glxtrlongnoshortdescname`

```

\newcommand*{\glxtrlongnoshortdescname}{%
\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={\glxtrlongnoshortdescname},
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}{%
\glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
}%

```

In case the user wants to mix and match font styles, these are redefined here.

```

\renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
\renewcommand*{\glxtrabbrvfont}[1]{\glxtrabbrvdefaultfont{##1}}%
\renewcommand*{\glxtrfirstabbrvfont}[1]{\glxtrfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glxtrfirstlongfont}[1]{\glxtrfirstlongdefaultfont{##1}}%
\renewcommand*{\glxtrlongfont}[1]{\glxtrlongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

\renewcommand*{\glxtrsubsequentfmt}[2]{%
\glxtrlongformat{##1}{##2}{\glslongfont}%
}%
\renewcommand*{\glxtrsubsequentplfmt}[2]{%
\glxtrlongplformat{##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]{%
  \glsxtrlongshorttplformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrlongshorttplformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
  {\glsfirstlongfont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrlongshorttplformat{##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}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}
```

```
}%
```

```
{%
```

```
\GlsXtrUseAbbrStyleFmts{long-noshort-desc}%
}
```

`\glxstrlongnoshortname`

```
\newcommand*{\glxstrlongnoshortname}{%
\glxspabbrvfont{\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.

```
\glxstrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxstrlongnoshortname},
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}{%
\glissetattribute{\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}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-noshort}%
}
```

2.2 Predefined Styles (Small Capitals)

These styles use `\textsc` for the short form.

`\glxtrscfont` Maintained for backward-compatibility.

```
\newcommand*{\glxtrscfont}[1]{\textsc{#1}}
```

`\glsabbrvscfont` Added for consistent naming.

```
\newcommand*{\glsabbrvscfont}{\glxtrscfont}
```

`\glxtrfirstscfont` Maintained for backward-compatibility.

```
\newcommand*{\glxtrfirstscfont}[1]{\glsabbrvscfont{#1}}
```

`\glsfirstabbrvscfont` Added for consistent naming.

```
\newcommand*{\glsfirstabbrvscfont}{\glxtrfirstscfont}
```

and for the default short form suffix:

`\glxtrscsuffix` `\protect` needs to come inside `\glxtrscsuffix` to avoid interfering with all caps.

```
\newcommand*{\glxtrscsuffix}{\protect\glstextup{\glsxtrabbrvpluralsuffix}}
```

`\glxtrscinvert` Cancel smallcaps.

```
\newcommand*{\glxtrscinvert}[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}%
  \glsattribute{\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]{\glsxtrscinvert{##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*{\glxstrfullformat}[2]{%
  \glxstrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\glxstrfullplformat}[2]{%
  \glxstrlongshortplformat{##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.

```
\glxstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxstrlongshortdescname},
  sort={\glxstrlongshortdescsort},%
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \protect\glxstrparen{\glsfirstxpabrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \protect\glxstrparen{\glsfirstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
  text={\glxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
}
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glxstrsetcomplexstyle{\the\glslabeltok}{3}%
}
```

```

\glshasattribute{\the\glslabeltok}{regular}%
{%
  \glsssetAttribute{\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={\glsfirstxppabrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glsxtrfullsep{\the\glslabeltok}%
    \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
  firstplural={\glsfirstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glsxtrfullsep{\the\glslabeltok}%
    \protect\glsxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
  text={\glsxppabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsxppabrvfont{\the\glsshortpltok}{\glscategorylabel}}}%

```

Unset the regular attribute if it has been set.

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

\renewcommand*{\abbrvpluralsuffix}{\glsxtrscsuffix}%
\renewcommand*{\glsabrvfont}[1]{\glsabrvscfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%

```

```
\renewcommand*\glxstrrevert[1]{\glxstrscrevert{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*\glxstrfullformat[2]{%
  \glxstrshortlongformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*\glxstrfullplformat[2]{%
  \glxstrshortlongplformat{##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.

```
\glxstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*\CustomAbbreviationFields{%
  name={\glxstrshortlongdescname},
  sort={\glxstrshortlongdescsort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \protect\glxstrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \protect\glxstrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
  text={\glsexpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsexpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%
```

Unset the regular attribute if it has been set.

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glstrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%

```

As short-sc-long style:

```

\GlsXtrUseAbbrStyleFmts{short-sc-long}%
}

```

short-sc

```

\newabbreviationstyle{short-sc}%
{%

```

Set accessibility attributes if enabled.

```

\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel

```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrshortnolongname},
  sort={\the\glssshorttok},
  first={\glsfirstxpabrvfont{\the\glssshorttok}{\glscategorylabel}},
  firstplural={\glsfirstxpabrvfont{\the\glssshortptok}{\glscategorylabel}},
  text={\glsexpabrvfont{\the\glssshorttok}{\glscategorylabel}},
  plural={\glsexpabrvfont{\the\glssshortptok}{\glscategorylabel}},
  description={\the\glslongtok}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

\renewcommand*{\abrvpluralsuffix}{\glstrscsuffix}%
\renewcommand*{\glsabrvfont[1]{\glsabrvscfont{##1}}}%
\renewcommand*{\glsfirstabrvfont[1]{\glsfirstabrvscfont{##1}}}%
\renewcommand*{\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}}%
\renewcommand*{\glslongfont[1]{\glslongdefaultfont{##1}}}%
\renewcommand*{\glstrrevert[1]{\glstrscrevert{##1}}}%

```

The inline full form displays the short form followed by the long form in parentheses.

```

\renewcommand*{\glstrinlinefullformat}[2]{%
  \glstrshortlongformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvscfont}%
}%
\renewcommand*{\glstrinlinefullplformat}[2]{%

```

```

\glstrshortlongplformat{##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*{\glstrfullformat}[2]{%
\glstrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
}%

\renewcommand*{\glstrfullplformat}[2]{%
\glstrshortplformat{##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.

```
\glstrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrshortdescname},
  sort={\the\glsshorttok},
  first={\glstrshortfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glstrshortfont{\the\glsshorttok}{\glscategorylabel}},
  text={\glstrshortfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glstrshortfont{\the\glsshorttok}{\glscategorylabel}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glstrsetattribute{\the\glstrlabeltok}{regular}{true}}%
}%
{%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrscsuffix}%
\renewcommand*{\glstrshortfont}[1]{\glstrshortfont{##1}}%
\renewcommand*{\glstrshortfont}[1]{\glstrshortfont{##1}}%
\renewcommand*{\glstrlongfont}[1]{\glstrlongfont{##1}}%
\renewcommand*{\glstrlongfont}[1]{\glstrlongfont{##1}}%
\renewcommand*{\glstrrevert}[1]{\glstrrevert{##1}}%
```

The inline full form displays the short format followed by the long form in parentheses.

```
\renewcommand*{\glstrinlinefullformat}[2]{%
  \glstrshortlongformat{##1}{##2}%
  {\glstrlongfont{\glstrshortfont{##1}}}%
}%
\renewcommand*{\glstrinlinefullplformat}[2]{%
  \glstrshortlongplformat{##1}{##2}%
  {\glstrlongfont{\glstrshortfont{##1}}}%
}%
\renewcommand*{\GlsXtrinlinefullformat}[2]{%
  \GlsXtrshortlongformat{##1}{##2}%
  {\glstrlongfont{\glstrshortfont{##1}}}%
}%
\renewcommand*{\GlsXtrinlinefullplformat}[2]{%
  \GlsXtrshortlongplformat{##1}{##2}%
  {\glstrlongfont{\glstrshortfont{##1}}}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
  {\glstrlongfont{\glstrshortfont{##1}}}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
  {\glstrlongfont{\glstrshortfont{##1}}}%
}%
```

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*\glstrfullformat}[2]{%
  \glstrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
}%
\renewcommand*\glstrfullplformat}[2]{%
  \glstrshortplformat{##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*\glstrinlinefullformat}[2]{%
  \glstrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%
\renewcommand*\glstrinlinefullplformat}[2]{%
  \glstrlongshortplformat{##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]{\glsxtrscinvert{##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:

```
\@glxtr@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 `\glxtrshort`.

```

\newabbreviationstyle{long-noshort-sc-desc}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
}%
{%

```

Use smallcaps and adjust the plural suffix to revert to upright.

```

\renewcommand*{\abbrvpluralsuffix}{\glxtrscsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glxtrrevert}[1]{\glxtrsc revert{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

\renewcommand*{\glxtrsubsequentfmt}[2]{%
  \glxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*{\glxtrsubsequentplfmt}[2]{%
  \glxtrlongplformat{##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*{\glxtrinlinefullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvscfont}%
}%

```

```

\renewcommand*\glxstrinlinefullplformat}[2]{%
  \glxstrlongshortplformat{##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*\glxstrfullformat}[2]{%
  \glxstrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*\glxstrfullplformat}[2]{%
  \glxstrlongplformat{##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:

```
\@glxstr@deprecated@abbrstyle{long-desc-sc}{long-noshort-sc-desc}
```

`short-sc-footnote`

```

\newabbreviationstyle{short-sc-footnote}%
{%

```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glstrfirstabbrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glstrabbrvfootnote{\the\glslabeltok}%
    {\protect\glstrfirstlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glstrfirstabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glstrabbrvfootnote{\the\glslabeltok}%
    {\protect\glstrfirstlongfootnotefont{\the\glslongpltok}}},%
  text={\glstrabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glstrabbrvfont{\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}{%
  \glstrsetattribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glstrhasattribute{\the\glslabeltok}{regular}%
  {%
    \glstrsetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{\%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrscsuffix}%
\renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvscfont{##1}}%
\renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvscfont{##1}}%
\renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlongfootnotefont{##1}}%
\renewcommand*{\glstrlongfont}[1]{\glstrlongfootnotefont{##1}}%
\renewcommand*{\glstrrevert}[1]{\glstrsc revert{##1}}%
```

The full format displays the short form followed by the long form as a footnote.

```
\renewcommand*{\glstrfullformat}[2]{%
  \glstrshortformat{##1}{##2}{\glstrfirstabbrvscfont}%
  \protect\glstrabbrvfootnote{##1}%
  {\glstrfootnotelongformat{##1}{\glstrfirstlongfootnotefont}}%
}%
\renewcommand*{\glstrfullplformat}[2]{%
  \glstrshortplformat{##1}{##2}{\glstrfirstabbrvscfont}%
  \protect\glstrabbrvfootnote{##1}%
  {\glstrfootnotelongplformat{##1}{\glstrfirstlongfootnotefont}}%
}%
\renewcommand*{\GlsXtrfullformat}[2]{%
  \GlsXtrshortformat{##1}{##2}{\glstrfirstabbrvscfont}%
```

```

\protect\glstrabbrvfootnote{##1}%
{\glstrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
\protect\glstrabbrvfootnote{##1}%
{\glstrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\Glsxtrshortformat{##1}{##2}{\glsfirstabbrvscfont}%
\protect\glstrabbrvfootnote{##1}%
{\glstrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvscfont}%
\protect\glstrabbrvfootnote{##1}%
{\glstrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%

```

The first use full form and the inline full form use the short (long) style.

```

\renewcommand*{\glxtrinlinefullformat}[2]{%
\glxtrshortlongformat{##1}{##2}%
{\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glxtrshortlongplformat{##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}%
}%
}

```

footnote-sc Backward compatibility:

```
\@glxtr@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.

```
\glstrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrfootnotedesname},
  sort={\glstrfootnotedesort},
  first={\glstxppabrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glstrabbrvfootnote{\the\glslabeltok}%
    {\protect\glstxlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glstrabbrvfootnote{\the\glslabeltok}%
    {\protect\glstxlongfootnotefont{\the\glslongpltok}}},%
  text={\glstxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glstxpabrvfont{\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}{%
  \glsssetAttribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-footnote}%
}
```

short-sc-postfootnote

```
\newabbreviationstyle{short-sc-postfootnote}%
{%
```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glstxppabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glstxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glstxpabrvfont{\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 `glxtrsetupfulldefs` but that interferes with other styles. Instead, this now uses `\glxtrifwasglslike`

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\csdef{glxtrpostlink\glscategorylabel}{%
\glxtrifwasglslikeandfirstuse
{%
\glxtrdopostpunc{\xpglxtrpostabbrvfootnote}%
}%
}%
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{%
```

Use smallcaps and adjust the plural suffix to revert to upright.

```
\renewcommand*{\abbrvpluralsuffix}{\glxtrscsuffix}%
\renewcommand*{\glabbrvfont}[1]{\glabbrvscfont{##1}}%
\renewcommand*{\glfirstabbrvfont}[1]{\glfirstabbrvscfont{##1}}%
\renewcommand*{\glfirstlongfont}[1]{\glfirstlongfootnotefont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
\renewcommand*{\glxtrrevert}[1]{\glxtrsc revert{##1}}%
```

The full format displays the short form. The long form is deferred.

```
\renewcommand*{\glxtrfullformat}[2]{%
\glxtrshortformat{##1}{##2}{\glfirstabbrvscfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glxtrshortplformat{##1}{##2}{\glfirstabbrvscfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\Glsxtrshortformat{##1}{##2}{\glfirstabbrvscfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\Glsxtrshortplformat{##1}{##2}{\glfirstabbrvscfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
\GLSxtrshortformat{##1}{##2}{\glfirstabbrvscfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
\GLSxtrshortplformat{##1}{##2}{\glfirstabbrvscfont}%
}%
```

The first use full form and the inline full form use the short (long) style.

```

\renewcommand*{\glxtrinlinefullformat}[2]{%
  \glxtrshortlongformat{##1}{##2}%
  {\glsfirstlongfootnotefont}{\glsfirstabbrvscfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glxtrshortlongplformat{##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:

```
\@glxtr@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.

```
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrfootnotedesname},
  sort={\glxtrfootnotedesort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glxppabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glxppabbrvfont{\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 `glxtrsetupfulldefs` but that interferes with other styles. Instead, this now uses `\glxtrifwasglslike`

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
```



```

\csdef{glxtrpostlink\glscategorylabel}{%
  \glxtrifwasglslikeandfirstuse
  {%
    \glxtrdopostpunc{\xpglxtrpostabbrvfootnote}%
  }%
  {}%
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
  \glissetattribute{\the\glslabeltok}{regular}{false}%
}%
{}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-postfootnote}%
}

```

2.3 Predefined Styles (Fake Small Capitals)

These styles require the `relabel` package, which must be loaded by the user. These styles all use:

`\glxtrsmfont` Maintained for backward compatibility.
`\newcommand*{\glxtrsmfont}[1]{\textsmaller{#1}}`

`\glxabbrvsmfont` Added for consistent naming.
`\newcommand*{\glxabbrvsmfont}{\glxtrsmfont}`

`\glxtrfirstsmfont` Maintained for backward compatibility.
`\newcommand*{\glxtrfirstsmfont}[1]{\glxabbrvsmfont{#1}}`

`\glfirstabbrvsmfont` Added for consistent naming.
`\newcommand*{\glfirstabbrvsmfont}{\glxtrfirstsmfont}`

and for the default short form suffix:

`\glxtrsmsuffix`
`\newcommand*{\glxtrsmsuffix}{\glxtrabbrvpluralsuffix}`

`\glxtrsmrevert`
`\newcommand*{\glxtrsmrevert}[1]{\textlarger{#1}}`

`long-short-sm`
`\newabbreviationstyle{long-short-sm}%`
`{%`

Set accessibility attributes if enabled.

`\glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel`

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrlongshortname},
  sort={\the\glsshorttok},
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
    \protect\glxtrfullsep{\the\glslabeltok}%
    \protect\glxtrparen{\glsfirstxpabrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
    \protect\glxtrfullsep{\the\glslabeltok}%
    \protect\glxtrparen{\glsfirstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
  text={\glxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glxpabrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  description={\the\glslongtok}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {%
  }%
}%
\renewcommand*{\glxtrfullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\GlsXtrFullFormat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\GlsXtrFullPlFormat}[2]{%
  \glxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrrevert}[1]{\glxtrsmrevert{##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*{\glxtrfullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\GlsXtrFullFormat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\GlsXtrFullPlFormat}[2]{%
  \glxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrlongshortplformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glxtrrevert}[1]{\glxtrsmrevert{##1}}%
```

```

\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}%
  }%
  {}%
}%
{%

```

As long-short-sm style:

```

\GlsXtrUseAbbrStyleFmts{long-short-sm}%
}

```

short-sm-long Now the short (long) version

```

\newabbreviationstyle{short-sm-long}%
{%

```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrshortlongname},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxpabrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}}%
  \protect\glstrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
  firstplural={\glsfirstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}}%
  \protect\glstrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
  text={\glsexpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsexpabrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glstrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetattribute{\the\glslabeltok}{regular}{false}%
  }%
}%
{}%
}%
{}%

\renewcommand*{\glsabrvfont}[1]{\glsabrvsmfont{##1}}%
\renewcommand*{\glsfirstabrvfont}[1]{\glsfirstabrvsmfont{##1}}%
\renewcommand*{\abrvpluralsuffix}{\glstrmsuffix}%
\renewcommand*{\glstrrevert}[1]{\glstrsmrevert{##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*{\glstrfullformat}[2]{%
  \glstrshortlongformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\glstrfullplformat}[2]{%
  \glstrshortlongplformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabrvsmfont}%
}%
\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.

```
\glstrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrshortnolongname},
  sort={\the\glsshorttok},
  first={\glstxppabrvfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}},
  text={\glstxpabrvfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}},
  description={\the\glslongtok}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%

\renewcommand*\glsabrvfont[1]{\glsabrvsmfont{##1}}%
\renewcommand*\glstxppabrvfont[1]{\glstxppabrvsmfont{##1}}%
\renewcommand*\abrvpluralsuffix{\glstxrsmssuffix}%
\renewcommand*\glstxrrevert[1]{\glstxrsmrevert{##1}}%
\renewcommand*\glstxfirstlongfont[1]{\glstxfirstlongdefaultfont{##1}}%
\renewcommand*\glstxlongfont[1]{\glstxlongdefaultfont{##1}}%
```

The inline full form displays the short form followed by the long form in parentheses.

```
\renewcommand*\glstxtrinlinefullformat[2]{%
  \glstxrshortlongformat{##1}{##2}%
  {\glstxfirstlongdefaultfont}{\glstxfirstabrvsmfont}%
}%
\renewcommand*\glstxtrinlinefullplformat[2]{%
  \glstxrshortlongplformat{##1}{##2}%
  {\glstxfirstlongdefaultfont}{\glstxfirstabrvsmfont}%
}%
\renewcommand*\Glsxtrinlinefullformat[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
  {\glstxfirstlongdefaultfont}{\glstxfirstabrvsmfont}%
}%
\renewcommand*\Glsxtrinlinefullplformat[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
  {\glstxfirstlongdefaultfont}{\glstxfirstabrvsmfont}%
}%
\renewcommand*\GLSxtrinlinefullformat[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
  {\glstxfirstlongdefaultfont}{\glstxfirstabrvsmfont}%
}%
\renewcommand*\GLSxtrinlinefullplformat[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
  {\glstxfirstlongdefaultfont}{\glstxfirstabrvsmfont}%
}%
```

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*\glstrfullformat}[2]{%
  \glstrshortformat{##1}{##2}{\glfirstabbrvsmfont}%
}%
\renewcommand*\glstrfullplformat}[2]{%
  \glstrshortplformat{##1}{##2}{\glfirstabbrvsmfont}%
}%
\renewcommand*\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glfirstabbrvsmfont}%
}%
\renewcommand*\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glfirstabbrvsmfont}%
}%
\renewcommand*\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glfirstabbrvsmfont}%
}%
\renewcommand*\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glfirstabbrvsmfont}%
}%
}
```

`short-sm-nolong`

```
\letabbreviationstyle{short-sm-nolong}{short-sm}
```

`short-sm-desc`

```
\newabbreviationstyle{short-sm-desc}{%
  {%
```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*\CustomAbbreviationFields{%
  name={\glstrshortdescname},
  sort={\the\glsshorttok},
  first={\glfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},
  text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
\renewcommand*\GlsXtrPostNewAbbreviation{%
  \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```

```
\renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
\renewcommand*\glfirstabbrvfont[1]{\glfirstabbrvsmfont{##1}}%
\renewcommand*\abbrvpluralsuffix{\glstrsmsuffix}%
\renewcommand*\glstrrevert[1]{\glstrsmrevert{##1}}%
\renewcommand*\glfirstlongfont[1]{\glfirstlongdefaultfont{##1}}%
\renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%
```

The inline full form displays the short format followed by the long form in parentheses.

```
\renewcommand*{\glxstrinlinefullformat}[2]{%
  \glxstrshortlongformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\glxstrinlinefullplformat}[2]{%
  \glxstrshortlongplformat{##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*{\glxstrfullformat}[2]{%
  \glxstrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\glxstrfullplformat}[2]{%
  \glxstrshortplformat{##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*{\glxtrinlinefullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glxtrlongshortplformat{##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 `\glxtrshort`.

```
\newabbreviationstyle{long-noshort-sm}%
{%
```

Set accessibility attributes if enabled.

```
\glxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrlongnoshortname},
  sort={\the\glsshorttok},
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
```

```

firstplural={\glstxplongfont{\the\glslongpltok}{\glscategorylabel}},
text={\glstxplongfont{\the\glslongtok}{\glscategorylabel}},
plural={\glstxplongfont{\the\glslongpltok}{\glscategorylabel}},%
description={\the\glslongtok}%
}%
\renewcommand*\GlsXtrPostNewAbbreviation}{%
\glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%

```

```

\renewcommand*\glssabrvfont[1]{\glssabrvsmfont{##1}}%
\renewcommand*\glssfirstabrvfont[1]{\glssfirstabrvsmfont{##1}}%
\renewcommand*\glssabrvpluralsuffix{\glstxrsmssuffix}%
\renewcommand*\glstxrrevert[1]{\glstxrsmrevert{##1}}%
\renewcommand*\glssfirstlongfont[1]{\glssfirstlongdefaultfont{##1}}%
\renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%

```

The format for subsequent use (not used when the regular attribute is set).

```

\renewcommand*\glstxrsubsequentfmt}[2]{%
\glstxrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*\glstxrsubsequentplfmt}[2]{%
\glstxrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*\Glsxrsubsequentfmt}[2]{%
\Glsxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*\Glsxrsubsequentplfmt}[2]{%
\Glsxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*\GLSxrsubsequentfmt}[2]{%
\GLSxtrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*\GLSxrsubsequentplfmt}[2]{%
\GLSxtrlongplformat{##1}{##2}{\glslongdefaultfont}%
}%

```

The inline full form displays the long format followed by the short form in parentheses.

```

\renewcommand*\glstxrinlinefullformat}[2]{%
\glstxrlongshortformat{##1}{##2}%
{\glssfirstlongdefaultfont}{\glssfirstabrvsmfont}%
}%
\renewcommand*\glstxrinlinefullplformat}[2]{%
\glstxrlongshortplformat{##1}{##2}%
{\glssfirstlongdefaultfont}{\glssfirstabrvsmfont}%
}%
\renewcommand*\Glsxrinlinefullformat}[2]{%
\Glsxtrlongshortformat{##1}{##2}%
{\glssfirstlongdefaultfont}{\glssfirstabrvsmfont}%
}%

```

```

\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-sm` Backward compatibility:

```

\@glsxtr@deprecated@abbrstyle{long-sm}{long-noshort-sm}

```

`long-noshort-sm-desc` 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*{\glxstrfullformat}[2]{%
  \glxstrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\glxstrfullplformat}[2]{%
  \glxstrlongplformat{##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:

```
\@glxstr@deprecated@abbrstyle{long-desc-sm}{long-noshort-sm-desc}
```

short-sm-footnote

```
\newabbreviationstyle{short-sm-footnote}%
{%
```

Set accessibility attributes if enabled.

```
\glxstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxstrfootnotename},
  sort={\the\glsshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxppabrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glxtrabbrvfootnote{\the\glslabeltok}%
    {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glsfirstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glxtrabbrvfootnote{\the\glslabeltok}%
    {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
  text={\glxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glxpabrvfont{\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}{%
  \glsssetAttribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glshasattribute{\the\glslabeltok}{regular}%
}
```

```

    {%
      \glsssetAttribute{\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*{\glxtrinlinefullplformat}[2]{%
  \glxtrshortlongplformat{##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:

```
\@glxtr@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.

```
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrfootnotedesname},
  sort={\glxtrfootnotedesort},
  first={\glxtrfirstpabrvfont{\the\glsshorttok}{\glscategorylabel}%
    \protect\glxtrabbrvfootnote{\the\glslabeltok}%
    {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
  firstplural={\glxtrfirstpabrvfont{\the\glsshortpltok}{\glscategorylabel}%
    \protect\glxtrabbrvfootnote{\the\glslabeltok}%
    {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
  text={\glxtpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glxtpabrvfont{\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}{%
  \glissetattribute{\the\glslabeltok}{nohyperfirst}{true}%
  \glshasattribute{\the\glslabeltok}{regular}%

```

```

    {%
      \glsssetattribute{\the\glslabelltok}{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\glssshorttok},
  description={\the\glslongtok},%
  first={\glsfirstxpabrvfont{\the\glssshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabrvfont{\the\glssshortpltok}{\glscategorylabel}},%
  text={\glsexpabrvfont{\the\glssshorttok}{\glscategorylabel}},%
  plural={\glsexpabrvfont{\the\glssshortpltok}{\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 `glsxtrsetupfuldefs` but that interferes with other styles. Instead, this now uses `\glsxtrifwasglsl`

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glsxtrpostlink\glscategorylabel}{%
    \glsxtrifwasglsllikeandfirstuse
    {%
      \glxtrdopostpunc{\xpglsxtrpostabrvfootnote}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabelltok}{regular}%
  {%
    \glsssetattribute{\the\glslabelltok}{regular}{false}%
  }%
  {}%
}%
}%
{%

```

```

\renewcommand*\glsabrvfont[1]{\glsabrvsmfont{##1}}%
\renewcommand*\glsfirstabrvfont[1]{\glsfirstabrvsmfont{##1}}%
\renewcommand*\abrvpluralsuffix{\glxtrsmsuffix}%

```



```

\renewcommand*\glxstrrevert[1]{\glxstrsmrevert{##1}}%
\renewcommand*\glsfirstlongfont[1]{\glsfirstlongfootnotefont{##1}}%
\renewcommand*\glslongfont[1]{\glslongfootnotefont{##1}}%

```

The full format displays the short form. The long form is deferred.

```

\renewcommand*\glxstrfullformat[2]{%
  \glxstrshortformat{##1}{##2}{\glsfirstabbrvsmfont}%
}%
\renewcommand*\glxstrfullplformat[2]{%
  \glxstrshortplformat{##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*\glxstrinlinefullformat[2]{%
  \glxstrshortlongformat{##1}{##2}%
  {\glsfirstlongfootnotefont}{\glsfirstabbrvsmfont}%
}%
\renewcommand*\glxstrinlinefullplformat[2]{%
  \glxstrshortlongplformat{##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:

```
\@glxtr@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.

```
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%  
  name={\glxtrfootnotedesname},  
  sort={\glxtrfootnotedesort},  
  first={\glsfirftpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%  
  firstplural={\glsfirftpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%  
  text={\glxppabbrvfont{\the\glsshorttok}{\glscategorylabel}},%  
  plural={\glxppabbrvfont{\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 `glxtrsetupfulldefs` but that interferes with other styles. Instead, this now uses `\glxtrifwasglsl`

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%  
  \csdef{glxtrpostlink\glscategorylabel}{%  
    \glxtrifwasglslandfirstuse  
  }%
```

Ensure `\glsl` is expanded as it may be lost by the time the footnote occurs.

```
\glxtrdopostpunc{\xpglxtrpostabbrvfootnote}%  
}%  
{}%  
}%  
\glshasattribute{\the\glslabeltok}{regular}%  
{%  
  \glissetattribute{\the\glslabeltok}{regular}{false}%  
}%  
{}%  
}%  
}%  
{%  
  \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:

\glxtremsuffix
    \newcommand*{\glxtremsuffix}{\glxtrabbrvpluralsuffix}

\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}}%

\glxtremrevert
    \newcommand*{\glxtremrevert}[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.
        \glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
Setup the default fields.
        \renewcommand*{\CustomAbbreviationFields}{%
            name={\glxtrlongshortname},
            sort={\the\glsshorttok},
            first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}%
            \protect\glxtrfullsep{\the\glslabeltok}%
            \protect\glxtrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
            firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}%
            \protect\glxtrfullsep{\the\glslabeltok}%
            \protect\glxtrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
            text={\glxspabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
            plural={\glxspabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
            description={\the\glslongtok}}%
        \renewcommand*{\GlsXtrPostNewAbbreviation}{%
            \glxtrsetcomplexstyle{\the\glslabeltok}{3}%
            \glshasattribute{\the\glslabeltok}{regular}%
            {%
                \glissetattribute{\the\glslabeltok}{regular}{false}%
            }%
        }%
    }%
}

\renewcommand*\glsabbrvfont[1]{\glsabbrvemfont{##1}}%
\renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvemfont{##1}}%
\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*\glxtrrevert[1]{\glxtremrevert{##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{\glsfirstxppabrvfont{\the\glsshorttok}{\glscategorylabel}}},%
firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
\protect\glsxtrfullsep{\the\glslabeltok}}%
\protect\glsxtrparen{\glsfirstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
text={\glsxpabrvfont{\the\glsshorttok}{\glscategorylabel}}},%
plural={\glsxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glstrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\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.

```
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields. `\glslongemfont` is used in the description since `\glsdesc` doesn't set the style.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrlongshortname},
  sort={\the\glsshorttok},
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\glsfirstxpabrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\glsfirstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}},%

  text={\glxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glxpabrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  description={\protect\glslongemfont{\the\glslongtok}}}%
}
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
  \glstrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{}%
```

```

\renewcommand*{\abbrvpluralsuffix}{\glstremsuffix}%
\renewcommand*{\glstrrevert}[1]{\glstremrevert{##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*{\glstrfullformat}[2]{%
  \glstrlongshortformat{##1}{##2}%
  {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glstrfullplformat}[2]{%
  \glstrlongshortplformat{##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.

```

\glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel

```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrlongshortdescname},
  sort={\glstrlongshortdescsort},%
  first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}},%
  firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}},%
}

```

```

text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%

```

Unset the regular attribute if it has been set.

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glxtrsetcomplexstyle{\the\glslabeltok}{3}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\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.

```

\glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel

```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortlongname},
sort={\the\glsshorttok},
description={\the\glslongtok},%
first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}%
\protect\glxtrfullsep{\the\glslabeltok}%
\protect\glxtrparen{\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
\protect\glxtrfullsep{\the\glslabeltok}%
\protect\glxtrparen{\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%

```

Unset the regular attribute if it has been set.

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glxtrsetcomplexstyle{\the\glslabeltok}{3}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{\%

```

Mostly as short-long style:

```
\renewcommand*\abbrvpluralsuffix{\glxtremsuffix}%
\renewcommand*\glxstrrevert[1]{\glxtremrevert{##1}}%
\renewcommand*\glxabbrvfont[1]{\glxabbrvemfont{##1}}%
\renewcommand*\glxfirstabbrvfont[1]{\glxfirstabbrvemfont{##1}}%
\renewcommand*\glxfirstlongfont[1]{\glxfirstlongdefaultfont{##1}}%
\renewcommand*\glxlongfont[1]{\glxlongdefaultfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*\glxstrfullformat[2]{%
  \glxstrshortlongformat{##1}{##2}%
  {\glxfirstlongdefaultfont}{\glxfirstabbrvemfont}}%
}%
\renewcommand*\glxstrfullplformat[2]{%
  \glxstrshortlongplformat{##1}{##2}%
  {\glxfirstlongdefaultfont}{\glxfirstabbrvemfont}}%
}%
\renewcommand*\GLxstrfullformat[2]{%
  \GLxstrshortlongformat{##1}{##2}%
  {\glxfirstlongdefaultfont}{\glxfirstabbrvemfont}}%
}%
\renewcommand*\GLxstrfullplformat[2]{%
  \GLxstrshortlongplformat{##1}{##2}%
  {\glxfirstlongdefaultfont}{\glxfirstabbrvemfont}}%
}%
\renewcommand*\GLSxtrfullformat[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
  {\glxfirstlongdefaultfont}{\glxfirstabbrvemfont}}%
}%
\renewcommand*\GLSxtrfullplformat[2]{%
  \GLSxtrshortlongplformat{##1}{##2}%
  {\glxfirstlongdefaultfont}{\glxfirstabbrvemfont}}%
}%
}
```

`short-em-long-desc` As before but user provides description

```
\newabbreviationstyle{short-em-long-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glxstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*\CustomAbbreviationFields{%
  name={\glxstrshortlongdescname},
  sort={\glxstrshortlongdescsort},
  first={\glxfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}%
  \protect\glxstrfullsep{\the\glslabeltok}%
  \protect\glxstrparen{\glxfirstxplongfont{\the\glxlongtok}{\glscategorylabel}}},%
  firstplural={\glxfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
  \protect\glxstrfullsep{\the\glslabeltok}}%
```



```

\protect\glxtrparen{\glfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
}%

```

Unset the regular attribute if it has been set.

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glxtrsetcomplexstyle{\the\glslabeltok}{3}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}%
\GlsXtrUseAbbrStyleFmts{short-em-long}%
}

```

short-em-long-em

```

\newabbreviationstyle{short-em-long-em}%
{%

```

Set accessibility attributes if enabled.

```

\glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel

```

Setup the default fields. `\glslongemfont` is used in the description since `\glstdesc` doesn't set the style.

```

\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortlongname},
sort={\the\glsshorttok},
description={\protect\glslongemfont{\the\glslongtok}},%
first={\glfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}}%
\protect\glxtrfullsep{\the\glslabeltok}%
\protect\glxtrparen{\glfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
firstplural={\glfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}%
\protect\glxtrfullsep{\the\glslabeltok}%
\protect\glxtrparen{\glfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%

text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}}}%

```

Unset the regular attribute if it has been set.

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
\glxtrsetcomplexstyle{\the\glslabeltok}{3}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%

```

```

}%
}%
{%

\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*{\glxtrrevert}[1]{\glxtremrevert{##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*{\glxtrfullformat}[2]{%
  \glxtrshortlongformat{##1}{##2}%
  {\glsfirstlongemfont}{\glsfirstabbrvemfont}}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrshortlongplformat{##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={\glxtrshortlongdescname},%
  sort={\glxtrshortlongdescsort},%
  first={\glsfirstxabbrvfont{\the\glsshorttok}{\glscategorylabel}}%
  \protect\glxtrfullsep{\the\glslabeltok}}%

```

```

\protect\glstrparen{\glstfirstxplongfont{\the\glslongtok}{\glscategorylabel}}},%
firstplural={\glstfirstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}%
\protect\glstrfullsep{\the\glslabeltok}%
\protect\glstrparen{\glstfirstxplongfont{\the\glslongpltok}{\glscategorylabel}}},%
text={\glstxpabrvfont{\the\glsshorttok}{\glscategorylabel}}},%
plural={\glstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
}%

```

Unset the regular attribute if it has been set.

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glstrsetcomplexstyle{\the\glslabeltok}{3}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{\%
\GlsXtrUseAbbrStyleFmts{short-em-long-em}%
}

```

short-em

```

\newabbreviationstyle{short-em}%
{%

```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
name={\glstrshortnolongname},
sort={\the\glsshorttok},
first={\glstfirstxpabrvfont{\the\glsshorttok}{\glscategorylabel}},
firstplural={\glstfirstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}},
text={\glstxpabrvfont{\the\glsshorttok}{\glscategorylabel}},
plural={\glstxpabrvfont{\the\glsshortpltok}{\glscategorylabel}},
description={\the\glslongtok}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glissetattribute{\the\glslabeltok}{regular}{true}}%
}%
}%

\renewcommand*{\abbrvpluralsuffix}{\glstxtremsuffix}%
\renewcommand*{\glstrrevert[1]}{\glstxtremrevert{##1}}%
\renewcommand*{\glstabrvfont[1]}{\glstabrvfont{##1}}%
\renewcommand*{\glstfirstabrvfont}[1]{\glstfirstabrvfont{##1}}%
\renewcommand*{\glstfirstlongfont}[1]{\glstfirstlongdefaultfont{##1}}%
\renewcommand*{\glstlongfont}[1]{\glstlongdefaultfont{##1}}%

```

The inline full form displays the short form followed by the long form in parentheses.

```

\renewcommand*{\glxtrinlinefullformat}[2]{%
  \glxtrshortlongformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glxtrshortlongplformat{##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*{\glxtrfullformat}[2]{%
  \glxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrshortplformat{##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 `\glstrshortdescname` could be modified to omit the long form, so include the `nameshortaccess` attribute.

```
\glstrAccSuppAbbrSetNoLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrshortdescname},
  sort={\the\glsshorttok},
  first={\glstxppabrvfont{\the\glsshorttok}{\glscategorylabel}},
  firstplural={\glstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}},
  text={\glstxppabrvfont{\the\glsshorttok}{\glscategorylabel}},
  plural={\glstxppabrvfont{\the\glsshortpltok}{\glscategorylabel}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%
```

```
\renewcommand*{\abbrvpluralsuffix}{\glstremsuffix}%
\renewcommand*{\glstxtrrevert}[1]{\glstxtrrevert{##1}}%
\renewcommand*{\glstxtrrevert}[1]{\glstxtrrevert{##1}}%
\renewcommand*{\glstxtrrevert}[1]{\glstxtrrevert{##1}}%
\renewcommand*{\glstxtrrevert}[1]{\glstxtrrevert{##1}}%
\renewcommand*{\glstxtrrevert}[1]{\glstxtrrevert{##1}}%
\renewcommand*{\glstxtrrevert}[1]{\glstxtrrevert{##1}}%
```

The inline full form displays the short format followed by the long form in parentheses.

```
\renewcommand*{\glstxtrinlinefullformat}[2]{%
  \glstxtrshortlongformat{##1}{##2}%
  {\glstxtrshortlongdefaultfont}{\glstxtrshortlongfont}}%
}%
\renewcommand*{\glstxtrinlinefullplformat}[2]{%
  \glstxtrshortlongplformat{##1}{##2}%
  {\glstxtrshortlongdefaultfont}{\glstxtrshortlongfont}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortlongformat{##1}{##2}%
  {\glstxtrshortlongdefaultfont}{\glstxtrshortlongfont}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshortlongplformat{##1}{##2}%
  {\glstxtrshortlongdefaultfont}{\glstxtrshortlongfont}}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortlongformat{##1}{##2}%
  {\glstxtrshortlongdefaultfont}{\glstxtrshortlongfont}}%
}%
```

```

\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]{%
  \glsxtrshorttplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshorttplformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshorttplformat{##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]{%
  \glsxtrlongshorttplformat{##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\glssshorttok},
    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:

```
\@glstr@deprecated@abbrstyle{long-em}{long-noshort-em}
```

long-em-noshort-em The short form is explicitly invoked through commands like `\glstrshort`.

```

\newabbreviationstyle{long-em-noshort-em}%
{%

```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrlongnoshortname},
sort={\the\glsshorttok},
first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},
firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},
text={\glsxplongfont{\the\glslongtok}{\glscategorylabel}},
plural={\glsxplongfont{\the\glslongpltok}{\glscategorylabel}},%
description={\protect\glslongemfont{\the\glslongtok}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
\glsetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%

```

```

\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*{\glxtrrevert}[1]{\glxtremrevert{##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*{\glstrsubsequentfmt}[2]{%
\glxtrlongformat{##1}{##2}{\glslongemfont}%
}%
\renewcommand*{\glstrsubsequentplfmt}[2]{%

```

```

\glxstrlongplformat{##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*{\glxtrinlinefullformat}[2]{%
\glxtrlongshortformat{##1}{##2}%
{\glsfirstlongemfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glxtrlongshortplformat{##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*{\glxtrfullformat}[2]{%
\glxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glxtrlongplformat{##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}%
    {%
      \glsssetattribute{\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*\glxstrsubsequentfmt}[2]{%
  \glxstrlongformat{##1}{##2}{\glslongdefaultfont}%
}%
\renewcommand*\glxstrsubsequentplfmt}[2]{%
  \glxstrlongplformat{##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*\glxstrinlinefullformat}[2]{%
  \glxstrlongshortformat{##1}{##2}%
  {\glsfirstlongdefaultfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*\glxstrinlinefullplformat}[2]{%
  \glxstrlongshortplformat{##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*\glxstrfullformat}[2]{%
```

```

\glxstrlongformat{##1}{##2}{\glsfirstlongdefaultfont}%
}%
\renewcommand*{\glxstrfullplformat}[2]{%
\glxstrlongplformat{##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:

```
\@glxstr@deprecated@abbrstyle{long-desc-em}{long-noshort-em-desc}
```

long-em-noshort-em-desc The short form is explicitly invoked through commands like `\glxstrshort`. The long form is emphasized. No accessibility attributes need to be set.

```

\newabbreviationstyle{long-em-noshort-em-desc}%
{%
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxstrlongnoshortdescname},
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}{%
\glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
}%

\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*{\glxstrrevert}[1]{\glxtremrevert{##1}}%
\renewcommand*{\glsabrvfont}[1]{\glsabrvfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvfont{##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*{\glxstrsubsequentfmt}[2]{%
\glxstrlongformat{##1}{##2}{\glslongemfont}%
}%

```

```

\renewcommand*{\glxtrsubsequentplfmt}[2]{%
  \glxtrlongplformat{##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*{\glxtrininlinefullformat}[2]{%
  \glxtrlongshortformat{##1}{##2}%
  {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glxtrininlinefullplformat}[2]{%
  \glxtrlongshorttplformat{##1}{##2}%
  {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrininlinefullformat}[2]{%
  \Glsxtrlongshortformat{##1}{##2}%
  {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\Glsxtrininlinefullplformat}[2]{%
  \Glsxtrlongshorttplformat{##1}{##2}%
  {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrininlinefullformat}[2]{%
  \GLSxtrlongshortformat{##1}{##2}%
  {\glsfirstlongemfont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\GLSxtrininlinefullplformat}[2]{%
  \GLSxtrlongshorttplformat{##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*{\glxtrfullformat}[2]{%
  \glxtrlongformat{##1}{##2}{\glsfirstlongemfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrlongplformat{##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}%
  {%
    \glsssetAttribute{\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={\glxppabrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxppabrvfont{\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}{%
\glsssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{%

\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*{\glxtrrevert}[1]{\glxtremrevert{##1}}%
\renewcommand*{\glsabrvfont}[1]{\glsabrvfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvfont{##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*{\glxtrfullformat}[2]{%
\glxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
\protect\glxtrabbrvfootnote{##1}%
{\glxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
\protect\glxtrabbrvfootnote{##1}%
{\glxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\Glsxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
\protect\glxtrabbrvfootnote{##1}%
{\glxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\Glsxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%
\protect\glxtrabbrvfootnote{##1}%
{\glxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
\GLSxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
\protect\glxtrabbrvfootnote{##1}%
{\glxtrfootnotelongformat{##1}{\glsfirstlongfootnotefont}}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
\GLSxtrshortplformat{##1}{##2}{\glsfirstabbrvfont}%

```



```

\protect\glxtrabbrvfootnote{##1}%
{\glxtrfootnotelongplformat{##1}{\glsfirstlongfootnotefont}}}%
}%

```

The first use inline full form uses the short (long) style.

```

\renewcommand*{\glxtrinlinefullformat}[2]{%
\glxtrshortlongformat{##1}{##2}%
{\glsfirstlongfootnotefont}{\glsfirstabbrvemfont}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glxtrshortlongplformat{##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:

```

\@glxtr@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.

```

\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel

```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrfootnotedesname},
sort={\glxtrfootnotedesort},
first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}%
\protect\glxtrabbrvfootnote{\the\glslabeltok}%
{\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}%
\protect\glxtrabbrvfootnote{\the\glslabeltok}%
{\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%

```

```

text={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\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}{%
\glsssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\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={\glxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
plural={\glxpabbrvfont{\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 `glsxtrsetupfuldefs` 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}%
{%

```

```

\glsssetAttribute{\the\glslabeltok}{regular}{false}%
}%
{}%
}%
}%
{%

\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*{\glxtrrevert}[1]{\glxtremrevert{##1}}%
\renewcommand*{\glsabrvfont}[1]{\glsabrvfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%

```

The full format displays the short form. The long form is deferred.

```

\renewcommand*{\glxtrfullformat}[2]{%
\glxtrshortformat{##1}{##2}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glxtrshortplformat{##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 inline full form uses the short (long) style.

```

\renewcommand*{\glxtrininlinefullformat}[2]{%
\glxtrshortlongformat{##1}{##2}%
{\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\glxtrininlinefullplformat}[2]{%
\glxtrshortlongplformat{##1}{##2}%
{\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrininlinefullformat}[2]{%
\Glsxtrshortlongformat{##1}{##2}%
{\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%
\renewcommand*{\Glsxtrininlinefullplformat}[2]{%
\Glsxtrshortlongplformat{##1}{##2}%
{\glsfirstlongfootnotefont}{\glsfirstabbrvfont}%
}%

```

```

\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={\glsxtrfootnotedesname},
  sort={\glsxtrfootnotedesort},
  first={\glsfirstxpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  firstplural={\glsfirstxpabbrvfont{\the\glsshortpltok}{\glscategorylabel}},%
  text={\glsexpabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glsexpabbrvfont{\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 `glxtrsetupfuldefs` but that interferes with other styles. Instead, this now uses `\glxtrifwasglslike`

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glxtrpostlink\glscategorylabel}{%
    \glxtrifwasglslikeandfirstuse
    {%
      \glxtrdopostpunc{\xpglsxtrpostabbrvfootnote}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\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:

`\glxtruserfield` Default is the `useri` field.

```
\newcommand*{\glxtruserfield}{useri}
```

`\glxtruserparens` Separator used inside parenthetical content.

```
\newcommand*{\glxtruserparens}{, }
```

`\glxtruserfieldfmt` Used to format the value of the field given by `\glxtruserfield`.

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

`\glxtruserparen` 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*{\glxtruserparen}[2]{%
    \glxtrfullsep{#2}%
    \glxtrparen
    {#1\ifglshasfield{\glxtruserfield}{#2}%
      {\expandafter\glxtrgenentrytextfmt\expandafter{\glxtruserparens}%
        \glxtruserfieldfmt{\expandafter\glxtrgenentrytextfmt\expandafter{\glscurrentfieldvalue}}}%
      }{}%
    }%
  }
}
{
  \newcommand*{\glxtruserparen}[2]{%
    \glxtrfullsep{#2}%
    \glxtrparen
    {#1\ifglshasfield{\glxtruserfield}{#2}%
      {\expandafter\glxtrgenentrytextfmt\expandafter{\glxtruserparens}%
        \glxtruserfieldfmt{\expandafter\glxtrgenentrytextfmt\expandafter{\@glo@thisvalue}}}%
      }{}%
    }%
  }
}
```

`\GLSxtruserparen` As above but converts the user supplied information to all-caps. The first argument should be provided in all-caps if required.

```
\ifdef\glscurrentfieldvalue
{
  \newcommand*{\GLSxtruserparen}[2]{%
    \glxtrfullsep{#2}%
    \glxtrparen
```

```

{#1\ifglshasfield{\glxtruserfield}{#2}%
  {\expandafter\glxtrgenentrytextfmt\expandafter{\glxtruserparens}%
    \glxtruserfieldfmt{\expandafter\mfirstucMakeUppercase\expandafter{\expandafter
      \glxtrgenentrytextfmt\expandafter{\glscurrentfieldvalue}}}%
    }{}%
  }%
}
}
{
  \newcommand*{\GLSxtruserparen}[2]{%
    \glxtrfullsep{#2}%
    \glxtrparen
    {#1\ifglshasfield{\glxtruserfield}{#2}%
      {\expandafter\glxtrgenentrytextfmt\expandafter{\glxtruserparens}%
        \glxtruserfieldfmt{\expandafter\mfirstucMakeUppercase\expandafter{\expandafter
          \glxtrgenentrytextfmt\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:

```

\glxtrusersuffix
  \newcommand*{\glxtrusersuffix}{\glxtrabbrvpluralsuffix}

```

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.

```
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%  
  name={\glstrlongshortname},  
  sort={\the\glsshorttok},  
  first={\protect\glstrfirstlonguserfont{\the\glslongtok}%  
    \protect\glstruserparen{\protect\glstrfirstabbrvuserfont{\the\glsshorttok}}%  
    {\the\glslabeltok}},%  
  firstplural={\protect\glstrfirstlonguserfont{\the\glslongpltok}%  
    \protect\glstruserparen  
    {\protect\glstrfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%  
  
  text={\protect\glstrabbrvuserfont{\the\glsshorttok}},%  
  plural={\protect\glstrabbrvuserfont{\the\glsshortpltok}},%  
  description={\protect\glstruserdescription{\the\glslongtok}%  
    {\the\glslabeltok}}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%  
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%  
  \glstrsetcomplexstyle{\the\glslabeltok}{2}%  
  \glshasattribute{\the\glslabeltok}{regular}%  
  {%  
    \glissetattribute{\the\glslabeltok}{regular}{false}%  
  }%  
  {%  
  }%  
}%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrusersuffix}%  
\renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvuserfont{##1}}%  
\renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvuserfont{##1}}%  
\renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlonguserfont{##1}}%  
\renewcommand*{\glstrlongfont}[1]{\glstrlonguserfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*{\glstrfullformat}[2]{%  
  \glstruserlongshortformat{##1}{##2}%  
  {\glstrfirstlonguserfont}{\glstrfirstabbrvuserfont}%  
}%  
\renewcommand*{\glstrfullplformat}[2]{%  
  \glstruserlongshortplformat{##1}{##2}%  
  {\glstrfirstlonguserfont}{\glstrfirstabbrvuserfont}%  
}%
```

```

\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{\glxtrusersuffix}%
\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*\glxtrfullformat}[2]{%
  \glxtrlongformat{##1}{##2}{\glsfirstlonguserfont}%
}%
\renewcommand*\glxtrfullplformat}[2]{%
  \glxtrlongplformat{##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:

```
\glxtrscusersuffix
\newcommand*\glxtrscusersuffix{\glxtrscsuffix}

\glxtrscuserrevert
\newcommand*\glxtrscuserrevert{\glxtrscscrevert}
```

\glxtrlongshortscusername The default name format for this style.

```
\newcommand*\glxtrlongshortscusername{%
  \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.

```
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrlongshortscusername},
  sort={\the\glsshorttok},
  first={\protect\glstrfirstlonguserfont{\the\glslongtok}},%
  firstplural={\protect\glstrfirstlonguserfont{\the\glslongpltok}},%
  text={\protect\glstrabbrvcuserfont{\the\glsshorttok}},%
  plural={\protect\glstrabbrvcuserfont{\the\glsshortpltok}},%
  description={\protect\glstruserdescription{\the\glslongtok}%
    {\the\glslabeltok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
  \csdef{glstrpostlink\glscategorylabel}{%
    \glstrifwasfirstuse
    {%
      \glstrpostusershortformat{\glslabell}{\glstrfirstabbrvcuserfont}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glstrscusersuffix}%
\renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvcuserfont{##1}}%
\renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvcuserfont{##1}}%
\renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlonguserfont{##1}}%
\renewcommand*{\glstrlongfont}[1]{\glstrlonguserfont{##1}}%
\renewcommand*{\glstrrevert}[1]{\glstrscuserrevert{##1}}%
```

First use full form:

```
\renewcommand*{\glstrfullformat}[2]{%
  \glstrlongformat{##1}{##2}{\glstrfirstlonguserfont}%
}%
\renewcommand*{\glstrfullplformat}[2]{%
  \glstrlongplformat{##1}{##2}{\glstrfirstlonguserfont}%
}%
\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:

```

\renewcommand*{\glsxtrinlinefullformat}[2]{%
\glsxtruserlongshortformat{##1}{##2}%
{\glsfirstlonguserfont}{\glsfirstabbrvscuserfont}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
\glsxtruserlongshortplformat{##1}{##2}%
{\glsfirstlonguserfont}{\glsfirstabbrvscuserfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
\GLSxtruserlongshortformat{##1}{##2}%
{\glsfirstlonguserfont}{\glsfirstabbrvscuserfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
\GLSxtruserlongshortplformat{##1}{##2}%
{\glsfirstlonguserfont}{\glsfirstabbrvscuserfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
\GLSxtruserlongshortformat{##1}{##2}%
{\glsfirstlonguserfont}{\glsfirstabbrvscuserfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
\GLSxtruserlongshortplformat{##1}{##2}%
{\glsfirstlonguserfont}{\glsfirstabbrvscuserfont}%
}%
}

```

`\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}%
{%

```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrlongshortuserdescname},
  sort={\the\glslongtok},
  first={\protect\glstrfirstlonguserfont{\the\glslongtok}},%
  firstplural={\protect\glstrfirstlonguserfont{\the\glslongpltok}},%

  text={\protect\glstrabbrvuserfont{\the\glsshorttok}},%
  plural={\protect\glstrabbrvuserfont{\the\glsshortpltok}}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \csdef{glstrpostlink\glscategorylabel}{%
    \glstrifwasfirstuse
    {%
      \glstrpostusershortformat{\glslabel}{\glstrfirstabbrvuserfont}%
    }%
    {}%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-postshort-user}%
}
```

glstrlongshortscuserdescname

```
\newcommand*{\glstrlongshortscuserdescname}{%
  \protect\glslonguserfont{\the\glslongtok}%
  \protect\glstruserparen
  {\protect\glstrabbrvscuserfont{\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.

```
\glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrlongshortscuserdescname},
  sort={\the\glslongtok},
  first={\protect\glstrfirstlonguserfont{\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}%
{%
\glissetattribute{\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}%

```

```

    {%
      \glsetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%

```

In case the user wants to mix and match font styles, these are redefined here.

```

\renewcommand*{\abbrvpluralsuffix}{\glxtrusersuffix}%
\renewcommand*{\glabbrvfont}[1]{\glabbrvuserfont{##1}}%
\renewcommand*{\glfirstabbrvfont}[1]{\glfirstabbrvuserfont{##1}}%
\renewcommand*{\glfirstlongfont}[1]{\glfirstlonguserfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%

```

First use full form:

```

\renewcommand*{\glxtrfullformat}[2]{%
  \glxtrshortformat{##1}{##2}{\glfirstabbrvuserfont}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrshortplformat{##1}{##2}{\glfirstabbrvuserfont}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \Glsxtrshortformat{##1}{##2}{\glfirstabbrvuserfont}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \Glsxtrshortplformat{##1}{##2}{\glfirstabbrvuserfont}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
  \GLSxtrshortformat{##1}{##2}{\glfirstabbrvuserfont}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
  \GLSxtrshortplformat{##1}{##2}{\glfirstabbrvuserfont}%
}%

```

In-line format should be the same.

```

}

```

`\glxtrshortlonguserdescname`

```

\newcommand*{\glxtrshortlonguserdescname}{%
  \protect\glabbrvuserfont{\the\glsshorttok}%
  \protect\glxtruserparen
    {\protect\glslonguserfont{\the\glslongpltok}}%
    {\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.

```

\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel

```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrshortlonguserdescname},
  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{glxtrpostlink\glscategorylabel}{%
    \glxtrifwasfirstuse
    {%
      \glxtrpostuserlongformat{\glslabel}{\glsfirstlonguserfont}%
    }%
  }%
  {}%
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
  \glissetattribute{\the\glslabeltok}{regular}{false}%
}%
{}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-postlong-user}%
}
```

long-short-user-desc

```
\newabbreviationstyle{long-short-user-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrlongshortuserdescname},
  sort={\glxtrlongshortdescsort},%

  first={\protect\glsfirstlonguserfont{\the\glslongtok}%
    \protect\glxtruserparen{\protect\glsfirstabbrvuserfont{\the\glsshorttok}}%
    {\the\glslabeltok}},%
  firstplural={\protect\glsfirstlonguserfont{\the\glslongpltok}%
    \protect\glxtruserparen
    {\protect\glsfirstabbrvuserfont{\the\glsshortpltok}}{\the\glslabeltok}},%
  text={\protect\glsabbrvfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
}%
```

Unset the `regular` attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glstrsetcomplexstyle{\the\glslabeltok}{2}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
}%
{}%
```

`short-long-user`

```
\newabbreviationstyle{short-long-user}%
{%
```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

`\glslonguserfont` is used in the description since `\glsdesc` doesn't set the style. (Now in `\glsuserdescription`.)

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrshortlongname},
  sort={\the\glsshorttok},
  description={\protect\glsuserdescription{\the\glslongtok}%
    {\the\glslabeltok}},%
  first={\protect\glsfirstabbrvuserfont{\the\glsshorttok}%
    \protect\glstruserparen{\protect\glsfirstlonguserfont{\the\glslongtok}}%
    {\the\glslabeltok}},%
  firstplural={\protect\glsfirstabbrvuserfont{\the\glsshortpltok}%
    \protect\glstruserparen{\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}%
  \glstrsetcomplexstyle{\the\glslabeltok}{2}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
}%
{}%
```


In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*\abbrvpluralsuffix{\glxtrusersuffix}%
\renewcommand*\glabbrvfont[1]{\glabbrvuserfont{##1}}%
\renewcommand*\glfirstabbrvfont[1]{\glfirstabbrvuserfont{##1}}%
\renewcommand*\glfirstlongfont[1]{\glfirstlonguserfont{##1}}%
\renewcommand*\glslongfont[1]{\glslonguserfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*\glxtrfullformat}[2]{%
  \glxtrusershortlongformat{##1}{##2}%
  {\glfirstlonguserfont}{\glfirstabbrvuserfont}}%
}%
\renewcommand*\glxtrfullplformat}[2]{%
  \glxtrusershortlongplformat{##1}{##2}%
  {\glfirstlonguserfont}{\glfirstabbrvuserfont}}%
}%
\renewcommand*\Glsxtrfullformat}[2]{%
  \Glsxtrusershortlongformat{##1}{##2}%
  {\glfirstlonguserfont}{\glfirstabbrvuserfont}}%
}%
\renewcommand*\Glsxtrfullplformat}[2]{%
  \Glsxtrusershortlongplformat{##1}{##2}%
  {\glfirstlonguserfont}{\glfirstabbrvuserfont}}%
}%
\renewcommand*\GLSxtrfullformat}[2]{%
  \GLSxtrusershortlongformat{##1}{##2}%
  {\glfirstlonguserfont}{\glfirstabbrvuserfont}}%
}%
\renewcommand*\GLSxtrfullplformat}[2]{%
  \GLSxtrusershortlongplformat{##1}{##2}%
  {\glfirstlonguserfont}{\glfirstabbrvuserfont}}%
}%
}
```

short-long-user-desc

```
\newabbreviationstyle{short-long-user-desc}%
{%
```

Set accessibility attributes if enabled.

```
\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*\CustomAbbreviationFields{%
  name={\glxtrshortlonguserdescname},
  sort={\glxtrshortlongdescsort},%

  first={\protect\glfirstabbrvuserfont{\the\glsshorttok}}%
  \protect\glxtruserparen{\protect\glfirstlonguserfont{\the\glslongtok}}%
  {\the\glslabeltok}},%
  firstplural={\protect\glfirstabbrvuserfont{\the\glsshortpltok}}%
  \protect\glxtruserparen{\protect\glfirstlonguserfont{\the\glslongpltok}}%
```

```

        {\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}%
\glsattribute{\the\glslabeltok}{regular}%
{%
\glsattribute{\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
}

```

`\glsxtrlonghyphenstart`

```

\glsxtrlonghyphenstart{\label}{\long}{\short}{\insert}

```

The `\long` and `\short` arguments may be the plural form. The `\long` argument may also be the first letter uppercase form. This unfortunately doesn't

fit in with the new `\glxtrshortformat` 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*{\glxtrlonghyphenshort}[4]{%
```

Grouping is needed to localise the redefinitions.

```
{%
```

If `<insert>` starts with a hyphen, redefine `\glxtrwordsep` to a hyphen. The inserted material is also inserted into the parenthetical part. (The inserted material is grouped as a precautionary measure.) No change is made to `\glxtrwordsep` if `<insert>` doesn't start with a hyphen.

```
\glxtrifhyphenstart{#4}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
\glxfirstlonghyphenfont{#2\ifglxtrininsertinside
  {\glxtrgenentrytextfmt{#4}}\fi}%
\ifglxtrininsertinside\else{\glxtrgenentrytextfmt{#4}}\fi
\glxtrfullsep{#1}%
\glxtrparen{\glxfirstabbrvhyphenfont{#3\ifglxtrininsertinside
  {\glxtrgenentrytextfmt{#4}}\fi}%
\ifglxtrininsertinside\else{\glxtrgenentrytextfmt{#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]{%
  {%
    \glxtrifhyphenstart{#4}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
    \glxfirstlonghyphenfont{#2\ifglxtrininsertinside
      {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#4}}}\fi}%
    \ifglxtrininsertinside\else
      {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#4}}}%
    \fi
    \glxtrfullsep{#1}%
    \glxtrparen{\glxfirstabbrvhyphenfont{#3\ifglxtrininsertinside
      {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#4}}}\fi}%
    \ifglxtrininsertinside\else
      {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#4}}}%
    \fi}%
  }%
}
```

```
\glxtrshorthyphennolong{<label>}{<short>}{<insert>}
```

`\glxtrshorthyphennolong`

The `<short>` argument may be the plural form and may also be the first letter uppercase form.

As `\glxtrlonghyphenshort` but where only the short form should be shown.

```
\newcommand*{\glxtrshorthyphennolong}[3]{%
```

Grouping is needed to localise the redefinitions.

```
{%
```

If *⟨insert⟩* starts with a hyphen, redefine `\glxtrwordsep` to a hyphen.

```
\glxtrifhyphenstart{#3}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
```

```
\glsfirstabbrvhyphenfont{#2\ifglxtrininsertinside
```

```
{\glxtrgenentrytextfmt{#3}}\fi}%
```

```
\ifglxtrininsertinside\else{\glxtrgenentrytextfmt{#3}}\fi
```

```
}%
```

```
}
```

`\GLSxtrshorthyphennolong` As above but all-caps.

```
\newcommand*{\GLSxtrshorthyphennolong}[3]{%
```

```
{%
```

```
\glxtrifhyphenstart{#3}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
```

```
\glsfirstabbrvhyphenfont{#2\ifglxtrininsertinside
```

```
{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#3}}}\fi}%
```

```
\ifglxtrininsertinside\else
```

```
{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#3}}}%
```

```
\fi
```

```
}%
```

```
}
```

`\glsabbrvhyphenfont`

```
\newcommand*{\glsabbrvhyphenfont}{\glsabbrvdefaultfont}%
```

`\glsfirstabbrvhyphenfont`

```
\newcommand*{\glsfirstabbrvhyphenfont}{\glsabbrvhyphenfont}%
```

`\glslonghyphenfont`

```
\newcommand*{\glslonghyphenfont}{\glslongdefaultfont}%
```

`\glsfirstlonghyphenfont`

```
\newcommand*{\glsfirstlonghyphenfont}{\glslonghyphenfont}%
```

The default short form suffix:

`\glxtrhyphensuffix`

```
\newcommand*{\glxtrhyphensuffix}{\glxtrabbrvpluralsuffix}
```

`\glxtrlonghyphensortsort`

```
\newcommand*{\glxtrlonghyphensortsort}{\expandonce\glxtrorgshort}
```

`long-hyphen-short-hyphen` Designed for use with the `markwords` attribute.

```
\newabbreviationstyle{long-hyphen-short-hyphen}%
```

```
{%
```

Set accessibility attributes if enabled.

```
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glstrlongshortname},
  sort={\glstrlonghyphensort},
  first={\protect\glstrfirstlonghyphenfont{\the\glslongtok}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\protect\glstrfirstabbrvhyphenfont{\the\glsshorttok}}},%
  firstplural={\protect\glstrfirstlonghyphenfont{\the\glslongpltok}%
    \protect\glstrfullsep{\the\glslabeltok}%
    \protect\glstrparen{\protect\glstrfirstabbrvhyphenfont{\the\glsshortpltok}}},%
  text={\protect\glstrabbrvhyphenfont{\the\glsshorttok}},%
  plural={\protect\glstrabbrvhyphenfont{\the\glsshortpltok}},%
  description={\protect\glstrlonghyphenfont{\the\glslongtok}}}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glsexclapplyinnerfmtfield{\the\glslabeltok}{desc}%
  \glstrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glstrhyphensuffix}%
  \renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvhyphenfont{##1}}%
  \renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlonghyphenfont{##1}}%
  \renewcommand*{\glstrlongfont}[1]{\glstrlonghyphenfont{##1}}%
```

The first use full form and the inline full form are the same for this style.

```
\renewcommand*{\glstrfullformat}[2]{%
  \glstrlonghyphenshort{##1}%
  {%
    \glusifattribute{##1}{markwords}{true}%
  }%
  \glssaccesslong{##1}%
  }%
  {%
    \glssaccessfmtlong{\glstrgenentrytextfmt}{##1}%
  }%
  {%
    \glusifattribute{##1}{markshortwords}{true}%
  }%
  \glssaccessshort{##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{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
        \GLSaccessshortpl{##1}%
    }%
    {%
        \GLSaccessfmtshortpl{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{##2}%
}%

```

Subsequent form also needs checking for a hyphen in case the short form has spaces.

```

\renewcommand*{\glstrsubsequentfmt}[2]{%
    \glstrshorthypennolong{##1}%
    {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
            \glsaccessshort{##1}%
        }%
        {%
            \glsaccessfmtshort{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*{\glstrsubsequentplfmt}[2]{%
    \glstrshorthypennolong{##1}%
    {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
            \glsaccessshortpl{##1}%
        }%
        {%
            \glsaccessfmtshortpl{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*{\Glsstrsubsequentfmt}[2]{%
    \glstrshorthypennolong{##1}%
    {%
        \glsifattribute{##1}{markshortwords}{true}%
    }%
}

```



```

        {%
            \Glsaccessshort{##1}%
        }%
        {%
            \Glsaccessfmtshort{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glstrshorthyphennolong{##1}%
    {%
        \glrifattribute{##1}{markshortwords}{true}%
        {%
            \Glsaccessshortpl{##1}%
        }%
        {%
            \Glsaccessfmtshortpl{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
    \GLSxtrshorthyphennolong{##1}%
    {%
        \glrifattribute{##1}{markshortwords}{true}%
        {%
            \GLSaccessshort{##1}%
        }%
        {%
            \GLSaccessfmtshort{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
    \GLSxtrshorthyphennolong{##1}%
    {%
        \glrifattribute{##1}{markshortwords}{true}%
        {%
            \GLSaccessshortpl{##1}%
        }%
        {%
            \GLSaccessfmtshortpl{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
}

```

long-hyphen-short-hyphen-desc 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.

```
\glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%  
  name={\glstrlongshortdescname},  
  sort={\glstrlongshortdescsort},  
  first={\protect\glstrfirstlonghyphenfont{\the\glstrlongtok}%  
    \protect\glstrfullsep{\the\glstrlabeltok}%  
    \protect\glstrparen{\protect\glstrfirstabbrhyphenfont{\the\glstrshorttok}}},%  
  firstplural={\protect\glstrfirstlonghyphenfont{\the\glstrlongpltok}%  
    \protect\glstrfullsep{\the\glstrlabeltok}%  
    \protect\glstrparen{\protect\glstrfirstabbrhyphenfont{\the\glstrshortpltok}}},%  
  text={\protect\glstrabbrhyphenfont{\the\glstrshorttok}},%  
  plural={\protect\glstrabbrhyphenfont{\the\glstrshortpltok}}%  
}%
```

Unset the regular attribute if it has been set.

```
\renewcommand*{\GlsXtrPostNewAbbreviation}{%  
  \glstrsetcomplexstyle{\the\glstrlabeltok}{3}%  
  \glshasattribute{\the\glstrlabeltok}{regular}%  
  {%  
    \glsssetAttribute{\the\glstrlabeltok}{regular}{false}%  
  }%  
  {}%  
}%  
}%  
{%  
  \GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%  
}
```

```
\glstrlonghyphennoshort{<label>}{<long>}{<insert>}
```

\glstrlonghyphennoshort

As with \glstrlonghyphenshort this doesn't fit in with the new \glstrshortformat 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*{\glstrlonghyphennoshort}[3]{%
```

Grouping is needed to localise the redefinitions.

```
{%
```

If <insert> starts with a hyphen, redefine \glstrwordsep 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 \glstrwordsep if <insert> doesn't start with a hyphen.

```

\glxtrifhyphenstart{#3}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
\glsfirstlonghyphenfont{#2\ifglxtrininsertinside
  {\glxtrgenentrytextfmt{#3}}\fi}%
\ifglxtrininsertinside\else{\glxtrgenentrytextfmt{#3}}\fi
}%
}

```

`\GLSxtrlonghyphennoshort` As above but convert insert to all-caps.

```

\newcommand*{\GLSxtrlonghyphennoshort}[3]{%
  {%
    \glxtrifhyphenstart{#3}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
    \glsfirstlonghyphenfont{#2\ifglxtrininsertinside
      {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#3}}}\fi}%
    \ifglxtrininsertinside\else
      {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#3}}}%
    \fi
  }%
}

```

`\glxtrlonghyphennoshortdescsort`

```

\newcommand*{\glxtrlonghyphennoshortdescsort}{\expandonce\glxtrorlong}

```

`long-hyphen-noshort-desc-noreg` This version doesn't show the short form (except explicitly with `\glxtrshort`). Since `\glxtrshort` doesn't support the hyphen switch, the short form just uses the default short-form font command. This style won't work with the regular as the regular form isn't flexible enough. No accessibility attributes need to be set.

```

\newabbreviationstyle{long-hyphen-noshort-desc-noreg}{%
  {%
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glxtrlongnoshortdescname},
      sort={\glxtrlonghyphennoshortdescsort},
      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}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
{}%
{}%

```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*\abbrvpluralsuffix{\glxtrabbrvpluralsuffix}%
\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{}\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*\glstrfullplformat}[2]{%
    \glstrlonghyphennoshort{##1}%
    {%
        \glsifattribute{##1}{keywords}{true}%
        {%
            \glsaccesslongpl{##1}%
        }%
        {%
            \glsaccessfmtlongpl{}\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*\Glsxtrfullformat}[2]{%
    \glstrlonghyphennoshort{##1}%
    {%
        \glsifattribute{##1}{keywords}{true}%
        {%
            \Glsaccesslong{##1}%
        }%
        {%
            \Glsaccessfmtlong{}\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*\Glsxtrfullplformat}[2]{%
    \glstrlonghyphennoshort{##1}%
    {%
        \glsifattribute{##1}{keywords}{true}%
        {%
            \Glsaccesslongpl{##1}%
        }%
        {%
            \Glsaccessfmtlongpl{}\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##2}%
}%
\renewcommand*\GLSxtrfullformat}[2]{%
    \GLSxtrlonghyphennoshort{##1}%
    {%
        \glsifattribute{##1}{keywords}{true}%
        {%

```

```

        \GLSaccesslong{##1}%
    }%
    {%
        \GLSaccessfmtlong{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{##2}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
    \GLSxtrlonghyphennoshort{##1}%
    {%
        \glrifattribute{##1}{keywords}{true}%
    }%
        \GLSaccesslongpl{##1}%
    }%
    {%
        \GLSaccessfmtlongpl{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{##2}%
}%

```

The format for subsequent use (not used when the regular attribute is set).

```

\renewcommand*{\glstrsubsequentfmt}[2]{%
    \glstrlonghyphennoshort{##1}%
    {%
        \glrifattribute{##1}{keywords}{true}%
    }%
        \glaccesslong{##1}%
    }%
    {%
        \glaccessfmtlong{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{##2}%
}%
\renewcommand*{\glstrsubsequentplfmt}[2]{%
    \glstrlonghyphennoshort{##1}%
    {%
        \glrifattribute{##1}{keywords}{true}%
    }%
        \glaccesslongpl{##1}%
    }%
    {%
        \glaccessfmtlongpl{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%

```



```

\glxstrlonghyphennoshort{##1}%
{%
  \glrifattribute{##1}{keywords}{true}%
  {%
    \Glsaccesslong{##1}%
  }%
  {%
    \Glsaccessfmtlong{}{\glxstrgenentrytextfmt}{##1}%
  }%
}%
{##2}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glxstrlonghyphennoshort{##1}%
  {%
    \glrifattribute{##1}{keywords}{true}%
    {%
      \Glsaccesslongpl{##1}%
    }%
    {%
      \Glsaccessfmtlongpl{}{\glxstrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \GLSxtrlonghyphennoshort{##1}%
  {%
    \glrifattribute{##1}{keywords}{true}%
    {%
      \GLSaccesslong{##1}%
    }%
    {%
      \GLSaccessfmtlong{}{\glxstrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
  \GLSxtrlonghyphennoshort{##1}%
  {%
    \glrifattribute{##1}{keywords}{true}%
    {%
      \GLSaccesslongpl{##1}%
    }%
    {%
      \GLSaccessfmtlongpl{}{\glxstrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}

```

```
}%
}
```

`\glxtrlonghyphennoshortsort`

```
\newcommand*{\glxtrlonghyphennoshortsort}{\expandonce\glxtrorgshort}
```

`long-hyphen-noshort-noreg`

It doesn't really make a great deal of sense to have a long-only style that doesn't have a description (unless no glossary is required), but the best course of action here is to use the short form as the name and the long form as the description.

```
\newabbreviationstyle{long-hyphen-noshort-noreg}{%
{%
```

Set accessibility attributes if enabled.

```
\glxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrlongnoshortname},
sort={\glxtrlonghyphennoshortsort},
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}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{%
```

```
\GlsXtrUseAbbrStyleFmts{long-hyphen-noshort-desc-noreg}%
}
```

`\glxtrlonghyphen`

```
\glxtrlonghyphen{<long>}{<label>}{<insert>}
```

Used by `long-hyphen-postshort-hyphen`. The *<insert>* is check to determine if it starts with a hyphen but isn't used here as it's moved to the post-link hook.

The *<long>* argument will need to include the inner format.

```
\newcommand*{\glxtrlonghyphen}[3]{%
```

Grouping is needed to localise the redefinitions.

```
{%
```

```

\glxtrifhyphenstart{#3}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
\glsfirstlonghyphenfont{#1}%
}%
}

```

```
\glxtrposthyphenshort{<label>}{<insert>}
```

\glxtrposthyphenshort

Used in the post-link hook for the long-hyphen-postshort-hyphen style. Much like \glxtrlonghyphenshort but omits the *<long>* part. This always uses the singular short form.

```

\newcommand*{\glxtrposthyphenshort}[2]{%
{%
\glxtrifhyphenstart{#2}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
\ifglxtrininsertinside
{\glsfirstlonghyphenfont{\glxtrgenentrytextfmt{#2}}}%
\else
{\glxtrgenentrytextfmt{#2}}%
\fi
\glxtrfullsep{#1}%
\glxtrparen
{%
\glxtrshortformat{#1}{#2}{\glsfirstabbrvhyphenfont}%
}%
}%
}

```

\GLSxtrposthyphenshort As above but all caps.

```

\newcommand*{\GLSxtrposthyphenshort}[2]{%
{%
\glxtrifhyphenstart{#2}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
\ifglxtrininsertinside
{\glsfirstlonghyphenfont{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}}}%
\else
{\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#2}}}%
\fi
\glxtrfullsep{#1}%
\glxtrparen
{%
\GLSxtrshortformat{#1}{#2}{\glsfirstabbrvhyphenfont}%
}%
}%
}

```

\glxtrposthyphenshortpl As above but plural.

```

\newcommand*{\glxtrposthyphenshortpl}[2]{%
{%
\glxtrifhyphenstart{#2}{\let\glxtrwordsep\glxtrwordsephyphen}{}%
\ifglxtrininsertinside

```

```

        {\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}{}%
        \ifglsxtrininsertinside
            {\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]{%
    \ifglsxtrininsertinside

```

```

\glsabbrvfont{\glsxtrgenentrytextfmt{#2}}}%
\else
  {\glsxtrgenentrytextfmt{#2}}}%
\fi
}

\GLSxtrposthyphensubsequent As above but all caps.
\newcommand*{\GLSxtrposthyphensubsequent}[2]{%
  \ifglsxtrininsertinside
    \glsabbrvfont{\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
  \else
    {\mfirstucMakeUppercase{\glsxtrgenentrytextfmt{#2}}}%
  \fi
}

\pglsxtrposthyphensubsequent Expand placeholders and check for all caps.
\newcommand*{\xpglsxtrposthyphensubsequent}{%
  \glsxtrifallcaps
  {%
    \expandafter\GLSxtrposthyphensubsequent\expandafter\glslabel
    \expandafter{\glsinsert}%
  }%
  {%
    \expandafter\glsxtrposthyphensubsequent\expandafter\glslabel
    \expandafter{\glsinsert}%
  }%
}

\glsxtrshorthyphennoinsert
\glsxtrshorthyphennoinsert {\label}{\short}{\insert}
As with \glsxtrshorthyphennoinsert 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={\glxstrlongshortname},
  sort={\glxstrlonghyphensort},
  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}{%
    \glxstrifwasfirstuse
    {%
      \xpglxtrposthyphenshort
    }%
    {%
```

Put the insertion into the post-link:

```
      \xpglxtrposthyphensubsequent
    }%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
\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*{\glxtrsubsequentfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glusifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{\glxtrgenentrytextfmt}{##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}%
}

```

```

    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glrifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshortpl{##1}%
    }%
    {%
      \GLSaccessfmtshortpl{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%

```

First use full form:

```

\renewcommand*{\glxtrfullformat}[2]{%
  \glxtrlonghyphen
  {%
    \glrifattribute{##1}{markwords}{true}%
    {%
      \glaccesslong{##1}%
    }%
    {%
      \glaccessfmtlong{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##1}{##2}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrlonghyphen
  {%
    \glrifattribute{##1}{markwords}{true}%
    {%
      \glaccesslongpl{##1}%
    }%
    {%
      \glaccessfmtlongpl{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##1}{##2}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glxtrlonghyphen
  {%
    \glrifattribute{##1}{markwords}{true}%
  }%
}

```



```

        \Glsaccesslong{##1}%
    }%
    {%
        \Glsaccessfmtlong{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{##1}{##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
    \glxtrlonghyphen
    {%
        \glsifattribute{##1}{markwords}{true}%
        {%
            \Glsaccesslongpl{##1}%
        }%
        {%
            \Glsaccessfmtlongpl{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##1}{##2}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
    \glxtrlonghyphen
    {%
        \glsifattribute{##1}{markwords}{true}%
        {%
            \GLSaccesslong{##1}%
        }%
        {%
            \GLSaccessfmtlong{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##1}{##2}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
    \glxtrlonghyphen
    {%
        \glsifattribute{##1}{markwords}{true}%
        {%
            \GLSaccesslongpl{##1}%
        }%
        {%
            \GLSaccessfmtlongpl{}{\glstrgenentrytextfmt}{##1}%
        }%
    }%
    {##1}{##2}%
}%

```

In-line format.

```

\renewcommand*{\glxtrinlinefullformat}[2]{%

```

```

\glstrlongformatgrp{##1}{##2}{\glfirstlonghyphenfont}%
}%
\renewcommand*{\glstrinlinefullplformat}[2]{%
\glstrlongplformatgrp{##1}{##2}{\glfirstlonghyphenfont}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\Glsxtrlongformatgrp{##1}{##2}{\glfirstlonghyphenfont}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\Glsxtrlongplformatgrp{##1}{##2}{\glfirstlonghyphenfont}%
}%
\renewcommand*{\GLSxtrinlinefullformat}[2]{%
\GLSxtrlongformatgrp{##1}{##2}{\glfirstlonghyphenfont}%
}%
\renewcommand*{\GLSxtrinlinefullplformat}[2]{%
\GLSxtrlongplformatgrp{##1}{##2}{\glfirstlonghyphenfont}%
}%
}

```

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.

```

\glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel

```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
name={\glstrlongshortdescname},
sort={\glstrlongshortdescsort},%
first={\protect\glfirstlonghyphenfont{\the\glslongtok}},%
firstplural={\protect\glfirstlonghyphenfont{\the\glslongpltok}},%
text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\csdef{glstrpostlink\glscategorylabel}{%
\glstrifwasfirstuse
{%
\xpglstrposthyphenshort
}%
}%

```

Put the insertion into the post-link:

```

\xpglstrposthyphensubsequent
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%

```

```

    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-postshort-hyphen}%
}

```

```
\glstrshorthyphenlong{<label>}{<short>}{<long>}{<insert>}
```

`\glstrshorthyphenlong`

The `<long>` and `<short>` arguments may be the plural form. The `<long>` argument may also be the first letter uppercase form.

As with `\glstrlonghyphenshort` this doesn't fit in with the new `\glstrshortformat` 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*{\glstrshorthyphenlong}[4]{%
```

Grouping is needed to localise the redefinitions.

```
{%
```

If `<insert>` starts with a hyphen, redefine `\glstrwordsep` to a hyphen. The inserted material is also inserted into the parenthetical part. (The inserted material is grouped as a precautionary measure.)

```

  \glstrifhyphenstart{#4}{\let\glstrwordsep\glstrwordsephyphen}{}%
  \glsfirstabbrvhyphenfont{#2\ifglstrinsertinside
    {\glstrgenentrytextfmt{#4}}\fi}%
  \ifglstrinsertinside\else{\glstrgenentrytextfmt{#4}}\fi
  \glstrfullsep{#1}%
  \glstrparen{\glsfirstlonghyphenfont{#3%
    \ifglstrinsertinside{\glstrgenentrytextfmt{#4}}\fi}%
    \ifglstrinsertinside\else{\glstrgenentrytextfmt{#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]{%
  {%
    \glstrifhyphenstart{#4}{\let\glstrwordsep\glstrwordsephyphen}{}%
    \glsfirstabbrvhyphenfont{#2\ifglstrinsertinside
      {\mfirstucMakeUppercase{\glstrgenentrytextfmt{#4}}}\fi}%
    \ifglstrinsertinside\else
      {\mfirstucMakeUppercase{\glstrgenentrytextfmt{#4}}}%
    \fi
    \glstrfullsep{#1}%
    \glstrparen{\glsfirstlonghyphenfont{#3%
      \ifglstrinsertinside{\mfirstucMakeUppercase{\glstrgenentrytextfmt{#4}}}\fi}%
      \ifglstrinsertinside\else

```

```

        {\mfirstucMakeUppercase{\glxtrgenentrytextfmt{#4}}}%
    \fi}%
}%
}

```

`\glxtrshorthyphenlongsort`

```
\newcommand*{\glxtrshorthyphenlongsort}{\expandonce\glxtrorgshort}
```

`short-hyphen-long-hyphen` Designed for use with the `markwords` attribute.

```

\newabbreviationstyle{short-hyphen-long-hyphen}%
{%

```

Set accessibility attributes if enabled.

```
\glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
```

Setup the default fields.

```

\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrshortlongname},
  sort={\glxtrshorthyphenlongsort},
  first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
    \protect\glxtrfullsep{\the\glslabeltok}%
    \protect\glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
  firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
    \protect\glxtrfullsep{\the\glslabeltok}%
    \protect\glxtrparen{\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}%
  \glxtrsetcomplexstyle{\the\glslabeltok}{3}%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%

```

```

{%
  \renewcommand*{\abbrvpluralsuffix}{\glxtrhyphensuffix}%
  \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*{\glxtrfullformat}[2]{%
  \glxtrshorthyphenlong{##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}%
    }%
  }%

```

```

    }%
}%
{%
    \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{}\glstrgenentrytextfmt}{##1}%
}%
}%
{##2}%
}%
\renewcommand*\GLSxtrfullplformat}[2]{%
  \GLSxtrshorthyphenlong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshortpl{##1}%
    }%
    {%
      \GLSaccessfmtshortpl{}\glstrgenentrytextfmt}{##1}%
    }%
  }%
  {%
    \glsifattribute{##1}{markwords}{true}%
    {%
      \GLSaccesslongpl{##1}%
    }%
    {%
      \GLSaccessfmtlongpl{}\glstrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%

```

Subsequent form also needs checking for a hyphen in case the short form has spaces.

```

\renewcommand*\glstrsubsequentfmt}[2]{%
  \glstrshorthyphennolong{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{}\glstrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*\glstrsubsequentplfmt}[2]{%
  \glstrshorthyphennolong{##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}%
  \glsattribute{\the\glslabeltok}{regular}%
  {%
    \glssattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-hyphen-long-hyphen}%
}

```

\glsxtrshorthyphen

```
\glsxtrshorthyphen{<short>}{<label>}{<insert>}
```

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*{\glxstrshorthyphen}[3]{%
```

Grouping is needed to localise the redefinitions.

```
{%
  \glxstrifhyphenstart{#3}{\let\glxstrwordsep\glxstrwordsephyphen}{}%
  \glsfirstabbrvhyphenfont{#1}%
}%
}
```

```
\glxstrposthyphenlong{<label>}{<insert>}
```

`\glxstrposthyphenlong`

Used in the post-link hook for the short-hyphen-postlong-hyphen style. Much like `\glxstrshorthyphenlong` but omits the *<short>* part. This always uses the singular long form.

```
\newcommand*{\glxstrposthyphenlong}[2]{%
{%
  \glxstrifhyphenstart{#2}{\let\glxstrwordsep\glxstrwordsephyphen}{}%
  \ifglxstrinsertinside
    {\glsfirstabbrvhyphenfont{\glxstrgenentrytextfmt{#2}}}%
  \else
    {\glxstrgenentrytextfmt{#2}}%
  \fi
  \glxstrfullsep{#1}%
  \glxstrparen
    {\glxstrlongformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
}%
}
```

`\GLSxtrposthyphenlong` As above but all-caps.

```
\newcommand*{\GLSxtrposthyphenlong}[2]{%
{%
  \glxstrifhyphenstart{#2}{\let\glxstrwordsep\glxstrwordsephyphen}{}%
  \ifglxstrinsertinside
    {\glsfirstabbrvhyphenfont{\mfirstucMakeUppercase{\glxstrgenentrytextfmt{#2}}}}%
  \else
    {\mfirstucMakeUppercase{\glxstrgenentrytextfmt{#2}}}%
  \fi
  \glxstrfullsep{#1}%
  \glxstrparen
    {\GLSxtrlongformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
}%
}
```

Plural versions in case they are required.

`\glxstrposthyphenlongpl`

```

\newcommand*{\glxstrposthyphenlongpl}[2]{%
  {%
    \glxstrifhyphenstart{#2}{\let\glxstrwordsep\glxstrwordsephyphen}{}%
    \ifglxstrinsertinside
      {\glsfirstabbrvhyphenfont{\glxstrgenentrytextfmt{#2}}}%
    \else
      {\glxstrgenentrytextfmt{#2}}%
    \fi
    \glxstrfullsep{#1}%
    \glxstrparen
      {\glxstrlongplformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
  }%
}

```

`\GLSxtrposthyphenlongpl` As above but all-caps.

```

\newcommand*{\GLSxtrposthyphenlongpl}[2]{%
  {%
    \glxstrifhyphenstart{#2}{\let\glxstrwordsep\glxstrwordsephyphen}{}%
    \ifglxstrinsertinside
      {\glsfirstabbrvhyphenfont{\mfirstucMakeUppercase{\glxstrgenentrytextfmt{#2}}}}%
    \else
      {\mfirstucMakeUppercase{\glxstrgenentrytextfmt{#2}}}%
    \fi
    \glxstrfullsep{#1}%
    \glxstrparen
      {\GLSxtrlongplformatgrp{#1}{#2}{\glsfirstlonghyphenfont}}%
  }%
}

```

`\xpglxstrposthyphenlong` Expand placeholders and check for all caps.

```

\newcommand*{\xpglxstrposthyphenlong}{%
  \glxstrifallcaps
  {%
    \expandafter\GLSxtrposthyphenlong\expandafter\glslabel
      \expandafter{\glsininsert}%
  }%
  {%
    \expandafter\glxstrposthyphenlong\expandafter\glslabel
      \expandafter{\glsininsert}%
  }%
}

```

`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.

```

\glxstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel

```

Setup the default fields.

```
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrshortlongname},
  sort={\glxtrshorthyphenlongsort},
  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}{%
    \glxtrifwasfirstuse
    {%
      \xpglxtrposthyphenlong
    }%
    {%
```

Put the insertion into the post-link:

```
      \xpglxtrposthyphensubsequent
    }%
  }%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
```

In case the user wants to mix and match font styles, these are redefined here.

```
\renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
\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*{\glxtrsubsequentfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glusifattribute{##1}{markshortwords}{true}%
  }%
  \glsaccessshort{##1}%
  }%
  {%
    \glsaccessfmtshort{\glxtrgenentrytextfmt}{##1}%
  }%
}%
}%
```

```

    {##2}%
}%
\renewcommand*{\glxtrsubsequentplfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glrifattribute{##1}{markshortwords}{true}%
    {%
      \glaccessshortpl{##1}%
    }%
    {%
      \glaccessfmtshortpl{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glrifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshort{##1}%
    }%
    {%
      \Glsaccessfmtshort{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glrifattribute{##1}{markshortwords}{true}%
    {%
      \Glsaccessshortpl{##1}%
    }%
    {%
      \Glsaccessfmtshortpl{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glrifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshort{##1}%
    }%
    {%
      \GLSaccessfmtshort{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%

```

```

    }%
  }%
  {##2}%
}%
\renewcommand*{\GLSxtrsubsequentplfmt}[2]{%
  \glxtrshorthyphennoinsert{##1}%
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \GLSaccessshortpl{##1}%
    }%
    {%
      \GLSaccessfmtshortpl{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##2}%
}%

```

First use full form:

```

\renewcommand*{\glxtrfullformat}[2]{%
  \glxtrshorthyphen
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshort{##1}%
    }%
    {%
      \glsaccessfmtshort{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##1}{##2}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrshorthyphen
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%
      \glsaccessshortpl{##1}%
    }%
    {%
      \glsaccessfmtshortpl{}{\glxtrgenentrytextfmt}{##1}%
    }%
  }%
  {##1}{##2}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glxtrshorthyphen
  {%
    \glsifattribute{##1}{markshortwords}{true}%
    {%

```

```

        \Glsaccessshort{##1}%
    }%
    {%
        \Glsaccessfmtshort{}{\glstrgenentrytextfmt}{##1}%
    }%
}%
{##1}{##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
    \glstrshorthyphen
    {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
            \Glsaccessshortpl{##1}%
        }%
        {%
            \Glsaccessfmtshortpl{}{\glstrgenentrytextfmt}{##1}%
        }%
    }{##1}{##2}%
}%
\renewcommand*{\GLSxtrfullformat}[2]{%
    \glstrshorthyphen
    {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
            \GLSaccessshort{##1}%
        }%
        {%
            \GLSaccessfmtshort{}{\glstrgenentrytextfmt}{##1}%
        }%
    }{##1}{##2}%
}%
\renewcommand*{\GLSxtrfullplformat}[2]{%
    \glstrshorthyphen
    {%
        \glsifattribute{##1}{markshortwords}{true}%
        {%
            \GLSaccessshortpl{##1}%
        }%
        {%
            \GLSaccessfmtshortpl{}{\glstrgenentrytextfmt}{##1}%
        }%
    }{##1}{##2}%
}%

```

In-line format. Commands like `\glstrfull` 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*\glstrinlinefullformat}[2]{%
  \glstrshortformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
}%
\renewcommand*\glstrinlinefullplformat}[2]{%
  \glstrshorttplformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
}%
\renewcommand*\Glsxtrinlinefullformat}[2]{%
  \Glsxtrshortformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
}%
\renewcommand*\Glsxtrinlinefullplformat}[2]{%
  \Glsxtrshorttplformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
}%
\renewcommand*\GLSxtrinlinefullformat}[2]{%
  \GLSxtrshortformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
}%
\renewcommand*\GLSxtrinlinefullplformat}[2]{%
  \GLSxtrshorttplformatgrp{##1}{##2}{\glsfirstabbrvhyphenfont}%
}%
}

```

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.

```

\glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel

```

Setup the default fields.

```

\renewcommand*\CustomAbbreviationFields{%
  name={\glstrshortlongdescname},
  sort={\glstrshortlongdescsort},%
  first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}},%
  firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}},%
  text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
}%
\renewcommand*\GlsXtrPostNewAbbreviation{%
  \csdef{glstrpostlink\glscategorylabel}{%
    \glstrifwasfirstuse
    {%
      \xpglstrposthyphenlong
    }%
  }%
}

```

Put the insertion into the post-link:

```

  \xpglstrposthyphensubsequent
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%

```



```

        \glsetattribute{\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:

```

\glstronlysuffix
    \newcommand*{\glstronlysuffix}{\glstrabbrvppluralsuffix}

\glstronlyname The default name format for this style.
    \newcommand*{\glstronlyname}{%
      \protect\glsabbrvonlyfont{\the\glsshorttok}%
    }

```

```

long-only-short-only
    \newabbreviationstyle{long-only-short-only}%
    {%

```

Set accessibility attributes if enabled.

```

    \glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel

```

Setup the default fields.

```

    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glstronlyname},
      sort={\the\glsshorttok},
      first={\glsfirstxplongfont{\the\glslongtok}{\glscategorylabel}},%
      firstplural={\glsfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},%
      text={\glspabbrvfont{\the\glsshorttok}{\glscategorylabel}},%
      plural={\glspabbrvfont{\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}%
  {%
    \glissetattribute{\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}%

```

}

Small-caps is awkward, so support for that is added.

`\glsabbrvsconlyfont`

`\newcommand*{\glsabbrvsconlyfont}{\glsabbrvscfont}%`

`\glsfirstabbrvsconlyfont`

`\newcommand*{\glsfirstabbrvsconlyfont}{\glsabbrvsconlyfont}%`

The default short form suffix:

`\glsxtrskonlysuffix`

`\newcommand*{\glsxtrskonlysuffix}{\glsxtrscsuffix}`

`\glsxtrskonlyrevert`

`\newcommand*{\glsxtrskonlyrevert}{\glsxtrscerevert}`

`\glsxtrskonlyname` The default name format for this style.

`\newcommand*{\glsxtrskonlyname}{%`
`\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={\glsxtrskonlyname},`
`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}%`
`{%`
`\glissetattribute{\the\glslabeltok}{regular}{false}%`
`}%`
`{}%`
`}%`
`{%`

```

\renewcommand*{\abbrvpluralsuffix}{\glxsxtrsconlysuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvsconlyfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvsconlyfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongonlyfont{##1}}%
\renewcommand*{\glxsxtrrevert}[1]{\glxsxtrsconlyrevert{##1}}%

```

The first use full form doesn't show the short form.

```

\renewcommand*{\glxsxtrfullformat}[2]{%
  \glsxtrlongformat{##1}{##2}{\glsfirstlongonlyfont}%
}%
\renewcommand*{\glxsxtrfullplformat}[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*{\glxsxtrinlinefullformat}[2]{%
  \glsxtrlongshortformat{##1}{##2}%
  {\glsfirstlongonlyfont}{\glsfirstabbrvsconlyfont}%
}%
\renewcommand*{\glxsxtrinlinefullplformat}[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}%
}%

```

```

    }%
  }

\glstrsconlydescsort
\newcommand*\glstrsconlydescsort{\glstronlydescsort}

\glstrsconlydescname
\newcommand*\glstrsconlydescname{\glstronlydescname}

long-only-short-sc-only-desc
\newabbreviationstyle{long-only-short-sc-only-desc}%
{%
Set accessibility attributes if enabled.
\glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
Setup the default fields.
\renewcommand*\CustomAbbreviationFields{%
  name={\glstrsconlydescname},
  sort={\glstrsconlydescsort},%
  first={\glstrfirstxplongfont{\the\glslongtok}{\glscategorylabel}},%
  firstplural={\glstrfirstxplongfont{\the\glslongpltok}{\glscategorylabel}},%
  text={\glstrxpabrvfont{\the\glsshorttok}{\glscategorylabel}},%
  plural={\glstrxpabrvfont{\the\glsshortpltok}{\glscategorylabel}}%
}%
Unset the regular attribute if it has been set.
\renewcommand*\GlsXtrPostNewAbbreviation{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\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{v1.49}{2022-10-14}

```

Declare package:

```
\ProvidesPackage{glossaries-extra-bib2gls}[2022/10/14 v1.49 (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}

    These are some convenient macros for use with custom rules.

\glshex
    \newcommand*{\glshex}{\string\u}

\glsapturedgroup
    \newcommand*{\glsapturedgroup}{\string\$}

\glshashchar Expands to a literal hash character (similar to \glsbackslash)
    \ifdef\glshashchar
    {}
    {\edef\glshashchar{\expandafter\@gobble\string\#}}

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
\newcommand#1%
}

```

`\glsxtr@wrglossarylocation{<wr-loc>}{<page>}`

`\glsxtr@wrglossarylocation`

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{<text>}{<label>}`

`\GlsXtrIndexCounterLink`

For use with `indexcounter` and `bib2gls`.

```

\ifdef\hyperref
{%
\newcommand*{\GlsXtrIndexCounterLink}[2]{%
\glsxtrifhasfield{indexcounter}{#2}%
{\hyperref[wrglossary.\glscurrentfieldvalue]{#1}}%
{#1}%
}
}
{
\newcommand*{\GlsXtrIndexCounterLink}[2]{#1}
}

```

`\GlsXtrDualField`

`\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{<text>}{<label>}`

`\GlsXtrDualBackLink`

Adds a hyperlink to the dual entry.

```

\newcommand*{\GlsXtrDualBackLink}[2]{%
\glsxtrifhasfield{\GlsXtrDualField}{#2}%
{\gls hyperlink[#1]{\glscurrentfieldvalue}}%
{#1}%
}

```

`\GlsXtrBibTeXEntryAliases` Convenient shortcut for use with `entry-type-aliases` to alias standard \LaTeX 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 \LaTeX 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}{\glsxtrbibschooll}%
  \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\glxtrsupplocationurl{#2}%
\glshypernumber{#1}%
}%
}
```

```
\glxtrdisplaysupplloc{<prefix>}{<counter>}{<format>}{<src>}{<location>}
```

`\glxtrdisplaysupplloc`

This is like `\glsnoidxdisplayloc` but is used for supplementary locations and so requires an extra argument.

```
\newcommand*\glxtrdisplaysupplloc[5]{%
\setentrycounter{#1}{#2}%
\glxtrmultisuppllocation{#5}{#4}{#3}%
}
```

`\glxtrdisplaylocnameref` `\glxtrdisplaylocnameref{<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*\glxtrdisplaylocnameref[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 `\glxtr<counter>locfmt` is defined. The prefix argument is redundant.

```
\newcommand*\glxtrdisplaylocnameref[8]{%
\def\glxtrrecentanchor{#6}%
\def\glxtrlocationanchor{#2.#7}%
```

Initialise `\glxtractualanchor`:

```
\let\glxtractualanchor\glxtrlocationanchor
\glxtrsetactualanchor{#2}%
\ifcsdef{glxtr#2locfmt}%
{\glxtrnameref{link}{#3}{\csuse{glxtr#2locfmt}{#4}{#5}}{\glxtractualanchor}{#8}}%
{%
\ifstrempy{#5}%
{%
```

No title, so just use the location as the link text.

```

\glxtrnamerefink{#3}{#4}{\glxtractualanchor}{#8}%
}%
{%
\ifstrequal{#2}{page}%
{\glxtrnamerefink{#3}{#4}{\glxtractualanchor}{#8}}%
{\glxtrtitlednamerefink{#3}{#4}{#5}{#8}}%
}%
}%
}
}

```

`\glxtractualanchor` Does nothing by default. May be redefined to override the default.

```
\newcommand{\glxtrsetactualanchor}[1]{}
```

```

\glxtrtitlednamerefink{<format>}{<location>}{<title>}
{<file>}

```

`\glxtrtitlednamerefink`

```

\newcommand{\glxtrtitlednamerefink}[4]{%
\glxtrnamerefink{#1}{#2}{\glxtrrecentanchor}{#4}%
}

```

```
\glxxtreuationlocfmt{<location>}{<title>}
```

`\glxxtreuationlocfmt`

```
\newcommand*{\glxxtreuationlocfmt}[2]{( #1 )}
```

```
\glxtrwrglossarylocfmt{<location>}{<title>}
```

`\glxtrwrglossarylocfmt`

```

\newcommand*{\glxtrwrglossarylocfmt}[2]{%
{\@@glxtrwrglosscountermark{#1}%
\let\glxtr@wrglossarylocation\@secondoftwo
#1}%
}

```

```

\glxtrfmtnamerefink{<format>}{<title>}{<href>}{<external
file>}

```

`\glxtrnamerefink`

```
\newcommand*{\glxtrnamerefink}[4]{%
```

Locally change `\glshypernumber` to `\@firstofone` to remove the normal location hyperlink.

```

\begin{group}
\let\glshypernumber\@firstofone

```

If the *<external file>* argument is empty, an internal link is used, otherwise an external one is needed.

```
\ifstrempy{#4}%
{\glxstrfmtinternalnameref{#3}{#1}{#2}}%
{\glxstrfmtexternalnameref{#3}{#1}{#2}{#4}}%
\endgroup
}
```

```
\glxtrnameloclink{<prefix>}{<counter>}{<format>}
{<location>}{<text>}{<external
file>}
```

`\glxtrnameloclink`

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{\glxtrnameloclink}[6]{%
\begingroup
\setentrycounter[#1]{#2}%
\def\glxtr@locationhypertext{#5}%
\let\glshypernumber\@firstofone
\def\@glsnumberformat{#3}%
\def\glxtrsupplocationurl{#6}%
\toks@={}%
\@glxtr@bibgls@removespaces#4 \@nil
\endgroup
}
```

`\@glxtr@bibgls@removespaces`

```
\def\@glxtr@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{\glentrycounter\@glo@counterprefix\the\toks@}%
\ifdefined\glxtrsupplocationurl
{%
\expandafter\glxstrfmtinternalnameref\expandafter{\@glo@tmp}%
{\@glsnumberformat}{\glxtr@locationhypertext}%
}%
{%
\expandafter\glxstrfmtexternalnameref\expandafter{\@glo@tmp}%
{\@glsnumberformat}{\glxtr@locationhypertext}{\glxtrsupplocationurl}%
}%
\fi
```

```

\else
  \@gls@ReturnAfterFi{%
    \@glsxtr@bibgls@removespaces#2\@nil
  }%
\fi
}

```

`\glsxtrfmtinternalnameref`

```
\glsxtrfmtinternalnameloc{<target>}{<format>}{<title>}
```

```

\newcommand*{\glsxtrfmtinternalnameref}[3]{%
  \csuse{#2}{\glsdohyperlink{#1}{#3}}%
}

```

`\glsxtrfmtexternalnameref`

```
\glsxtrfmtexternalnameloc{<target>}{<format>}{<title>}
{<file>}
```

```

\newcommand*{\glsxtrfmtexternalnameref}[4]{%
  \csuse{#2}{\hyperref{#4}{#1}{#3}}%
}

```

`\glsxtrSetWidest`

```
\glsxtrSetWidest{<type>}{<level>}{<text>}
```

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.

```

\ifstrempy{#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.

```

\ifstrempy{#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.

```

\ifstrempy{#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
}%
}%
{%

```

Only alttree.

```

\ifstrempy{#1}
{%
  \glsssetwidest[#2]{#3}%
}%
{%
  \apptoglossarypreamble[#1]{\glsssetwidest[#2]{#3}}%
}%
}%
}%
{%
  \ifdef\glslongextraUpdateWidest
  {%

```

glossary-longextra has been loaded.

```

\ifstrempy{#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.

```

{}%
}%
}%
}

```

`\glsxtrSetWidestFallback{ $\langle max\ depth \rangle$ }{ $\langle list \rangle$ }`

`\glsxtrSetWidestFallback`

Used when bib2gls can't determine the widest name. The $\langle list \rangle$ argument is a comma-separated list of glossary labels. The $\langle max\ depth \rangle$ 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
}

```

`\@glxtr@labelprefixes` List of label prefixes.

```
\newcommand*{\@glxtr@labelprefixes}{}
```

`\glxtrclearlabelprefixes` List of label prefixes.

```

\newcommand*{\glxtrclearlabelprefixes}{%
\renewcommand*{\@glxtr@labelprefixes}{}}%
}

```

`\glxtraddlabelprefix` 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*{\glxtraddlabelprefix}[1]{%
\ifstrepty{#1}%
{\glxtraddlabelprefix{\empty}}%
{%
\ifdefempty\@glxtr@labelprefixes
{\def\@glxtr@labelprefixes{#1}}%
{\appto\@glxtr@labelprefixes{,#1}}%
}%
}

```

`\glxtrprependlabelprefix` Inserts at the start of the list.

```

\newcommand*{\glxtrprependlabelprefix}[1]{%
\ifstrepty{#1}%
{\glxtrprependlabelprefix{\empty}}%
}

```

```

{%
  \ifdefempty\@glsxtr@labelprefixes
  {\def\@glsxtr@labelprefixes{#1}}%
  {\preto\@glsxtr@labelprefixes{#1,}}%
}%
}

```

`\glsxtrifinlabelprefixlist{prefix}{true}{false}`

`\glsxtrifinlabelprefixlist`

Test if the given prefix is in the list.

```

\newcommand*\glsxtrifinlabelprefixlist}[3]{%
  \ifstrempy{#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

```

```

\@glxtr@use@equation@counter
\fi
\@gls@setdefault@glslink@opts
\@glxtr@glslink@prekeys
\setkeys{glslink}{#1}%
\glxtr@do@autoadd{glslink}%

```

Can't increment associated counter.

```

\ifKV@glslink@noindex
\GlossariesExtraWarning{Can't obtain a match for prefix
candidates: \@glxtr@prefixedlist. Check the label spelling or rerun}%
\else
\ifdefempty{\@glxtr@thevalue}%
{%
\ifx\@glxtr@org@theHvalue\@glxtr@theHvalue
\else
\let\theHglentrycounter\@glxtr@theHvalue
\fi
}%
{%
\let\theHglentrycounter\@glxtr@thevalue
\let\theHglentrycounter\@glxtr@theHvalue
}%
\glxtr@saveentrycounter
\@glxtr@dorecord\@glxtr@prefixedlist
\glxtr@select@entry\glxtr@do@select@nameref@record

```

Issue warning.

```

\GlossariesExtraWarning{Can't obtain a match for prefix
candidates: \@glxtr@prefixedlist. Check the label spelling, use bib2gls v3.0+ to
select entry and rerun LaTeX}%
\fi
\@glxtrundeftag#3%
\endgroup
}

```

`\glxtr@select@entry` Instruction to bib2gls to select the first found label in the list.

```
\newcommand*{\glxtr@select@entry}[5]{}
```

`\glxtr@select@entry@nameref` Instruction to bib2gls to select the first found label in the list as though it has a record.

```
\newcommand*{\glxtr@select@entry@nameref}[8]{}
```

`\glxtr@do@select@nameref@record` Instruction to bib2gls to select the first found label in the list as though it has a record.

```

\newcommand*{\glxtr@do@select@nameref@record}[5]{%
\gls@ifnotmeasuring
{%
\protected@write\@auxout{}{\string\glxtr@select@entry@nameref
{#1}{#2}{#3}{#4}{#5}%

```

```

        {\csuse{@currentlabelname}}{\csuse{@currentHref}}}%
        {\theHglentrycounter}}%
    }%
}

```

`sXtrPrefixLabelFallbackLast` 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

```

`\@glxtr@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 L^AT_EX run.

```

\newcommand*{\@glxtr@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\@glxtr@prefixedfirstlabel\@gls@thislabel
\def\@glxtr@set@prefixedfirstlabel{%
    \let\@glxtr@prefixedfirstlabel\@gls@thislabel
    \let\@glxtr@set@prefixedfirstlabel\relax
}%

```

List of labels in the event that no combination is found.

```

\let\@glxtr@prefixedlist\@empty

```

Iterate over all labels.

```

\count@=0\relax
\@for\@glxtr@prefix:=\@glxtr@labelprefixes\do
{%
    \advance\count@ by 1\relax
    \protected@edef\@gls@thislabel{\@glxtr@prefix#1}%
    \@glxtr@set@prefixedfirstlabel
}

```

Check if this label exists.

```

\ifglentryexists{\@gls@thislabel}%
{%
    \@endfortrue
}

```

Found a label that exists. Clear the list.

```

\let\@glxtr@prefixedlist\@empty
}%
{%

```

Append or prepend to list.

```

\ifdefempty\@glxtr@prefixedlist
{\let\@glxtr@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
\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 system, 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@}]

\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\dglsfielddcurrentfielddlabel{#2}%
  \let\dglsfielddactualfielddlabel\dglsfielddcurrentfielddlabel

```

Grouping is used in case of a nested for loop.

```
\begingroup
```

Initialise to the unprefix 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}@\dglsgfieldfallbackfieldlabel}%
}%
{%
  \let\@gls@fallbacklabel\@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
```

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.

```
\let\@gls@thislabel\@gls@fallbacklabel
\let\dglsfieldactualfieldlabel\dglsfieldfallbackfieldlabel
\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{<options>}{<label>}{<field>}{<cs>}
```

\@@dgls@@field

```
\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}{}
```

`\dglsgfieldfallbackfieldlabel` The field to use if the required field isn't set.

```
\newcommand*{\dglsgfieldfallbackfieldlabel}{text}
```

`\dglsgfieldactualfieldlabel` This is the field that's actually used.

```
\newcommand*{\dglsgfieldactualfieldlabel}{\dglsgfieldcurrentfieldlabel}
```

```
\dglsgfield[\langle options \rangle]{\langle label \rangle}{\langle field \rangle}[\langle insert \rangle]
```

`\dglsgfield`

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

`\@dglsgfield`

```
\newcommand*{\@dglsgfield}[3][{}]{%  
  \@dglsgs@field{#1}{#2}{#3}{\@dglsgs@field}}
```

`\@dglsgs@field`

```
\def\@dglsgs@field#1#2[#3]{%  
  \@glsgs@field@link{#1}{#2}{\glsgxtrusefield{#2}{\dglsgfieldactualfieldlabel}#3}%  
}
```

```
\dGlsfield[\langle options \rangle]{\langle label \rangle}{\langle field \rangle}[\langle insert \rangle]
```

`\dGlsfield`

```
\newrobustcmd*{\dGlsfield}{\@gls@hyp@opt\dGlsfield}  
\glsgmfuaddmap{\dglsgfield}{\dGlsfield}
```

`\@dGlsfield`

```
\newcommand*{\@dGlsfield}[3][{}]{%  
  \@dglsgs@field{#1}{#2}{#3}{\@dGls@field}%  
}
```

`\@dGls@field`

```
\def\@dGls@field#1#2[#3]{%  
  \@glsgs@field@link{#1}{#2}{\glsgxtrusefield{#2}{\dglsgfieldactualfieldlabel}#3}%  
}
```

```
\dGLSfield[\langle options \rangle]{\langle label \rangle}{\langle field \rangle}[\langle insert \rangle]
```

`\dGLSfield`

```
\newrobustcmd*{\dGLSfield}{\@gls@hyp@opt\dGLSfield}  
\glsgmfublocker{\dGLSfield}
```

`\@dGLSfield`

```
\newcommand*{\@dGLSfield}[3][{}]{%  
  \@dglsgs@field{#1}{#2}{#3}{\@dGLS@field}%  
}
```

```

\@dGLS@field
\def\@dGLS@field#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\GLSxtrusefield{#2}{\dglSfieldactualfieldlabel}{#3}%
}

\@dinner@glSfield{\langle default options \rangle}{\langle field \rangle}{\langle modifier \rangle}
[\langle options \rangle]{\langle label \rangle}[\langle insert \rangle]

\@dinner@glSfield
\newrobustcmd*{\@dinner@glSfield}[2]{%
  \ifstrepty{#1}
  {\def\@dinner@glSfield@opts{}}%
  {\def\@dinner@glSfield@opts{#1,}}%
  \def\dglSfieldcurrentfieldlabel{#2}%
  \@gls@hyp@opt\@dinner@glSfield
}

\@dinner@glSfield
\newcommand*{\@dinner@glSfield}[2][{}]{%
  \expandafter\@dglS@@field\expandafter
  {\@dinner@glSfield@opts#1}{#2}{\dglSfieldcurrentfieldlabel}{\@dglS@field}}

\@dinner@Glsfield
\newrobustcmd*{\@dinner@Glsfield}[2]{%
  \ifstrepty{#1}
  {\def\@dinner@glSfield@opts{}}%
  {\def\@dinner@glSfield@opts{#1,}}%
  \def\dglSfieldcurrentfieldlabel{#2}%
  \@gls@hyp@opt\@dinner@Glsfield
}

\@dinner@Glsfield
\newcommand*{\@dinner@Glsfield}[2][{}]{%
  \expandafter\@dglS@@field\expandafter
  {\@dinner@glSfield@opts#1}{#2}{\dglSfieldcurrentfieldlabel}{\@dGls@field}}

\@dinner@GLSfield
\newrobustcmd*{\@dinner@GLSfield}[2]{%
  \ifstrepty{#1}
  {\def\@dinner@glSfield@opts{}}%
  {\def\@dinner@glSfield@opts{#1,}}%
  \def\dglSfieldcurrentfieldlabel{#2}%
  \@gls@hyp@opt\@dinner@GLSfield
}

\@dinner@GLSfield
\newcommand*{\@dinner@GLSfield}[2][{}]{%
  \expandafter\@dglS@@field\expandafter
  {\@dinner@glSfield@opts#1}{#2}{\dglSfieldcurrentfieldlabel}{\@dGLS@field}}

```

\newdglsglsfield

\newdglsglsfield[*<options>*]{*<field>*}{*<cs>*}

```
\newrobustcmd*{\newdglsglsfield}[3][]{%
  \newrobustcmd*{#3}{\d@inner@glsglsfield{#1}{#2}}%
}
```

\newdglsglsfieldlike

\newdglsglsfieldlike[*<options>*]{*<field>*}{*<cs>*}{*<Cs>*}{*<CS>*}

```
\newrobustcmd*{\newdglsglsfieldlike}[5][]{%
  \newrobustcmd*{#3}{\d@inner@glsglsfield{#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

\glsxtrmultientryadjustedname{*<list1>*}{*<name>*}{*<list2>*}{*<label>*}

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 *<label>*.

The final argument *<label>* is the multi-entry label from which the set was obtained. The first argument *<list1>* is the list of other labels that come before the main label. The third argument *<list2>* is the remaining list of other labels. The *<name>* 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
}
```

glsxtrmultientryadjustedname First letter upper case

```
\newrobustcmd*{\Glsxtrmultientryadjustedname}[4]{%
  \bgroup
  \let\@glsxtrmultientryadjustednamesep\glsxtrmultientryadjustednamesep
  \let\@glsxtrmultientryadjustednamepresep\glsxtrmultientryadjustednamepresep
```

```

\let\@glxstrmultientryadjustednamepostsep\glxstrmultientryadjustednamepostsep
\let\@glxstrmultientryadjustednameother\glxstrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefmt\glxstrmultientryadjustednamefmt
\let\@glxstrmultientryadjustednamefirsttother\Glsxtrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefirstfmt\Glsxtrmultientryadjustednamefmt
\@glxstrmultientryadjustedname{#1}{#2}{#3}{#4}%
\egroup
}
\glsmfuaddmap{\glxstrmultientryadjustedname}{\Glsxtrmultientryadjustedname}

\GlsXtrmultientryadjustedname Title case
\newrobustcmd*{\GlsXtrmultientryadjustedname}[4]{%
\bgroup
\let\@glxstrmultientryadjustednamesep\glxstrmultientryadjustednamesep
\let\@glxstrmultientryadjustednamepresep\glxstrmultientryadjustednamepresep
\let\@glxstrmultientryadjustednamepostsep\glxstrmultientryadjustednamepostsep
\let\@glxstrmultientryadjustednameother\GlsXtrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefmt\GlsXtrmultientryadjustednamefmt
\let\@glxstrmultientryadjustednamefirsttother\GlsXtrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefirstfmt\GlsXtrmultientryadjustednamefmt
\@glxstrmultientryadjustedname{#1}{#2}{#3}{#4}%
\egroup
}
\glsmfublocker{\GlsXtrmultientryadjustedname}

\GlsXtrmultientryadjustedname All caps.
\newrobustcmd*{\GlsXtrmultientryadjustedname}[4]{%
\bgroup
\let\@glxstrmultientryadjustednamesep\glxstrmultientryadjustednamesep
\let\@glxstrmultientryadjustednamepresep\glxstrmultientryadjustednamepresep
\let\@glxstrmultientryadjustednamepostsep\glxstrmultientryadjustednamepostsep
\let\@glxstrmultientryadjustednameother\GlsXtrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefmt\GlsXtrmultientryadjustednamefmt
\let\@glxstrmultientryadjustednamefirsttother\GlsXtrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefirstfmt\GlsXtrmultientryadjustednamefmt
\@glxstrmultientryadjustedname{#1}{#2}{#3}{#4}%
\egroup
}
\glsmfublocker{\GlsXtrmultientryadjustedname}

\glxstrmultientryadjustedname
\newcommand*{\@glxstrmultientryadjustedname}[4]{%
\letcs\mglscurrentmainlabel{\@glsc@combined@#4@main}%
\letcs\mglscurrentmainlist{\@glsc@combined@#4@list}%
\letcs\mglscurrentmainoptions{\@glsc@combined@#4@options}%
\ifblank{#1}%
{%
\@glxstrmultientryadjustednamefirstfmt{#2}%
}%
}

```

```

{%
\def\@mgls@previouslabel{}%
\let\@gls@xtradjustedother\@glxtrmultientryadjustednamefirstother
\@for\mgls@currentlabel:=#1\do{%
\ifx\@mgls@previouslabel\empty
\else
\@glxtrmultientryadjustednamesep{\@mgls@previouslabel}{\mgls@currentlabel}%
\fi
\@gls@xtradjustedother{\mgls@currentlabel}%
\let\@mgls@previouslabel\mgls@currentlabel
\let\@gls@xtradjustedother\@glxtrmultientryadjustednameother
}%
\@glxtrmultientryadjustednamepresep{\@mgls@previouslabel}{\mgls@currentmainlabel}%
\@glxtrmultientryadjustednamefmt{#2}%
}%
\ifblank{#3}%
{}%
{%
\let\@mgls@previouslabel\mgls@currentmainlabel
\let\@gls@xtrmultientryadjustednamesep\@glxtrmultientryadjustednamepostsep
\@for\mgls@currentlabel:=#3\do{%
\@gls@xtrmultientryadjustednamesep{\@mgls@previouslabel}{\mgls@currentlabel}%
\@glxtrmultientryadjustednameother{\mgls@currentlabel}%
\let\@mgls@previouslabel\mgls@currentlabel
\let\@gls@xtrmultientryadjustednamesep\@glxtrmultientryadjustednamesep
}%
}%
}

```

trmultientryadjustednamesep

```
\newcommand*{\glxtrmultientryadjustednamesep}{\glscombinedfirstsepfirst}
```

multientryadjustednamepresep Separator before main name.

```
\newcommand*{\glxtrmultientryadjustednamepresep}{\glxtrmultientryadjustednamesep}
```

ltientryadjustednamepostsep Separator after main name.

```
\newcommand*{\glxtrmultientryadjustednamepostsep}{\glxtrmultientryadjustednamesep}
```

trmultientryadjustednamefmt

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

multientryadjustednameother

```
\newcommand*{\glxtrmultientryadjustednameother}[1]{\glsentryname{#1}}
```

trmultientryadjustednamefmt

```
\newcommand*{\Glsxtrmultientryadjustednamefmt}[1]{\glsentencecase{#1}}
```

multientryadjustednameother

```
\newcommand*{\Glsxtrmultientryadjustednameother}[1]{\Glsentryname{#1}}
```

```

\multientryadjustednameother
\newcommand*{\GlsXtrmultientryadjustednameother}[1]{%
\glsentrytitlecase{#1}{name}}

\trmultientryadjustednamefmt
\ifdef\glscapitalisewords
{%
\newcommand*{\GlsXtrmultientryadjustednamefmt}[1]{\glscapitalisewords{#1}}
}
{
\newcommand*{\GlsXtrmultientryadjustednamefmt}[1]{\capitalisewords{#1}}
}

\multientryadjustednameother
\newcommand*{\GLSxtrmultientryadjustednameother}[1]{%
\glsupercase{\glsentryname{#1}}}

\trmultientryadjustednamefmt
\newcommand*{\GLSxtrmultientryadjustednamefmt}[1]{\glsupercase{#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}}

\Upsilon
\providecommand*\Upsilon{\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

```

```
% <\glxtrdigitrules
% <\glxtrfractionrules
% <\glxtrGeneralLatinIVrules
% <\glxtrMathItalicGreekIrules
%}
%
```

`\glxtrIgnorableRules` A shortcut command for common ignorable characters.

```
\newcommand{\glxtrIgnorableRules}{%
\glxtrcontrolrules
\string;\glxtrspacerules
\string;\glxtrnonprintablerules
}
```

`\glxtrGeneralInitRules` A shortcut command for common initial rules for ignorables, diacritics, punctuation and digits.

```
\newcommand{\glxtrGeneralInitRules}{%
\glxtrIgnorableRules
\string;\glxtrcombiningdiacriticrules
\string;\glxtrhyphenrules
\string<\glxtrgeneralpuncrules
\string<\glxtrdigitrules
\string<\glxtrfractionrules
}
```

`\glxtrcontrolrules` 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. `\string` is used for punctuation characters in case they’ve been made active.

```
\newcommand*{\glxtrcontrolrules}{%
\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}
```

}

`\glxtrspacerules` These are space characters.

```
\newcommand*{\glxtrspacerules}{%
\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'
}
```

`\glxtrnonprintablerules` These are non-printable characters (BOM, tabs, line feed and carriage return).

```
\newcommand*{\glxtrnonprintablerules}{%
\string'\glshex FEFF\string'\string;
\string'\glshex 000A\string'\string;
\string'\glshex 0009\string'\string;
\string'\glshex 000C\string'\string;
\string'\glshex 000B\string'
}
```

`\glxtrcombiningdiacriticrules` Combining diacritic marks. This is split into multiple macros.

```
\newcommand*{\glxtrcombiningdiacriticrules}{%
\glxtrcombiningdiacriticIrules\string;
\glxtrcombiningdiacriticIIrules\string;
\glxtrcombiningdiacriticIIIrules\string;
\glxtrcombiningdiacriticIVrules
}
```

`\glxtrcombiningdiacriticIrules` First set of combining diacritic marks.

```
\newcommand*{\glxtrcombiningdiacriticIrules}{%
\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\string;% combining macron
}

```

trcombingdiacriticIIrules Second set of combining diacritic marks.

```

\newcommand*{\glxtrcombingdiacriticIIrules}{%
\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\string;% combining acute accent below
}

```

trcombingdiacriticIIIrules Third set of combining diacritic marks.

```

\newcommand*{\glxtrcombingdiacriticIIIrules}{%
\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
}

```

ctrcombiningdiacriticIVrules Fourth set of combining diacritic marks.

```

\newcommand*{\glxtrcombiningdiacriticIVrules}{%
\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
}

```

\glxtrhyphenrules Hyphens.

```

\newcommand*{\glxtrhyphenrules}{%
\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
}

```

\glxtrgeneralpuncrules General punctuation.

```

\newcommand*{\glxtrgeneralpuncrules}{%
  \glxtrgeneralpuncIrules
  \string<\glxtrcurrencyrules
  \string<\glxtrgeneralpuncIIrules
}

```

\glxtrgeneralpuncIrules First set of general punctuation.

```

\newcommand*{\glxtrgeneralpuncIrules}{%
  \glxtrgeneralpuncmarksrules
  \string<\glxtrgeneralpuncaccentsrules
  \string<\glxtrgeneralpuncquoterules
  \string<\glxtrgeneralpuncbracketrules
  \string<\glxtrgeneralpuncsignrules
}

```

\glxtrgeneralpuncmarksrules Punctuation marks subset.

```

\newcommand*{\glxtrgeneralpuncmarksrules}{%
  \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
}

```

\glxtrgeneralpuncaccentsrules Punctuation marks subset: accent characters.

```

\newcommand*{\glxtrgeneralpuncaccentsrules}{%
  \glshex 00B4% acute accent
  \string<\string'\glshex 0060\string'% grave accent
  \string<\string'\glshex 005E\string'% circumflex accent
  \string<\glshex 00A8% diaeresis
  \string<\string'\glshex 007E\string'% tilde
  \string<\glshex 00B7% middle dot
  \string<\glshex 00B8% cedilla
}

```

\glstrgeneralpuncquoterules Punctuation marks subset: quotes.

```
\newcommand*{\glstrgeneralpuncquoterules}{%
\string'\glshex 0027\string'% straight apostrophe
\string<\string'\glshex 0022\string'% straight double quote
\string<\glshex 00AB% left guillemet
\string<\glshex 00BB% right guillemet
}
```

\glstrgeneralpuncbracketrules Punctuation marks subset: brackets.

```
\newcommand*{\glstrgeneralpuncbracketrules}{%
\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
}
```

\glstrgeneralpuncsignrules Punctuation marks subset: signs.

```
\newcommand*{\glstrgeneralpuncsignrules}{%
\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
}
```

\glstrcurrencyrules General punctuation.

```
\newcommand*{\glstrcurrencyrules}{%
\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
}
```

`\glxstrgeneralpuncIIrules` Second set of general punctuation.

```
\newcommand*{\glxstrgeneralpuncIIrules}{%
  \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
}
```

`\glxstrGeneralLatinIrules` Basic Latin alphabet.

```
\newcommand*{\glxstrGeneralLatinIrules}{%
  \glxstrLatinA
  \string<b,B%
  \string<c,C%
  \string<d,D%
  \string<\glxstrLatinE
  \string<f,F%
  \string<g,G%
  \string<\glxstrLatinH
  \string<\glxstrLatinI
  \string<j,J%
  \string<\glxstrLatinK
  \string<\glxstrLatinL
  \string<\glxstrLatinM
  \string<\glxstrLatinN
  \string<\glxstrLatinO
  \string<\glxstrLatinP
  \string<q,Q%
  \string<r,R%
  \string<\glxstrLatinS
  \string<\glxstrLatinT
  \string<u,U%
  \string<v,V%
  \string<w,W%
  \string<\glxstrLatinX
  \string<y,Y%
  \string<z,Z
```


}

`\glxtrGeneralLatinIIrules` General Latin alphabet (eth between D and E, § treated as SS).

```
\newcommand*{\glxtrGeneralLatinIIrules}{%
  \glxtrLatinA
  \string<b,B%
  \string<c,C%
  \string<d,D%
  \string<\glxtrLatinEth
  \string<\glxtrLatinE
  \string<f,F%
  \string<g,G%
  \string<\glxtrLatinH
  \string<\glxtrLatinI
  \string<j,J%
  \string<\glxtrLatinK
  \string<\glxtrLatinL
  \string<\glxtrLatinM
  \string<\glxtrLatinN
  \string<\glxtrLatinO
  \string<\glxtrLatinP
  \string<q,Q%
  \string<r,R%
  \string<\glxtrLatinS
  \string& SS \string, \glxtrLatinEszettSs
  \string<\glxtrLatinT
  \string<u,U%
  \string<v,V%
  \string<w,W%
  \string<\glxtrLatinX
  \string<y,Y%
  \string<z,Z%
}
```

`\glxtrGeneralLatinIIIrules` General Latin alphabet (eth between D and E, § treated as SZ).

```
\newcommand*{\glxtrGeneralLatinIIIrules}{%
  \glxtrLatinA
  \string<b,B%
  \string<c,C%
  \string<d,D%
  \string<\glxtrLatinEth
  \string<\glxtrLatinE
  \string<f,F%
  \string<g,G%
  \string<\glxtrLatinH
  \string<\glxtrLatinI
  \string<j,J%
  \string<\glxtrLatinK
  \string<\glxtrLatinL
  \string<\glxtrLatinM
```

```

\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SZ, \glxtrLatinEszettSz
\string<\glxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}

```

`\glxtrGeneralLatinIVrules` General Latin alphabet (Æ treated as AE and Æ treated as OE, Þ treated as TH, ð treated as SS, eth between D and E).

```

\newcommand*{\glxtrGeneralLatinIVrules}{%
\glxtrLatinA
\string& AE , \glxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string& OE , \glxtrLatinOELigature
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SS , \glxtrLatinEszettSs
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
}

```

```

\string<z,Z%
}

```

`\glxtrGeneralLatinVrules` General Latin alphabet (eth between D and E, ð treated as SS, Þ treated as TH).

```

\newcommand*{\glxtrGeneralLatinVrules}{%
\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SS , \glxtrLatinEszettSs
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}

```

`\glxtrGeneralLatinVirules` General Latin alphabet (eth between D and E, ð treated as SZ, Þ treated as TH).

```

\newcommand*{\glxtrGeneralLatinVirules}{%
\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%

```

```

\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SZ , \glxtrLatinEszettSz
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}

```

`\glxtrGeneralLatinVIIrules` General Latin alphabet (\mathring{A} between A and B, eth between D and E, insular G as G, \mathring{C} between O and P, long S equivalent to S, \mathring{P} between T and U and wynn as W).

```

\newcommand*{\glxtrGeneralLatinVIIrules}{%
\glxtrLatinA
\string<\glxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<\glxtrLatinInsularG
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinOELigature
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glshex 017F=\glxtrLatinS % s and long s

```

```

\string<\glxtrLatinT
\string<\glxtrLatinThorn
\string<u,U%
\string<v,V%
\string< w\string=\glshex 01BF, W\string=\glshex 01F7
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}

```

`\glxtrGeneralLatinVIIIrules` General Latin alphabet (Æ treated as AE and Ċ treated as OE, Þ treated as TH, ß treated as SS, eth treated as D, Ø treated as O, Ł treated as L).

```

\newcommand*{\glxtrGeneralLatinVIIIrules}{%
\glxtrLatinA
\string& AE , \glxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<\glshex 00F0\string;d,\glshex 00D0\string;D% D and eth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glshex 0142\string=\glxtrLatinL\string=\glshex 0141% L and \L
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glshex 00F8\string=\glxtrLatinO\string=\glshex 00D8% O and \O
\string& OE , \glxtrLatinOELigature
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SS , \glxtrLatinEszettSs
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}

```

Fragments.

`\glxtrGeneralLatinAtoMrules` Basic Latin alphabet A–M.

```

\newcommand*{\glxtrGeneralLatinAtoMrules}{%

```

```

\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
}

```

`\glxtrGeneralLatinNtoZrules` Basic Latin alphabet N–Z.

```

\newcommand*{\glxtrGeneralLatinNtoZrules}{%
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string<\glxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z
}

```

`\glxtrGeneralLatinAtoGrules` Basic Latin alphabet A–G.

```

\newcommand*{\glxtrGeneralLatinAtoGrules}{%
\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
}

```

`\glxtrGeneralLatinHtoMrules` Basic Latin alphabet H–M.

```

\newcommand*{\glxtrGeneralLatinHtoMrules}{%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK

```

```

        \string<\glxtrLatinL
        \string<\glxtrLatinM
    }

\glxtrGeneralLatinNtoSrules Basic Latin alphabet N–S.
    \newcommand*{\glxtrGeneralLatinNtoSrules}{%
        \string<\glxtrLatinN
        \string<\glxtrLatinO
        \string<\glxtrLatinP
        \string<q,Q%
        \string<r,R%
        \string<\glxtrLatinS
    }

\glxtrGeneralLatinTtoZrules Basic Latin alphabet T–Z.
    \newcommand*{\glxtrGeneralLatinTtoZrules}{%
        \string<\glxtrLatinT
        \string<u,U%
        \string<v,V%
        \string<w,W%
        \string<\glxtrLatinX
        \string<y,Y%
        \string<z,Z
    }

\glxtrLatinA
    \newcommand*{\glxtrLatinA}{%
        a\string=\glshex 00AA\string=\glshex 2090,A
    }

\glxtrLatinE
    \newcommand*{\glxtrLatinE}{%
        e\string=\glshex 2091,E
    }

\glxtrLatinH
    \newcommand*{\glxtrLatinH}{%
        h\string=\glshex 2095,H
    }

\glxtrLatinI
    \newcommand*{\glxtrLatinI}{%
        i\string=\glshex 2071,I
    }

\glxtrLatinK
    \newcommand*{\glxtrLatinK}{%
        k\string=\glshex 2096,K
    }

```

```

\glxtrLatinL
\newcommand*{\glxtrLatinL}{%
  l\string=\glshex 2097,L
}

\glxtrLatinM
\newcommand*{\glxtrLatinM}{%
  m\string=\glshex 2098,M
}

\glxtrLatinN
\newcommand*{\glxtrLatinN}{%
  n\string=\glshex 207F\string=\glshex 2099,N
}

\glxtrLatinO
\newcommand*{\glxtrLatinO}{%
  o\string=\glshex 00BA\string=\glshex 2092,O
}

\glxtrLatinP
\newcommand*{\glxtrLatinP}{%
  p\string=\glshex 209A,P
}

\glxtrLatinS
\newcommand*{\glxtrLatinS}{%
  s\string=\glshex 209B,S
}

\glxtrLatinT
\newcommand*{\glxtrLatinT}{%
  t\string=\glshex 209C,T
}

\glxtrLatinX
\newcommand*{\glxtrLatinX}{%
  x\string=\glshex 2093,X
}

\glxtrLatinSchwa Latin schwa (lower case, subscript and upper case).
\newcommand*{\glxtrLatinSchwa}{%
  \glshex 0259\string=\glshex 2094,\glshex 018F
}

\glxtrLatinEszettSs SS=ss
\newcommand*{\glxtrLatinEszettSs}{%
  \glshex 00DF% eszett
  \string=\glshex 017Fs % "long S"s
}

```



```

\glxtrLatinEszettSz SS=sz
    \newcommand*\glxtrLatinEszettSz{%
        \glshex 00DF% eszett
        \string= \glshex 017Fz % "long S"z
    }

\glxtrLatinEth
    \newcommand*\glxtrLatinEth{%
        \glshex 00F0,\glshex 00D0% eth
    }

\glxtrLatinThorn
    \newcommand*\glxtrLatinThorn{%
        \glshex 00FE,\glshex 00DE% thorn
    }

\glxtrLatinAELigature
    \newcommand*\glxtrLatinAELigature{%
        \glshex 00E6,\glshex 00C6% AE-ligature
    }

\glxtrLatinOELigature
    \newcommand*\glxtrLatinOELigature{%
        \glshex 0153,\glshex 0152% OE-ligature
    }

\glxtrLatinAA
    \newcommand*\glxtrLatinAA{%
        \glshex 00E5=a\glshex 030A,% \aa
        \glshex 00C5=A\glshex 030A% \AA
    }

\glxtrLatinWynn
    \newcommand*\glxtrLatinWynn{%
        \glshex 01BF,\glshex 01F7% wynn
    }

\glxtrLatinInsularG
    \newcommand*\glxtrLatinInsularG{%
        \glshex 1D79,\glshex A77D% insular G
        \string; g, G
    }

\glxtrLatinOslash
    \newcommand*\glxtrLatinOslash{%
        \glshex 00F8,\glshex 00D8% \o, \O
    }

```

```

\glxtrLatinLslash
\newcommand*{\glxtrLatinLslash}{%
\glshex 0142,\glshex 0141% \l, \L
}

```

`\glxtrMathUpGreekIrules` Includes digamma between epsilon and zeta.

```

\newcommand*{\glxtrMathUpGreekIrules}{%
\glxtrUpAlpha
\string<\glxtrUpBeta
\string<\glxtrUpGamma
\string<\glxtrUpDelta
\string<\glxtrUpEpsilon
\string<\glxtrUpDigamma
\string<\glxtrUpZeta
\string<\glxtrUpEta
\string<\glxtrUpTheta
\string<\glxtrUpIota
\string<\glxtrUpKappa
\string<\glxtrUpLambda
\string<\glxtrUpMu
\string<\glxtrUpNu
\string<\glxtrUpXi
\string<\glxtrUpOmicron
\string<\glxtrUpPi
\string<\glxtrUpRho
\string<\glxtrUpSigma
\string<\glxtrUpTau
\string<\glxtrUpUpsilon
\string<\glxtrUpPhi
\string<\glxtrUpChi
\string<\glxtrUpPsi
\string<\glxtrUpOmega
}

```

`\glxtrMathUpGreekIIrules` Doesn't include digamma.

```

\newcommand*{\glxtrMathUpGreekIIrules}{%
\glxtrUpAlpha
\string<\glxtrUpBeta
\string<\glxtrUpGamma
\string<\glxtrUpDelta
\string<\glxtrUpEpsilon
\string<\glxtrUpZeta
\string<\glxtrUpEta
\string<\glxtrUpTheta
\string<\glxtrUpIota
\string<\glxtrUpKappa
\string<\glxtrUpLambda
\string<\glxtrUpMu
\string<\glxtrUpNu

```

```

\string<\glxtrUpXi
\string<\glxtrUpOmicron
\string<\glxtrUpPi
\string<\glxtrUpRho
\string<\glxtrUpSigma
\string<\glxtrUpTau
\string<\glxtrUpUpsilon
\string<\glxtrUpPhi
\string<\glxtrUpChi
\string<\glxtrUpPsi
\string<\glxtrUpOmega
}

```

`\glxtrMathItalicGreekIrules` Includes (upright) digamma between epsilon and zeta (there isn't an italic digamma), so don't mix with `\glxtrMathUpGreekIrules` or there may be unexpected results.

```

\newcommand*{\glxtrMathItalicGreekIrules}{%
\glxtrMathItalicAlpha
\string<\glxtrMathItalicBeta
\string<\glxtrMathItalicGamma
\string<\glxtrMathItalicDelta
\string<\glxtrMathItalicEpsilon
\string<\glxtrUpDigamma
\string<\glxtrMathItalicZeta
\string<\glxtrMathItalicEta
\string<\glxtrMathItalicTheta
\string<\glxtrMathItalicIota
\string<\glxtrMathItalicKappa
\string<\glxtrMathItalicLambda
\string<\glxtrMathItalicMu
\string<\glxtrMathItalicNu
\string<\glxtrMathItalicXi
\string<\glxtrMathItalicOmicron
\string<\glxtrMathItalicPi
\string<\glxtrMathItalicRho
\string<\glxtrMathItalicSigma
\string<\glxtrMathItalicTau
\string<\glxtrMathItalicUpsilon
\string<\glxtrMathItalicPhi
\string<\glxtrMathItalicChi
\string<\glxtrMathItalicPsi
\string<\glxtrMathItalicOmega
}

```

`\glxtrMathItalicGreekIIrules` Doesn't include digamma.

```

\newcommand*{\glxtrMathItalicGreekIIrules}{%
\glxtrMathItalicAlpha
\string<\glxtrMathItalicBeta
\string<\glxtrMathItalicGamma

```

```

\string<\glxtrMathItalicDelta
\string<\glxtrMathItalicEpsilon
\string<\glxtrMathItalicZeta
\string<\glxtrMathItalicEta
\string<\glxtrMathItalicTheta
\string<\glxtrMathItalicIota
\string<\glxtrMathItalicKappa
\string<\glxtrMathItalicLambda
\string<\glxtrMathItalicMu
\string<\glxtrMathItalicNu
\string<\glxtrMathItalicXi
\string<\glxtrMathItalicOmicron
\string<\glxtrMathItalicPi
\string<\glxtrMathItalicRho
\string<\glxtrMathItalicSigma
\string<\glxtrMathItalicTau
\string<\glxtrMathItalicUpsilon
\string<\glxtrMathItalicPhi
\string<\glxtrMathItalicChi
\string<\glxtrMathItalicPsi
\string<\glxtrMathItalicOmega
}

```

`\glxtrMathItalicUpperGreekIrules` Upper case only (includes upright digamma).

```

\newcommand*{\glxtrMathItalicUpperGreekIrules}{%
\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)
}

```

MathItalicUpperGreekIIrules Upper case only (doesn't include upright digamma).

```

\newcommand*{\glxtrMathItalicUpperGreekIIrules}{%
\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)
}

```

MathItalicLowerGreekIrules Lower case only (includes upright digamma).

```

\newcommand*{\glxtrMathItalicLowerGreekIrules}{%
\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*{\glxtrMathItalicLowerGreekIIrules}{%
\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)
}

```

`\glxtrMathGreekIrules` Includes both upright and italic with digamma between epsilon and zeta.

```

\newcommand*{\glxtrMathGreekIrules}{%
\glxtrMathItalicAlpha
\string;\glxtrUpAlpha
\string<\glxtrMathItalicBeta
\string;\glxtrUpBeta
\string<\glxtrMathItalicGamma
\string;\glxtrUpGamma
\string<\glxtrMathItalicDelta
\string;\glxtrUpDelta
\string<\glxtrMathItalicEpsilon
\string;\glxtrUpEpsilon
\string<\glxtrUpDigamma
\string<\glxtrMathItalicZeta
\string;\glxtrUpZeta
\string<\glxtrMathItalicEta
\string;\glxtrUpEta
\string<\glxtrMathItalicTheta
\string;\glxtrUpTheta
\string<\glxtrMathItalicIota
\string;\glxtrUpIota
\string<\glxtrMathItalicKappa
\string;\glxtrUpKappa
\string<\glxtrMathItalicLambda
\string;\glxtrUpLambda
\string<\glxtrMathItalicMu
\string;\glxtrUpMu
\string<\glxtrMathItalicNu
\string;\glxtrUpNu
\string<\glxtrMathItalicXi
\string;\glxtrUpXi
\string<\glxtrMathItalicOmicron
\string;\glxtrUpOmicron
\string<\glxtrMathItalicPi
\string;\glxtrUpPi
\string<\glxtrMathItalicRho
\string;\glxtrUpRho
\string<\glxtrMathItalicSigma
\string;\glxtrUpSigma
\string<\glxtrMathItalicTau
\string;\glxtrUpTau
\string<\glxtrMathItalicUpsilon
\string;\glxtrUpUpsilon
\string<\glxtrMathItalicPhi
\string;\glxtrUpPhi
\string<\glxtrMathItalicChi
\string;\glxtrUpChi
}

```

```

\string<\glxtrMathItalicPsi
\string;\glxtrUpPsi
\string<\glxtrMathItalicOmega
\string;\glxtrUpOmega
}

```

`\glxtrMathGreekIIrules` Includes both upright and italic (digamma not included).

```

\newcommand*{\glxtrMathGreekIIrules}{%
\glxtrMathItalicAlpha
\string;\glxtrUpAlpha
\string<\glxtrMathItalicBeta
\string;\glxtrUpBeta
\string<\glxtrMathItalicGamma
\string;\glxtrUpGamma
\string<\glxtrMathItalicDelta
\string;\glxtrUpDelta
\string<\glxtrMathItalicEpsilon
\string;\glxtrUpEpsilon
\string<\glxtrMathItalicZeta
\string;\glxtrUpZeta
\string<\glxtrMathItalicEta
\string;\glxtrUpEta
\string<\glxtrMathItalicTheta
\string;\glxtrUpTheta
\string<\glxtrMathItalicIota
\string;\glxtrUpIota
\string<\glxtrMathItalicKappa
\string;\glxtrUpKappa
\string<\glxtrMathItalicLambda
\string;\glxtrUpLambda
\string<\glxtrMathItalicMu
\string;\glxtrUpMu
\string<\glxtrMathItalicNu
\string;\glxtrUpNu
\string<\glxtrMathItalicXi
\string;\glxtrUpXi
\string<\glxtrMathItalicOmicron
\string;\glxtrUpOmicron
\string<\glxtrMathItalicPi
\string;\glxtrUpPi
\string<\glxtrMathItalicRho
\string;\glxtrUpRho
\string<\glxtrMathItalicSigma
\string;\glxtrUpSigma
\string<\glxtrMathItalicTau
\string;\glxtrUpTau
\string<\glxtrMathItalicUpsilon
\string;\glxtrUpUpsilon
\string<\glxtrMathItalicPhi
\string;\glxtrUpPhi
}

```



```

\string<\glxtrMathItalicChi
\string;\glxtrUpChi
\string<\glxtrMathItalicPsi
\string;\glxtrUpPsi
\string<\glxtrMathItalicOmega
\string;\glxtrUpOmega
}

\glxtrUpAlpha
\newcommand*{\glxtrUpAlpha}{%
\glshex 03B1,% lower case alpha
\glshex 0391% upper case alpha
}

\glxtrUpBeta
\newcommand*{\glxtrUpBeta}{%
\glshex 03B2,% lower case beta
\glshex 0392% upper case beta
}

\glxtrUpGamma
\newcommand*{\glxtrUpGamma}{%
\glshex 03B3,% lower case gamma
\glshex 0393% upper case gamma
}

\glxtrUpDelta
\newcommand*{\glxtrUpDelta}{%
\glshex 03B4,% lower case delta
\glshex 0394% upper case delta
}

\glxtrUpEpsilon
\newcommand*{\glxtrUpEpsilon}{%
\glshex 03B5% lower case epsilon
\string=\glshex 03F5,% lower case epsilon variant
\glshex 0395% upper case epsilon
}

\glxtrUpDigamma
\newcommand*{\glxtrUpDigamma}{%
\glshex 03DD,% lower case digamma
\glshex 03DC% upper case digamma
}

\glxtrUpZeta
\newcommand*{\glxtrUpZeta}{%
\glshex 03B6,% lower case zeta
\glshex 0396% upper case zeta
}

```

```

\glxtrUpEta
\newcommand*\glxtrUpEta}{%
\glshex 03B7,% lower case eta
\glshex 0397% upper case eta
}

\glxtrUpTheta
\newcommand*\glxtrUpTheta}{%
\glshex 03B8% lower case theta
\string=\glshex 03D1,% lower case theta variant
\glshex 0398% upper case theta
}

\glxtrUpIota
\newcommand*\glxtrUpIota}{%
\glshex 03B9,% lower case iota
\glshex 0399% upper case iota
}

\glxtrUpKappa
\newcommand*\glxtrUpKappa}{%
\glshex 03BA% lower case kappa
\string=\glshex 03F0,% lower case kappa variant
\glshex 039A% upper case kappa
}

\glxtrUpLambda
\newcommand*\glxtrUpLambda}{%
\glshex 03BB,% lower lambda
\glshex 039B% upper case lambda
}

\glxtrUpMu
\newcommand*\glxtrUpMu}{%
\glshex 03BC,% lower case mu
\glshex 039C% upper case mu
}

\glxtrUpNu
\newcommand*\glxtrUpNu}{%
\glshex 03BD,% lower case nu
\glshex 039D% upper case nu
}

\glxtrUpXi
\newcommand*\glxtrUpXi}{%
\glshex 03BE,% lower case xi
\glshex 039E% upper case xi
}

```

```

\glxtrUpOmicron
\newcommand*\glxtrUpOmicron}{%
\glshex 03BF,% lower case omicron
\glshex 039F% upper case omicron
}

\glxtrUpPi
\newcommand*\glxtrUpPi}{%
\glshex 03C0% lower case pi
\string=\glshex 03D6,% lower case pi variant
\glshex 03A0% upper case pi
}

\glxtrUpRho
\newcommand*\glxtrUpRho}{%
\glshex 03C1% lower case rho
\string=\glshex 03F1,% lower case rho variant
\glshex 03A1% upper case rho
}

\glxtrUpSigma
\newcommand*\glxtrUpSigma}{%
\glshex 03C2% lower case sigma
\string=\glshex 03C3,% lower case sigma
\glshex 03A3% upper case sigma
}

\glxtrUpTau
\newcommand*\glxtrUpTau}{%
\glshex 03C4,% lower case tau
\glshex 03A4% upper case tau
}

\glxtrUpUpsilon
\newcommand*\glxtrUpUpsilon}{%
\glshex 03C5,% lower case upsilon
\glshex 03A5% upper case upsilon
}

\glxtrUpPhi
\newcommand*\glxtrUpPhi}{%
\glshex 03C6% lower case phi
\string=\glshex 03D5,% lower case phi variant
\glshex 03A6% upper case phi
}

\glxtrUpChi
\newcommand*\glxtrUpChi}{%

```

```

        \glshex 03C7,% lower case chi
        \glshex 03A7% upper case chi
    }

\glxtrUpPsi
    \newcommand*{\glxtrUpPsi}{%
        \glshex 03C8,% lower case psi
        \glshex 03A8% upper case psi
    }

\glxtrUpOmega
    \newcommand*{\glxtrUpOmega}{%
        \glshex 03C9,% lower case omega
        \glshex 03A9% upper case omega
    }

\glxtrMathItalicAlpha
    \newcommand*{\glxtrMathItalicAlpha}{%
        \glshex 1D6FC,% lower case alpha (maths italic)
        \glshex 1D6E2% upper case alpha (maths italic)
    }

\glxtrMathItalicBeta
    \newcommand*{\glxtrMathItalicBeta}{%
        \glshex 1D6FD,% lower case beta (maths italic)
        \glshex 1D6E3% upper case beta (maths italic)
    }

\glxtrMathItalicGamma
    \newcommand*{\glxtrMathItalicGamma}{%
        \glshex 1D6FE,% lower case gamma (maths italic)
        \glshex 1D6E4% upper case gamma (maths italic)
    }

\glxtrMathItalicDelta
    \newcommand*{\glxtrMathItalicDelta}{%
        \glshex 1D6FF,% lower case delta (maths italic)
        \glshex 1D6E5% upper case delta (maths italic)
    }

\glxtrMathItalicEpsilon
    \newcommand*{\glxtrMathItalicEpsilon}{%
        \glshex 1D700% lower case epsilon (maths italic)
        \string=\glshex 1D716,% lower case epsilon variant (maths italic)
        \glshex 1D6E6% upper case epsilon (maths italic)
    }

```

```

\glxtrMathItalicZeta
\newcommand*\glxtrMathItalicZeta}{%
\glshex 1D701,% lower case zeta (maths italic)
\glshex 1D6E7% upper case zeta (maths italic)
}

\glxtrMathItalicEta
\newcommand*\glxtrMathItalicEta}{%
\glshex 1D702,% lower case eta (maths italic)
\glshex 1D6E8% upper case eta (maths italic)
}

\glxtrMathItalicTheta
\newcommand*\glxtrMathItalicTheta}{%
\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)
}

\glxtrMathItalicIota
\newcommand*\glxtrMathItalicIota}{%
\glshex 1D704,% lower case iota (maths italic)
\glshex 1D6EA% upper case iota (maths italic)
}

\glxtrMathItalicKappa
\newcommand*\glxtrMathItalicKappa}{%
\glshex 1D705% lower case kappa (maths italic)
\string=\glshex 1D718,% lower case kappa variant (maths italic)
\glshex 1D6EB% upper case kappa (maths italic)
}

\glxtrMathItalicLambda
\newcommand*\glxtrMathItalicLambda}{%
\glshex 1D706,% lower case lambda (maths italic)
\glshex 1D6EC% upper case lambda (maths italic)
}

\glxtrMathItalicMu
\newcommand*\glxtrMathItalicMu}{%
\glshex 1D707,% lower case mu (maths italic)
\glshex 1D6ED% upper case mu (maths italic)
}

\glxtrMathItalicNu
\newcommand*\glxtrMathItalicNu}{%
\glshex 1D708,% lower case nu (maths italic)
\glshex 1D6EE% upper case nu (maths italic)
}

```

```

\glxtrMathItalicXi
\newcommand*\glxtrMathItalicXi{%
  \glshex 1D709,% lower case xi (maths italic)
  \glshex 1D6EF% upper case xi (maths italic)
}

\glxtrMathItalicOmicron
\newcommand*\glxtrMathItalicOmicron{%
  \glshex 1D70A,% lower case omicron (maths italic)
  \glshex 1D6F0% upper case omicron (maths italic)
}

\glxtrMathItalicPi
\newcommand*\glxtrMathItalicPi{%
  \glshex 1D70B% lower case pi (maths italic)
  \string=\glshex 1D71B,% lower case pi variant (maths italic)
  \glshex 1D6F1% upper case pi (maths italic)
}

\glxtrMathItalicRho
\newcommand*\glxtrMathItalicRho{%
  \glshex 1D70C% lower case rho (maths italic)
  \string=\glshex 1D71A,% lower case rho variant (maths italic)
  \glshex 1D6F2% upper case rho (maths italic)
}

\glxtrMathItalicSigma
\newcommand*\glxtrMathItalicSigma{%
  \glshex 1D70D% lower case final sigma (maths italic)
  \string=\glshex 1D70E,% lower case sigma (maths italic)
  \glshex 1D6F4% upper case sigma (maths italic)
}

\glxtrMathItalicTau
\newcommand*\glxtrMathItalicTau{%
  \glshex 1D70F,% lower case tau (maths italic)
  \glshex 1D6F5% upper case tau (maths italic)
}

\glxtrMathItalicUpsilon
\newcommand*\glxtrMathItalicUpsilon{%
  \glshex 1D710,% lower case upsilon (maths italic)
  \glshex 1D6F6% upper case upsilon (maths italic)
}

\glxtrMathItalicPhi
\newcommand*\glxtrMathItalicPhi{%
  \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)
}

```

`\glxtrdigitrules` Digits from the Basic Latin set and subscript and superscript digit rules.

```

\newcommand*{\glxtrdigitrules}{%
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%
}

```

`\glxtrSubScriptDigitrules` Subscript digits.

```

\newcommand*{\glxtrSubScriptDigitrules}{%
\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
}

```

`\glxtrSuperScriptDigitrules` Superscript digits.

```

\newcommand*{\glxtrSuperScriptDigitrules}{%
\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
}

```

`\glxtrfractionrules` Vulgar fractions.

```

\newcommand*{\glxtrfractionrules}{%
\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)
}

```

`\@glxtrdialecthook` Check for scripts associated with the document dialects.

```

\renewcommand{\@glxtrdialecthook}{%
\ifundef\CurrentTrackedScript
{%
\TrackLangIfHasDefaultScript{\CurrentTrackedLanguage}%
{%
\edef\CurrentTrackedScript{%
\TrackLangGetDefaultScript\CurrentTrackedLanguage}%
}%
}%%
}%%
\ifdef\CurrentTrackedScript
{%
\let\gls@orgTrackLangRequireDialectPrefix\TrackLangRequireDialectPrefix
\def\TrackLangRequireDialectPrefix{glossariesxtr-}%
\let\CurrentTrackedTag\CurrentTrackedScript
\IfFileExists{\TrackLangRequireDialectPrefix\CurrentTrackedTag.1df}
{\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 description 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{v1.49}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossaries-extra-stylemods}[2022/10/14 v1.49 (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}%
  }
}

\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:

```
\@glxstrloadstyles
```

Adjust the styles so that they all have the post description hook. Also, instead of having a hard-coded `\space` before the location, use:

`\glxstrprelocation` This uses `\providecommand` as the same command is also provided by `glossary-bookindex`.

```
\providecommand*{\glxstrprelocation}{\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}%
  }%
}

```

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]{%
        \glstryitem{##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}{\ifglsgroupskip\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.

```
\ifcsdef{\@glsstyle@altlistgroup}
{%
  \renewglossarystyle{altlistgroup}{%
    \setglossarystyle{altlist}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glslistgroupheaderitem{##1}%
      {\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}}%
    \glslistgroupafterheader
  }%
}
}
```

Similarly for altlisthypergroup.

```
\ifcsdef{\@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 `\glstrprelocation`.

```
\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
}
```

```

        \glstrprelocation ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
        \glstrprelocation ##3\tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}
}
{}

```

Three column style:

```

\ifcsdef{@glstyle@long3col}
{%
    \renewglossarystyle{long3col}{%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glstdescwidth}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]{%
        \glssubentryitem{##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}{l l l l}}%
      {\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 `\glxtrprelocation`.

```
\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]{%
  \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\glstrprelocation ##2%
  \tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}%
  \glspostdescription\glstrprelocation ##3%
  \tabularnewline
}%
\ifglsgroupskip
  \renewcommand*\glsgroupskip}{}%
\else
  \renewcommand*\glsgroupskip}{ & \tabularnewline}%
\fi
}
}
{}
```

Three and four column styles don't use \glstrprelocation since the number list is in its own column.

```
\ifcsdef{@glstyle@longragged3col}
{%
\renewglossarystyle{longragged3col}{%
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}%
  >{\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]{%
  \glentryitem{##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{@glstyle@altlongragged4col}
{%
\renewglossarystyle{altlongragged4col}{%
\renewenvironment{theglossary}%
{\begin{longtable}{l>{\raggedright}p{\glsglwidth}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]{%
\glstryitem{##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 `\glxtrprelocation`.

```

\ifcsdef{@glstyle@super}

```

```

{%
\renewglossarystyle{super}{%
\renewenvironment{theglossary}%
{\tablehead{}\tabletail{}}%
\begin{supertabular}{lp{\glstdescwidth}}%
{\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]{%
\glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1}\glspostdescription
\glstrprelocation ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
\glstrprelocation ##3\tabularnewline
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
}
{}

```

Three column style:

```

\ifcsdef{@glstyle@super3col}
{%
\renewglossarystyle{super3col}{%
\renewenvironment{theglossary}%
{\tablehead{}\tabletail{}}%
\begin{supertabular}{lp{\glstdescwidth}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]{%
\glstryitem{##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{@glstyle@super4col}
{%
  \renewglossarystyle{super4col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}}%
      \begin{supertabular}{1111}}{%
        \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]{%
      \glentryitem{##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 `\glxstrprelocation`.

```
\ifcsdef{@glstyle@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]{%
  \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\glxstrprelocation ##2%
  \tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
  \glxstrprelocation ##3%
  \tabularnewline
}%
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
}
{}

```

Three column style:

```
\ifcsdef{@glstyle@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]{%
  \glssentryitem{##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{@glstyle@altsuperragged4col}
{%
  \renewglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}}%
      \begin{supertabular}{1>{\raggedright}p{\glwdescwidth}1%
        >{\raggedright}p{\glspagelistwidth}}}%
      {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand{\glossentry}[2]{%
      \glssentryitem{##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
  }
}

```

```

\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\sffcode{\.}}

```

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{\<previous group level>}{\<level>}
{\<parent label>}{\<group label>}{\<title>}}
```

`\glstreeSubPreHeader`

```
\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}{\glstrprelocation}
```

`\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\glentryitem{##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{\ifglsgnogroupskip\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>}
```

`\glsindexsubgroupitem`

```
\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]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\item\glstreegroupheaderfmt{\glstr@grptitle}%
\glstreegroupheaderskip\@afterheading
}%
Sub-groups are only supported with \printunsrtglossary.
\renewcommand*\glssubgroupheading}[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glindexsubgroupitem{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\@afterheading
}%
}
}
{}

Similarly for indexhypergroup.
\ifdef{\@glstyle@indexhypergroup}
{%
\renewglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*\glossaryheader{%
\item\glstreenavigationfmt{\glsnavigation}%
\glstreegroupheaderskip\@afterheading}%
Group heading.
\renewcommand*\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glstr@grptitle}}}%
\glstreegroupheaderskip\@afterheading}%
Sub-groups are only supported with \printunsrtglossary.
\renewcommand*\glssubgroupheading}[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glindexsubgroupitem{##1}{##2}{##3}{##4}%
{\glsnavhypertarget{##4}{\glstr@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`.

```

\glxtrtreepredesc

```

```

\newcommand{\glxtrtreepredesc}{\glstreepredesc}

```

```

\glxtrtreechildpredesc

```

```

\newcommand{\glxtrtreechildpredesc}{\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]{%
  \glxtrtreepredesc\glossentrydesc{#1}\glspostdescription
}

```

```

\glstreeDescLoc{<label>}{<location>}

```

```

\glstreeDescLoc

```

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

```

```

\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]{%
  \glstxttreechildpredesc\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
  \glstryitem{##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
}
```

```

\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\glstreechildsymbol{##2}%
\glstreeChildDescLoc{##2}{##3}\par
}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\glstreegroupskip\fi}%
}%
}
{}

```

The `treegroup` style is redefined to discourage a page break after the heading.

```

\ifdef{\@glstyle@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]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent\glstreegroupheaderfmt{\glstr@grptitle}%
\glstreegroupheaderskip\@afterheading}%

```

Sub-groups are only supported with `\printunsrtglossary`.

```

\renewcommand*{\glssubgroupheading}[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glstreesubgroupitem{##1}{##2}{##3}{##4}{\glstr@grptitle}%
}%
}
}
{}

```

Similarly for `treehypergroup`

```

\ifdef{\@glstyle@treehypergroup}
{%

```

```

\renewglossarystyle{treehypergroup}{%
\setglossarystyle{tree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}%
\glstreegroupheaderskip\@afterheading}%

```

Group heading.

```

\renewcommand*{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glstr@grptitle}}%
\glstreegroupheaderskip\@afterheading}%

```

Sub-groups are only supported with `\printunsrtglossary`.

```

\renewcommand*{\glssubgroupheading}[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glstreesubgroupitem{##1}{##2}{##3}{##4}%
{\glsnavhypertarget{##4}{\glstr@grptitle}}%
}%
}
}
{}

```

Adjust `treenoname` style to remove hard coded space before number list.

```

\ifdef{\@glstyle@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
  \glstryitem{##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{\@glstyle@treenonamegroup}
{%
  \renewglossarystyle{treenonamegroup}{%
    \setglossarystyle{treenoname}%
    \renewcommand{\glsgroupheading}[1]{%

```



```

\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent\glstreegroupheaderfmt{\glstr@grptitle}%
\glstreegroupheaderskip\@afterheading
}%
}
}
{}

Similarly for treenonamehypergroup
\ifdef{\@glstyle@treenonamehypergroup}
{%
\renewglossarystyle{treenonamehypergroup}{%
\setglossarystyle{treenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}%
\glstreegroupheaderskip\@afterheading}%
\renewcommand*{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@grptitle}}%
\glstreegroupheaderskip\@afterheading}%
}
}
}
{}

```

The `alttree` style is redefined to make it easier to make minor adjustments.

```

\ifdef{\@glstyle@alttree}
{%

```

Only redefine this style if it's already been defined.

```

\glsalttreepredesc

```

```

\newcommand{\glsalttreepredesc}{%

```

```

\glsalttreechildpredesc

```

```

\newcommand{\glsalttreechildpredesc}{\glsalttreepredesc}

```

```

\glxtralttreeSymbolDescLocation{<label>}{<location
list>}

```

```

\glxtralttreeSymbolDescLocation

```

Layout the symbol, description and location for top-level entries.

```

\newcommand{\glxtralttreeSymbolDescLocation}[2]{%
{%
\let\par\glxtrAltTreePar

\let\glxtrtreepredesc\glsalttreepredesc
\let\glxtrtreechildpredesc\glsalttreechildpredesc
\ifglshassymbol{#1}{(\glossentrysymbol{#1})\space}{}%

```

```

\glstreeDescLoc{#1}{#2}\par
}%
}

```

`\glxtrAltTreeIndent` Paragraph indent for subsequent paragraphs in multi-paragraph descriptions.

```

\newlength\glxtrAltTreeIndent

```

`\glxtrAltTreePar` Multi-paragraph descriptions need to keep the hanging indent.

```

\newcommand{\glxtrAltTreePar}{%
  @@par
  \glxtrAltTreeSetHangIndent
  \setlength{\parindent}{\dimexpr\hangindent+\glxtrAltTreeIndent}%
}

```

```

\glxtraltrtreeSubSymbolDescLocation{<level>}{<label>}
{<location
list>}

```

`\glxtraltrtreeSubSymbolDescLocation`

Layout the symbol, description and location for sub-entries. Defaults to the same as the top-level.

```

\newcommand{\glxtraltrtreeSubSymbolDescLocation}[3]{%
  \glxtraltrtreeSymbolDescLocation{#2}{#3}%
}

```

`\glxtrtreetopindent` 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\glxtrtreetopindent

```

`\glxtraltrtreeInit` User-level initialisation for the `altrtree` style.

```

\newcommand*{\glxtraltrtreeInit}{%
  \settowidth{\glxtrtreetopindent}{\glstreenamfmt{\glsgetwidestname\space}}%
  \glxtrAltTreeIndent=\parindent
}

```

`\gglsetwidest` The original `\glsetwidest` only uses `\def`. This uses `\gdef`.

```

\newcommand*{\gglsetwidest}[2][0]{%
  \csgdef{@glswidestname\romannumeral#1}{#2}%
}

```

`\eglsetwidest` The original `\glsetwidest` only uses `\def`. This uses `\protected@csedef`.

```

\newcommand*{\eglsetwidest}[2][0]{%
  \protected@csedef{@glswidestname\romannumeral#1}{#2}%
}

```

`\xglissetwidest` Like the above but uses `\protected@csxdef`.

```
\newcommand*{\xglissetwidest}[2][0]{%
  \protected@csxdef{@glswidestname\romannumeral#1}{#2}%
}
```

`\glsupdatewidest` Only sets if new value is wider than old value.

```
\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
  }%
}
```

`\gglsupdatewidest` As above but global definition.

```
\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
  }%
}
```

`\eglsupdatewidest` As `\glsupdatewidest` but expands value.

```
\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
  }%
}
```

`\xglsupdatewidest` As above but global.

```
\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
}%
}

\glswidestname Provide a user-level macro to obtain the widest top-level name.
\newcommand*{\glswidestname}{\@glswidestname}

\glswidestsubname Provide a user-level macro to obtain the widest sub-entry name.
\newcommand*{\glswidestsubname}[1]{%
\ifcsundef{@glswidestname\romannumeral#1}%
{\@glswidestname}%
{\csuse{@glswidestname\romannumeral#1}}%
}

\glswidesttoplevelname CamelCase is easier for long command names. Provide a CamelCase synonym
of \glswidesttoplevelname.
\let\glswidesttoplevelname\glswidesttoplevelname

\glswidestusedtoplevelname Like \glswidesttoplevelname 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*{\glswidestusedtoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\glstmplen=0pt\relax
\forallglossaries[#1]{\@gls@type}%
{%
\forallglsentries[\@gls@type]{\@glo@label}%
{%
\ifglsused{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
{}}%
}%
\settowidth{\dimen@}%
{\glstreenamfmt{\glstentryname{\@glo@label}}}%
\ifdim\dimen@>\glstmplen
\glstmplen=\dimen@
\eglswidest{\glstentryname{\@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
  \foralllglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \settowidth{\dimen@}%
          {\glstreenamfmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \eglssetwidest{\glsentryname{\@glo@label}}%
        \fi
      }%
    }%
  }%
}
```

`\glsFindWidestAnyName` Like the above but doesn't check if the entry has been used.

```
\newrobustcmd*{\glsFindWidestAnyName}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \foralllglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \settowidth{\dimen@}%
        {\glstreenamfmt{\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
  \foralllglossaries[#1]{\@gls@type}%
  {%
```



```

\dimen@ii=0pt\relax
\foralllglossaries[#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
  }%
}%
}

```

\FindWidestUsedAnyNameSymbol 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
\foralllglossaries[#1]{\@gls@type}%
{%
  \forglsentries[\@gls@type]{\@glo@label}%
  {%
    \ifglsused{\@glo@label}%
    {%
      \settowidth{\dimen@}%
        {\glstreenamfmt{\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
    }%
  }%
}%
}

```

`\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
  \foralllglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \settowidth{\dimen@}%
        {\glstreenamfmt{\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
    }%
  }%
}%
}

```

`\glsFindWidestUsedAnyNameSymbolLocation` 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
  \foralllglossaries[#1]{\@gls@type}%
  {%
    \forallglsentries[\@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
      }%
    }%
  }%
}
```

\WidestAnyNameSymbolLocation 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
  \foralllglossaries[#1]{\@gls@type}%
  {%
    \forallglsentries[\@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*{\glsFindWidestUsedAnyNameSymbolLocation}` 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
  \foralllglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglsused{\@glo@label}%
      {%
        \settowidth{\dimen@}%
          {\glstreenamfmt{\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
      }%
    }%
  }%
}%
}

```

`\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
  \foralllglossaries[#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
    }%
  }%
}

```

`\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
}

```

`\glsxtrComputeTreeSubIndent{<level>}{<label>}{<register>}`

`\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}%
  {%
    \settowidth{#3}{\glstreenamefmt{\@glswidestname\space}}%
  }%
  {%
    \settowidth{#3}{\glstreenamefmt{%
      \csname @glswidestname\romannumeral#1\endcsname\space}}%
  }%
}

```

`\glsxtrAltTreeSetHangIndent` Set `\hangindent` for top-level entries:

```

\newcommand*{\glsxtrAltTreeSetHangIndent}{\hangindent\glstreeindent}

```

`glxtrAltTreeSetSubHangIndent` Set `\hangindent` for sub-entries:

```
\newcommand*{\glxtrAltTreeSetSubHangIndent}[1]{\hangindent\glstreeindent}
```

Redefine `alttree`:

```
\renewglossarystyle{alttree}{%
  \renewenvironment{theglossary}%
    {%
      \glxtralttreeInit
      \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
    \glxtrComputeTreeIndent{##1}%
  \fi
  \parindent\glstreeindent
  \glxtrAltTreeSetSubHangIndent
  \makebox[0pt][r]{%
    {%
      \glstreenamibox{\glstreeindent}%
      {%
        \glsentryitem{##1}%
        \glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
      }%
    }%
  \glxtralttreeSymbolDescLocation{##1}{##2}%
  \def\@gls@prevlevel{0}%
}
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \glxtrComputeTreeSubIndent{##1}{##2}{\gls@tmplen}%
    \ifnum\@gls@prevlevel<##1\relax
      \setlength\glstreeindent\gls@tmplen
      \addtolength\glstreeindent\parindent
      \parindent\glstreeindent
    \else
      \ifnum\@gls@prevlevel=0\relax
        \glxtrComputeTreeIndent{##2}%
      \else
        \glxtrComputeTreeSubIndent{\@gls@prevlevel}{##2}{\glstreeindent}%
      \fi
    \fi
  \fi
}
```

```

\fi
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\glstrAltTreeSetSubHangIndent{##1}%
\makebox[0pt][r]{\glstreenamebox{\glstmpLen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\glstrAltTreeSubSymbolDescLocation{##1}{##2}{##3}%
\def\@glsprevlevel{##1}%
}%
\renewcommand*\@glsgroupskip{\ifglsgnোগroupskip\else\glstreegroupskip\fi}%
}
}%
{%
}

```

Redefine `alttreegroup` so that it discourages a break after group headings.

```

\ifdef{\@glstyle@alttreegroup}
{

```

```

\glsalttreesubgroupheader{\<previous group level>}{\<group
level>}{\<parent label>}{\<group label>}{\<title>}{\<width>}

```

`\glsalttreesubgroupitem`

```

\newcommand*\@glsalttreesubgroupheader[6]{%
\par\smallskip
\makebox[0pt][r]{\glstreenamebox{#6}%
{\glstreegroupheaderfmt{#5}}}%
\smallskip\par
}

\renewglossarystyle{alttreegroup}{%
\setglossarystyle{alttree}%
\renewcommand{\@glsgroupheading}[1]{\par
\def\@glsprevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\glstreegroupheaderfmt{\glstr@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]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%

```

This is similar to `\subglossentry`

```

\ifnum\@gls@prevlevel=##2\relax
\else
\ifcsundef{@glswidestname\romannumeral##2}%
{%
\settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
}%
{%
\settowidth{\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
\glstrComputerTreeIndent{##2}%
\else
\ifcsundef{@glswidestname\romannumeral##2}%
{%
\settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
}%
{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral##2\endcsname\space}}%
}%
\fi
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\glstrAltTreeSetSubHangIndent{##2}%
\glsalttreesubgroupheader{##1}{##2}{##3}{##4}{\glstr@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]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glstr@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]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%

```

This is similar to \subglossentry

```

\ifnum\@gls@prevlevel=##2\relax
\else
\ifcsundef{\glswidestname\romannumeral##2}%
{%
\settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
}%
{%
\settowidth{\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
\glstrComputeTreeIndent{##2}%
\else
\ifcsundef{\glswidestname\romannumeral##2}%
{%
\settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
}%
{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral##2\endcsname\space}}%

```

```

        }%
    \fi
    \addtolength\parindent{-\glstreeindent}%
    \setlength\glstreeindent\parindent
\fi
\fi
\glstrAltTreeSetSubHangIndent{##2}%
\glstreesubgroupheader{##1}{##2}{##3}{##4}%
    {\glshypertarget{##4}{\glstr@grptitle}}{\glstmplen}%
\def\@glsprevlevel{##2}%
\par
}%
}
}%
{%
}

```

4.9 Multicolumn Styles

Adjust `mcolindexgroup` to discourage page breaks after the group headings.

```

\ifdef{\@glstyle@mcolindexgroup}
{%
    \renewglossarystyle{mcolindexgroup}{%
        \setglossarystyle{mcolindex}%

```

Group heading as indexgroup.

```

    \renewcommand*\glsgroupheading}[1]{%
        \glstrgetgrouptitle{##1}{\glstr@grptitle}%
        \glstreePreHeader{##1}{\glstr@grptitle}%
        \item\glstreegroupheaderfmt{\glstr@grptitle}%
        \glstreegroupheaderskip\@afterheading
    }%

```

Sub-groups are only supported with `\printunsrtglossary`.

```

    \renewcommand*\glssubgroupheading}[4]{%
        \glstrgetgrouptitle{##4}{\glstr@grptitle}%
        \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
        \glindexsubgroupitem{##1}{##2}{##3}{##4}{\glstr@grptitle}%
        \@afterheading
    }%
}
}%
{%
}

```

Similarly for `mcolindexhypergroup`.

```

\ifdef{\@glstyle@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}{\glsncols}\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]{%

```

```

\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glstreesubgroupitem{##1}{##2}{##3}{##4}%
    {\glsnavigationtarget{##4}{\glstr@grptitle}}%
\@afterheading
}%
}
}%
{%
}

```

Similarly for mcoltreegroup.

```

\ifdef{\@glstyle@mcoltreegroup}
{%
\renewglossarystyle{mcoltreegroup}{%
\setglossarystyle{mcoltree}%

```

Group heading.

```

\renewcommand{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent\glstreegroupheaderfmt{\glstr@grptitle}%
\glstreegroupheaderskip\@afterheading
}%

```

Sub-groups are only supported with \printunsrtglossary.

```

\renewcommand*{\glssubgroupheading}[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glstreesubgroupitem{##1}{##2}{##3}{##4}{\glstr@grptitle}%
}%
}
}%
{%
}

```

Similarly for mcoltreehypergroup.

```

\ifdef{\@glstyle@mcoltreehypergroup}
{%
\renewglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}%
\glstreegroupheaderskip
}%

```

Group heading.

```

\renewcommand*{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent
\glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glstr@grptitle}}%

```

```

\glstreegroupheaderskip\@afterheading
}%

```

Sub-groups are only supported with \printunsrtglossary.

```

\renewcommand*\glssubgroupheading[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glstreesubgroupitem{##1}{##2}{##3}{##4}%
{\glsnahypertarget{##4}{\glstr@grptitle}}%
}%
}
}%
{%
}

```

Similarly for mcoltreesspannav.

```

\ifdef{\@glstyle@mcoltreesspannav}
{%
\renewglossarystyle{mcoltreesspannav}{%
\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]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glstr@grptitle}}%
\glstreegroupheaderskip\@afterheading
}%

```

Sub-groups are only supported with \printunsrtglossary.

```

\renewcommand*\glssubgroupheading[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%
\glstreesubgroupitem{##1}{##2}{##3}{##4}%
{\glsnahypertarget{##4}{\glstr@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]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glstr@grptitle}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}

```

Similarly for mcoltreenonamehypergroup.

```

\ifdef{\@glsstyle@mcoltreenonamehypergroup}
{%
  \renewglossarystyle{mcoltreenonamehypergroup}{%
    \setglossarystyle{mcoltreenoname}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }%
    \renewcommand*{\glsgroupheading}[1]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@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]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
    }
  }

```

```

\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glstr@grptitle}}%
\glstreegroupheaderskip\@afterheading}%
}
}%
{%
}

```

`mcolalmtree` needs adjusting so that it uses `\glstralmtreeInit`. This doesn't use `\mbox{\par}` which would unbalance the top of the columns.

```

\ifdef{\@glstyle@mcolalmtree}
{%
\renewglossarystyle{mcolalmtree}{%
\setglossarystyle{almtree}%
\renewenvironment{theglossary}%
{%
\glstralmtreeInit
\def\@gl@prevlevel{-1}%
\begin{multicols}{\glsmcols}%
}%
{\par\end{multicols}}}%
}
}%
{%
}

```

Redefine `mcolalmtreegroup` to discourage page breaks after the group headings.

```

\ifdef{\@glstyle@mcolalmtreegroup}
{%
\renewglossarystyle{mcolalmtreegroup}{%
\setglossarystyle{mcolalmtree}%
\renewcommand{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par
\def\@gl@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glstr@grptitle}%
\glstreegroupheaderskip
}%
}

```

Sub-groups are only supported with `\printunsrtglossary`.

```

\renewcommand*\glssubgroupheading[4]{%
\glstrgetgrouptitle{##4}{\glstr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%

```

This is similar to `\subglossentry`

```

\ifnum\@gl@prevlevel=##2\relax
\else
\ifcsundef{\glswidestname\romannumeral##2}%

```

```

    {%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
    }%
    {%
      \settowidth{\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
        \glstrComputerTreeIndent{##2}%
      \else
        \ifcsundef@glswidestname\romannumeral##2}%
        {%
          \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
        }%
        {%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \csname @glswidestname\romannumeral##2\endcsname\space}}%
        }%
      \fi
      \addtolength\parindent{-\glstreeindent}%
      \setlength\glstreeindent\parindent
    \fi
    \fi
    \glstrAltTreeSetSubHangIndent{##2}%
    \glsalttreesubgroupheader{##1}{##2}{##3}{##4}{\glstr@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}%
      \hangindentOpt\relax
      \parindentOpt\relax
      \glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }
  }
}

```

```

}%
\renewcommand*{\glsgroupheading}[1]{%
  \glstrgetgrouptitle{##1}{\glstr@grptitle}%
  \glstreePreHeader{##1}{\glstr@grptitle}%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glshypertarget{##1}{\glstr@grptitle}}%
  \glstreegroupheaderskip
}%

```

Sub-groups are only supported with \printunsrtglossary.

```

\renewcommand*{\glssubgroupheading}[4]{%
  \glstrgetgrouptitle{##4}{\glstr@grptitle}%
  \glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glstr@grptitle}%

```

This is similar to \subglossentry

```

\ifnum\@gls@prevlevel=##2\relax
\else
  \ifcsundef{\glswidestname\romannumeral##2}%
  {%
    \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}%
  }%
  {%
    \settowidth{\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
    \glstrComputeTreeIndent{##2}%
  \else
    \ifcsundef{\glswidestname\romannumeral##2}%
    {%
      \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
    }%
    {%
      \settowidth{\glstreeindent}{\glstreenamefmt{%
        \csname @glswidestname\romannumeral##2\endcsname\space}}%
    }%
  \fi
  \addtolength\parindent{-\glstreeindent}%
  \setlength\glstreeindent\parindent
\fi
\fi
\glstrAltTreeSetSubHangIndent{##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{almtree}%
\renewenvironment{theglossary}%
{%
\glsxtralmtreeInit
\def\@gls@prevlevel{-1}%
\begin{multicols}{\gls@col}%
[\noindent\glstreenavigationfmt{\glsnavigation}]%
}%
{\par\end{multicols}}%
\renewcommand*\@gls@groupheading[1]{%
\glsxtr@group@title{##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*\@gls@subgroupheading[4]{%
\glsxtr@group@title{##4}{\glsxtr@grptitle}%
\glstreeSubPreHeader{##1}{##2}{##3}{##4}{\glsxtr@grptitle}%
}

```

This is similar to \subglossentry

```

\ifnum\@gls@prevlevel=##2\relax
\else
\ifcsundef{\@gls@widestname\romannumeral##2}%
{%
\settowidth{\gls@tmplen}{\glstreenamfmt{\@gls@widestname\space}}%
}%
{%
\settowidth{\gls@tmplen}{\glstreenamfmt{
\csname \@gls@widestname\romannumeral##2\endcsname\space}}%
}%
\ifnum\@gls@prevlevel<##2\relax
\setlength\glstreeindent\gls@tmplen

```



```

\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\ifnum\@gls@prevlevel=0\relax
\glstrComputerTreeIndent{##2}%
\else
\ifcsundef{\@glswidestname\romannumeral##2}%
{%
\settowidth{\glstreeindent}{\glstreenamfmt{\@glswidestname\space}}%
}%
{%
\settowidth{\glstreeindent}{\glstreenamfmt{
\csname \@glswidestname\romannumeral##2\endcsname\space}}%
}%
\fi
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\glstrAltTreeSetSubHangIndent{##2}%
\glsalttreesubgroupheader{##1}{##2}{##3}{##4}%
{\@glsnavhypertarget{##4}{\glstr@grptitle}}{\@gls@tmplen}%
\def\@gls@prevlevel{##2}%
\par
}%
}
}%
{%
}

Reset the default style
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glstr@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{v1.49}{2022-10-14}

Declare package:
\ProvidesPackage{glossary-bookindex}[2022/10/14 v1.49 (NLCT)]

Load required packages.
\RequirePackage{multicol}
\RequirePackage{glossary-tree}

```

`\glxtrbookindexcols` Number of columns.
`\newcommand{\glxtrbookindexcols}{2}`

`\glxtrbookindexname` Format used for top-level entries. (Argument is the label.)
`\newcommand*{\glxtrbookindexname}[1]{\glossentryname{#1}}`

`\glxtrbookindexsubname` Format used for sub entries.
`\newcommand*{\glxtrbookindexsubname}[1]{\glxtrbookindexname{#1}}`

`\glxtrprelocation` Provide in case glossaries-stylemods isn't loaded.
`\providecommand*{\glxtrprelocation}{\space}`

`\glxtrbookindexprelocation` 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*{\glxtrbookindexprelocation}[1]{%`
`\glxtrifhasfield{location}{#1}%`
`{,\glxtrprelocation}%`
`{\glxtrprelocation}%`
`}`

`\glxtrbookindexsubprelocation` Separator used before location list for sub-entries.
`\newcommand*{\glxtrbookindexsubprelocation}[1]{%`
`\glxtrbookindexprelocation{#1}%`
`}`

`\glxtrbookindexlocation` `\glxtrbookindexlocation{<label>}{<location>}`
Displays the location.
`\newcommand*{\glxtrbookindexlocation}[2]{#2}`

`\glxtrbookindexsublocation` `\glxtrbookindexlocation{<label>}{<location>}`
Displays the location for sub-entries.
`\newcommand*{\glxtrbookindexsublocation}{\glxtrbookindexlocation}`

`\glxtrbookindexparentchildsep` Separator used between top-level parent and child entry.
`\newcommand{\glxtrbookindexparentchildsep}{\nopagebreak}`

`\glxtrbookindexparentsubchildsep` Separator used between sub-level parent and child entry.
`\newcommand{\glxtrbookindexparentsubchildsep}{\glxtrbookindexparentchildsep}`

`\glxtrbookindexbetween` Between two top-level entries identified by the labels in the arguments.
`\newcommand{\glxtrbookindexbetween}[2]{}`

<code>\glxtrbookindexsubbetween</code>	Between two level 1 entries identified by the labels in the arguments. <code>\newcommand{\glxtrbookindexsubbetween}[2]{}</code>
<code>\glxtrbookindexsubsubbetween</code>	Between two level 2 entries identified by the labels in the arguments. <code>\newcommand{\glxtrbookindexsubsubbetween}[2]{}</code>
<code>\glxtrbookindexatendgroup</code>	At the end of a letter group. The argument is the label of the last top-level entry. <code>\newcommand{\glxtrbookindexatendgroup}[1]{}</code>
<code>\glxtrbookindexsubatendgroup</code>	At the end of a letter group. The argument is the label of the last level 1 entry. <code>\newcommand{\glxtrbookindexsubatendgroup}[1]{}</code>
<code>\glxtrbookindexsubsubatendgroup</code>	At the end of a letter group. The argument is the label of the last level 2 entry. <code>\newcommand{\glxtrbookindexsubsubatendgroup}[1]{}</code>
<code>\glxtrbookindexgroupskip</code>	Group separator. <code>\newcommand{\glxtrbookindexgroupskip}{\ifglsnogroupskip\else\indexspace\fi}</code>
<code>\glxtrbookindexpregroupskip</code>	After group header. The argument is the skip that would normally be inserted if there wasn't a group header. <code>\newcommand{\glxtrbookindexpregroupskip}[1]{#1}</code>
<code>\glxtrbookindexpostgroupskip</code>	After group header. <code>\newcommand{\glxtrbookindexpostgroupskip}{\indexspace}</code>
<code>\glxtrbookindexpresubgroupskip</code>	Before sub-group separator. The first argument is the skip that would normally be used at this point if there wasn't a header. <code>\newcommand{\glxtrbookindexpresubgroupskip}[3]{\par\medskip}</code>
<code>\glxtrbookindexpostsubgroupskip</code>	After sub-group separator. <code>\newcommand{\glxtrbookindexpostsubgroupskip}[2]{\par\medskip}</code>
<code>\glxtrbookindexformatheader</code>	Group header. <code>\newcommand*{\glxtrbookindexformatheader}[1]{%\par{\centering\glstreegroupheaderfmt{#1}\par}%}</code> Format sub-group title.

`\glxtrbookindexpresubgroupskip{<default>}{<prev group level>}{<group level>}`

`\glstrbookindexformatsubheader` Sub-group header. This defaults to the same format as the top-level group.

```

\newcommand*{\glstrbookindexformatsubheader}[5]{%
\ifnum#2>1\relax
\glstreesubsubitem\glstreegroupheaderfmt{#5}%
\else
\glstreesubitem\glstreegroupheaderfmt{#5}%
\fi
}

```

`\glstrbookindexbookmark` Book mark group heading if supported.

```

\ifdef\pdfbookmark
{%
\newcommand*{\glstrbookindexbookmark}[2]{%
\ifdefstring{\@@glossarysec}{chapter}%
{\pdfbookmark[1]{#1}{#2}}%
{\pdfbookmark[2]{#1}{#2}}%
}
}
{%
\newcommand*{\glstrbookindexbookmark}[2]{}
}

```

`\glstrbookindexsubbookmark` Book mark sub-group heading if supported.

```

\ifdef\pdfbookmark
{%
\newcommand*{\glstrbookindexsubbookmark}[3]{%
\ifdefstring{\@@glossarysec}{chapter}%
{\expandafter\pdfbookmark\expandafter[\number\numexpr#1+1]{#3}{#2}}%
{\expandafter\pdfbookmark\expandafter[\number\numexpr#1+2]{#3}{#2}}%
}
}
{%
\newcommand*{\glstrbookindexsubbookmark}[3]{}
}

```

`\glstrbookindexbookmarkprefix` Make the bookmark label prefix used for letter groups depend on the glossary label (instead of original hardcoded “index.”).

```

\newcommand*{\glstrbookindexbookmarkprefix}{\currentglossary.}

```

`\glstrbookindexcolspread`

```

\newcommand*{\glstrbookindexcolspread}{}

```

`\glstrbookindexmulticolenv`

```

\newcommand*{\glstrbookindexmulticolenv}{\multicols}

```

`bookindex` Define the style.

```

\newglossarystyle{bookindex}{%
\setglossarystyle{index}%
\renewenvironment{theglossary}%

```

```

{%
\ifnum\glstrbookindexcols>1\relax
\ifdefempty\glstrbookindexcolspread
{%
\edef\glstr@beginbookindex{%
\noexpand\begin{\glstrbookindexmulticolseenv}
{\glstrbookindexcols}%
}%
}%
{%
\edef\glstr@beginbookindex{%
\noexpand\begin{\glstrbookindexmulticolseenv}%
{\glstrbookindexcols}[\glstrbookindexcolspread]%
}%
}%
\else
\def\glstr@beginbookindex{}%
\fi
\glstr@beginbookindex
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\@glstr@bookindex@sep\glstrbookindexparentchildsep
\let\@glstr@bookindex@subsep\glstrbookindexparentsubchildsep
\let\@glstr@bookindex@between\@gobble
\let\@glstr@bookindex@subbetween\@gobble
\let\@glstr@bookindex@subsubbetween\@gobble
\let\@glstr@bookindex@atendgroup\relax
\let\@glstr@bookindex@subatendgroup\relax
\let\@glstr@bookindex@subsubatendgroup\relax
\let\@glstr@bookindexgroupskip\relax
}%
{%

```

Do end group hooks.

```

\@glstr@bookindex@subsubatendgroup
\@glstr@bookindex@subatendgroup
\@glstr@bookindex@atendgroup

```

End multicol environment.

```

\ifnum\glstrbookindexcols>1\relax
\edef\glstr@endbookindex{%
\noexpand\end{\glstrbookindexmulticolseenv}%
}%
\else
\def\glstr@endbookindex{}%
\fi
\glstr@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}%
  \glstarget{##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.

```
\@glxtr@bookindex@subsep
\@glxtr@bookindex@subsubbetween{##2}%
```

Update separators.

```
\let\@glxtr@bookindex@subsep\relax
\edef\@glxtr@bookindex@subsubbetween{%
  \noexpand\glxtrbookindexsubsubbetween{##2}%
}%
\edef\@glxtr@bookindex@atsubsubendgroup{%
  \noexpand\glxtrbookindexatsubsubendgroup{##1}%
}%
```

Start sub-sub-item.

```
\glstreesubsubitem
\fi
```

Format entry.

```
\glstarget{##2}{\glxtrbookindexsubname{##2}}%
\glxtrbookindexsubprelocation{##2}%
\glxtrbookindexsublocation{##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.

```
\@glxtr@bookindex@subsubatendgroup
\@glxtr@bookindex@subatendgroup
\@glxtr@bookindex@atendgroup
\glxtrbookindexpregroupskip\@glxtr@bookindexgroupskip
```

Update separators.

```
\let\@glxtr@bookindexgroupskip\glxtrbookindexgroupskip
\let\@glxtr@bookindex@between\@gobble
\let\@glxtr@bookindex@atendgroup\relax
\let\@glxtr@bookindex@subatendgroup\relax
\let\@glxtr@bookindex@subsubatendgroup\relax
```

Fetch the group title from the label supplied in #1.

```
\glxtrgetgrouptitle{##1}{\glxtrcurrentgrptitle}%
```

Do the PDF bookmark if supported.

```
\glxtrbookindexbookmark{\glxtrcurrentgrptitle}{\glxtrbookindexbookmarkprefix##1}%
```

Format the group title.

```
\glxtrbookindexformatheader{\glxtrcurrentgrptitle}%  
\nopagebreak\glxtrbookindexpostgroupskip\nopagebreak\@afterheading  
}%
```

Sub-groups are only supported with `\printunsrtglossary`.

```
\renewcommand*{\glssubgroupheading}[4]{%
```

Do end group hooks.

```
\@glxtr@bookindex@subsubatendgroup  
\@glxtr@bookindex@subatendgroup  
\@glxtr@bookindex@atendgroup  
\glxtrbookindexpresubgroupskip\@glxtr@bookindexgroupskip{##1}{##2}%
```

Update separators.

```
\let\@glxtr@bookindexgroupskip\glxtrbookindexgroupskip  
\let\@glxtr@bookindex@between\@gobble  
\let\@glxtr@bookindex@atendgroup\relax  
\let\@glxtr@bookindex@subatendgroup\relax  
\let\@glxtr@bookindex@subsubatendgroup\relax
```

Get group title.

```
\glxtrgetgrouptitle{##4}{\glxtrcurrentgrptitle}%
```

Do the PDF bookmark if supported.

```
\glxtrbookindexsubbookmark{##2}{##4}{\glxtrcurrentgrptitle}%
```

Format the group title.

```
\glxtrbookindexformatsubheader{##1}{##2}{##3}{##4}{\glxtrcurrentgrptitle}%  
\nopagebreak\glxtrbookindexpostsubgroupskip{##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 `\glxtrbookindexthepage` instead of `\thepage` in case the page numbering has been set to something that contains formatting commands.

`\glxtrbookindexthepage` The `\@printglossary` sets `\currentglossary` to the current glossary label. This is used as a prefix in case the page number is reset.

```
\newcommand{\glxtrbookindexthepage}{%  
\ifdef\currentglossary{\currentglossary.\arabic{page}}{\arabic{page}}%  
}
```

`\glxtrbookindexmarkentry` Writes entry information to the `.aux` file. The argument is the entry label.

```
\newcommand*{\glxtrbookindexmarkentry}[1]{%  
\protected@write\@auxout  
{\let\glxtrbookindexthepage\relax}%  
{\string\glxtr@setbookindexmark{\glxtrbookindexthepage}{#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{v1.49}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-longextra}[2022/10/14 v1.49 (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`

`\glslongextraSubDescFmt{<level>}{<label>}`

Governs the way the child description is displayed.

```
\newcommand{\glslongextraSubDescFmt}[2]{%
  \glslongextraDescFmt{#2}%
}
```

`\glslongextraSubSymbolFmt`

`\glslongextraSubSymbolFmt{<level>}{<label>}`

Governs the way the child symbol is displayed.

```
\newcommand{\glslongextraSubSymbolFmt}[2]{%
  \glslongextraSymbolFmt{#2}%
}
```

`\glslongextraSubSymbolTargetFmt`

`\glslongextraSubSymbolTargetFmt{<level>}{<label>}`

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}{\glxtrshort[noindex,hyper=false]{#2}}%
  \glxtrpostnamehook{#2}%
}
```

```
\glslongextraSubLongFmt{<label>}
```

`\glslongextraSubLongFmt`

Governs the way the long form is displayed.

```
\newcommand{\glslongextraSubLongFmt}[2]{\glslongextraLongFmt{#2}}
```

```
\glslongextraSubLocationFmt{<level>}{<label>}{<location
list>}
```

`\glslongextraSubLocationFmt`

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}{1}`

`\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{<number of columns>}{<prev
group level>}{<group
level>}{<parent entry>}{<group label>}`

`\glslongextraSubGroupHeading`
`\newcommand*{\glslongextraSubGroupHeading}[5]{}`

`\glslongextraHeaderFormat` Format for the column headers.
`\newcommand{\glslongextraHeaderFmt}[1]{\textbf{#1}}`

`\glslongextraNameDescHeader`
`\newcommand{\glslongextraNameDescHeader}{%`
`\glslongextraNameDescTabularHeader\endhead`
`\glslongextraNameDescTabularFooter\endfoot`
`}`

`\glslongextraNameDescTabularHeader`
`\newcommand{\glslongextraNameDescTabularHeader}{%`
`\toprule`
`\glslongextraHeaderFmt\entryname &`
`\glslongextraHeaderFmt\descriptionname\tabularnewline`
`\midrule`
`}`

`\glslongextraNameDescTabularFooter`
`\newcommand{\glslongextraNameDescTabularFooter}{%`
`\bottomrule`
`}`

Unlike the `alttree` style, there aren't different widths for the hierarchical levels.

`\glslongextraSetWidest` Provide in case the tree styles haven't been loaded.
`\newcommand*{\glslongextraSetWidest}[1]{%`
`\def\@glslongextrawidestname{#1}%`
`}`

`\@glslongextrawidestname` Pick up the widest name from the `alttree` style if it has been set. (Will expand to nothing otherwise.)

```
\newcommand*{\@glslongextrawidestname}{\csuse{\@glswidestname}}
```

`\glslongextraUpdateWidest`

```
\newcommand*{\glslongextraUpdateWidest}[1]{%
  \ifundef\@glslongextrawidestname
  {\def\@glslongextrawidestname{#1}}%
  {%
    \settowidth{\dimen@}{\@glslongextrawidestname}%
    \settowidth{\dimen@ii}{#1}%
    \ifdim\dimen@ii>\dimen@
    \def\@glslongextrawidestname{#1}%
    \fi
  }%
}
```

`\glslongextraUpdateWidestChild{<level>}{<text>}`

`\glslongextraUpdateWidestChild`

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}{%
  \settowidth{\gls@tmplen}{\glslongextraHeaderFmt\entryname}%
  Has the widest name been set.
  \settowidth{\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.

```
\settowidth{\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.

```
\settowidth{\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}%
}
```

glslongextraSymLocSetDescWidth 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}%
}
```

glslongextraShortNoNameSetDescWidth 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.

```
\settowidth{\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
}%
\ifglsgnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

extraNameDescLocationHeader

```

\newcommand{\glslongextraNameDescLocationHeader}{%
  \glslongextraNameDescLocationTabularHeader\endhead
  \glslongextraNameDescLocationTabularFooter\endfoot
}

```

meDescLocationTabularHeader

```

\newcommand{\glslongextraNameDescLocationTabularHeader}{%
  \toprule
  \glslongextraHeaderFmt\entryname &
  \glslongextraHeaderFmt\descriptionname &
  \glslongextraHeaderFmt\pagelistname\tabularnewline
  \midrule
}

```

meDescLocationTabularFooter

```

\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
}%
\ifglsgroupskip
  \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
}%
{%
\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
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

extraLocationDescNameHeader

```

\newcommand{\glslongextraLocationDescNameHeader}{%
\glslongextraLocationDescNameTabularHeader\endhead
\glslongextraLocationDescNameTabularFooter\endfoot
}

```

locationDescNameTabularHeader

```

\newcommand{\glslongextraLocationDescNameTabularHeader}{%
\toprule
\glslongextraHeaderFmt\pagelistname&
\glslongextraHeaderFmt\descriptionname&
\glslongextraHeaderFmt\entryname \tabularnewline
\midrule
}

```

locationDescNameTabularFooter

```

\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
}%
\ifglsgnognroupskip
\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
}

```

extraNameDescSymTabularFooter

```

\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
}%
\ifglsgnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

raNameDescSymLocationHeader

```

\newcommand{\glslongextraNameDescSymLocationHeader}{%
\glslongextraNameDescSymLocationTabularHeader\endhead
\glslongextraNameDescSymLocationTabularFooter\endfoot
}

```

DescSymLocationTabularHeader

```

\newcommand{\glslongextraNameDescSymLocationTabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\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
}

```


glslongextraNameSymDescHeader

```
\newcommand{\glslongextraNameSymDescHeader}{%
\glslongextraNameSymDescTabularHeader\endhead
\glslongextraNameSymDescTabularFooter\endfoot
}
```

glslongextraNameSymDescTabularHeader

```
\newcommand{\glslongextraNameSymDescTabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt\symbolname &
\glslongextraHeaderFmt\descriptionname\tabularnewline
\midrule
}
```

glslongextraNameSymDescTabularFooter

```
\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
\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
}

```

raNameSymDescLocationHeader

```

\newcommand{\glslongextraNameSymDescLocationHeader}{%
\glslongextraNameSymDescLocationTabularHeader\endhead
\glslongextraNameSymDescLocationTabularFooter\endfoot
}

```

ymDescLocationTabularHeader

```

\newcommand{\glslongextraNameSymDescLocationTabularHeader}{%
\toprule
\glslongextraHeaderFmt\entryname &
\glslongextraHeaderFmt\symbolname &
\glslongextraHeaderFmt\descriptionname &
\glslongextraHeaderFmt\pagelistname\tabularnewline
\midrule
}

```

ymDescLocationTabularFooter

```

\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
}%
\ifglsgnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

glslongextraSymDescNameHeader

```

\newcommand{\glslongextraSymDescNameHeader}{%
\glslongextraSymDescNameTabularHeader\endhead
\glslongextraSymDescNameTabularFooter\endfoot
}

```

glslongextraSymDescNameTabularHeader

```

\newcommand{\glslongextraSymDescNameTabularHeader}{%
\toprule
\glslongextraHeaderFmt\symbolname &
\glslongextraHeaderFmt\descriptionname &
\glslongextraHeaderFmt\entryname\tabularnewline
\midrule
}

```

glslongextraSymDescNameTabularFooter

```

\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
  }%
  \ifglsgnognroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}

\newcommand{\glslocationSymDescNameHeader}{%
  \glslongextraLocationSymDescNameTabularHeader\endhead
  \glslongextraLocationSymDescNameTabularFooter\endfoot
}

\newcommand{\glslocationSymDescNameTabularHeader}{%
  \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
}%
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`glslongextraDescSymNameHeader`

```
\newcommand{\glslongextraDescSymNameHeader}{%
  \glslongextraDescSymNameTabularHeader\endhead
  \glslongextraDescSymNameTabularFooter\endfoot
}
```

`glslongextraDescSymNameTabularHeader`

```
\newcommand{\glslongextraDescSymNameTabularHeader}{%
  \toprule
  \glslongextraHeaderFmt\descriptionname &
  \glslongextraHeaderFmt\symbolname &
  \glslongextraHeaderFmt\entryname\tabularnewline
  \midrule
}
```

`glslongextraDescSymNameTabularFooter`

```
\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
}

```


raLocationDescSymNameHeader

```
\newcommand{\glslongextraLocationDescSymNameHeader}{%
\glslongextraLocationDescSymNameTabularHeader\endhead
\glslongextraLocationDescSymNameTabularFooter\endfoot
}
```

ionDescSymNameTabularHeader

```
\newcommand{\glslongextraLocationDescSymNameTabularHeader}{%
\toprule
\glslongextraHeaderFmt\pagelistname &
\glslongextraHeaderFmt\descriptionname &
\glslongextraHeaderFmt\symbolname &
\glslongextraHeaderFmt\entryname\tabularnewline
\midrule
}
```

ionDescSymNameTabularFooter

```
\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
}%
\ifglsgroupskip
    \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} &
\glslongextraSubDescFmt{##1}{##2}\tabularnewline
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

\glslongextraSymDescHeader

```

\newcommand{\glslongextraSymDescHeader}{%
\glslongextraSymDescTabularHeader\endhead
\glslongextraSymDescTabularFooter\endfoot
}

```

ngextraSymDescTabularHeader

```

\newcommand{\glslongextraSymDescTabularHeader}{%
\toprule
\glslongextraHeaderFmt\symbolname &
\glslongextraHeaderFmt\descriptionname\tabularnewline
\midrule
}

```

ngextraSymDescTabularFooter

```
\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
}%
\ifglsgnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

```

\glslongextraDescSymHeader

\newcommand{\glslongextraDescSymHeader}{%
\glslongextraDescSymTabularHeader\endhead
\glslongextraDescSymTabularFooter\endfoot
}

```

```

\glslongextraDescSymTabularHeader

\newcommand{\glslongextraDescSymTabularHeader}{%
\toprule
\glslongextraHeaderFmt\descriptionname &
\glslongextraHeaderFmt\symbolname\tabularnewline
\midrule
}

```

```

\glslongextraDescSymTabularFooter

\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}}%
  {\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
}

```

glslongextraLongShortHeader

```

\newcommand{\glslongextraLongShortHeader}{%
  \glslongextraLongShortTabularHeader\endhead
  \glslongextraLongShortTabularFooter\endfoot
}

```

extraLongShortTabularHeader

```

\newcommand{\glslongextraLongShortTabularHeader}{%
  \toprule

```



```

\glslongextraHeaderFmt\glslongextraLongHeader &
\glslongextraHeaderFmt\glslongextraShortHeader\tabularnewline
\midrule
}

```

extraLongShortTabularFooter

```

\newcommand{\glslongextraLongShortTabularFooter}{%
\bottomrule
}

```

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{v1.49}{2022-10-14}

```

Declare package:

```
\ProvidesPackage{glossary-topic}[2022/10/14 v1.49 (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{\glosentry}[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]{%
% \glstrgetgrouptitle{#1}{\thisgrptitle}%
% \section*{\thisgrptitle}%
%}
%

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

`\glstopicSubGroupHeading`

```
\glstopicSubGroupHeading{<prev group level>}{<group
level>}{<parent entry>}{<group label>}
```

```
\newcommand*{\glstopicSubGroupHeading}[4]{%
\begingroup
\glspare\glstopicPreSkip\glspare\noindent
\glstrgetgrouptitle{#4}{\glstopicSubGrouptitle}%
\glstopicAssignSubIndent{#2}%
\glstopicSubItemBox{#2}{\glstopicTitleFont{\glstopicSubGrouptitle}}%
\glstopicSubItemSep
\glspare\nobreak\glstopicPostSkip

```

```

\par
\endgroup
}

```

`\glstopicItem`

`\glstopicItem{<label>}{<location list>}`

```

\newcommand*{\glstopicItem}[2]{%
  \glsparenc{\glstopicPreSkip}\glsparenc{\noindent}
  \glstopicMarker{#1}%
  \glstopicTitleFont
  {%
    \glsentryitem{#1}\glstarget{#1}{\glstopicTitle{#1}}%
  }%
  \ifglshasdesc{#1}%
  {\glsparenc{\nobreak}\glstopicMidSkip\glsparenc{\nobreak}
    \@afterheading\glstopicDesc{#1}\glsparenc{\glstopicPostSkip}
  }%
  {\glsparenc{\nobreak}\glstopicPostSkip}%
  \glstopicLoc{#1}{#2}%
}

```

`\glstopicMarker` May be used to insert a bookmark etc if required.

```

\newcommand*{\glstopicMarker}[1]{}

```

`\glstopicName`

```

\newcommand*{\glstopicName}[1]{\Glossentryname{#1}%
  \ifglshassymbol{#1}{\space\glossentrysymbol{#1}}{}}%
}

```

`\glstopicTitleFont`

```

\newcommand*{\glstopicTitleFont}[1]{\textbf{\large #1}}

```

`\glstopicDesc`

```

\newcommand*{\glstopicDesc}[1]{\Glossentrydesc{#1}\glstopicPostDescription}

```

`\glstopicLoc`

```

\newcommand*{\glstopicLoc}[2]{}

```

`\glstopicParIndent`

```

\newlength\glstopicParIndent
\setlength\glstopicParIndent{20pt}

```

`\glstopicSubIndent`

```

\newlength\glstopicSubIndent
\setlength\glstopicSubIndent{20pt}

```

`\glstopicInit`

```

\newcommand{\glstopicInit}{}

```

`\glstopicAssignSubIndent{<level>}`

`\glstopicAssignSubIndent`

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{<level>}`

`\glstopicAssignWidest`

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}%
    {%
      \settowidth{\glstopicwidest}{%
        \glstopicSubNameFont{\csuse{@glswidestname\romannumeral#1}}%
        \glstopicSubItemSep
      }%
    }%
    {\setlength{\glstopicwidest}{0pt}}%
  }%
  \csedef{@glswidestlength\romannumeral#1}{\the\glstopicwidest}%
  {\setlength{\glstopicwidest}{\csuse{@glswidestlength\romannumeral#1}}}%
}
```

Save the value so that it doesn't have to keep being recalculated.

```

\glstopicPreSkip
\newcommand*{\glstopicPreSkip}{\medskip}

\glstopicMidSkip
\newcommand*{\glstopicMidSkip}{\smallskip}

\glstopicPostSkip
\newcommand*{\glstopicPostSkip}{\smallskip}

\glstopicSubItem
\newcommand*{\glstopicSubItem}[3]{%
  \glstopicSubItemBox{#1}{\glstopicSubNameFont{\glentryitem{#2}%
    \glstarget{#2}{\glossentryname{#2}}}%
    \glstopicSubItemSep
  }%
  \ifglshassymbol{#2}{(\glossentrysymbol{#2})\space}{}%
  \ifglshasdesc{#2}%
    {\glossentrydesc{#2}\glspostdescription\glstopicSubPreLocSep}{}%
    \glstopicSubLoc{#2}{#3}%
  }

\glstopicSubItemSep
\newcommand*{\glstopicSubItemSep}{\quad}

\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}%
}%
}%
{%
\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}[2022/10/14 v1.49 (NLCT)]
```

Load required packages.

```
\RequirePackage{glossary-longbooktabs}
```

`\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
\define@key{printglosstable}{blocksep}{\renewcommand{\glstable@blockalignsep}{#1}}
```

`\glstable@blockalignsep` Alignment spec between blocks.

```
\newcommand{\glstable@blockalignsep}{{}}
```

`glstablesubentries` (*env.*)

```
\newenvironment{glstablesubentries}{\begin{tabular}[t]{l}}{\end{tabular}}
```

`\glstablePreChildren`

```
\newcommand{\glstablePreChildren}{\glstableifpar{\par}}
```

`\glstableblocksubentrysep`

```
\newcommand{\glstableblocksubentrysep}{\\}
```

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{\glstabledeswidth}}

\glstablesymbolcolalign The alignment for the symbol column.
    \newcommand{\glstablesymbolcolalign}{\glstablecenteralign{\glstablesymbolwidth}}

\glstableNameTarget
    \newcommand{\glstableNameTarget}[1]{%
        \glstarget{#1}{\glstableNameFmt{#1}}%
    }

\glstableNameFmt Entry item needs to be included in measuring to ensure there's enough room
for it as well.
    \newcommand{\glstableNameFmt}[1]{\glentryitem{#1}\glossentryname{#1}}

\glstableSubNameTarget
    \newcommand{\glstableSubNameTarget}[1]{%
        \glstarget{#1}{\glstableSubNameFmt{#1}}%
    }

\glstableSubNameFmt
    \newcommand{\glstableSubNameFmt}[1]{%
        \glssubentryitem{#1}%
    }

\glstableotherfield
    \newcommand{\glstableotherfield}{}

    Add an extra key to allow this value to be changed.
    \define@key{printglosstable}{other}{\renewcommand{\glstableotherfield}{#1}}

\glstableOther
    \newcommand{\glstableOther}[1]{%
        \glxtrusefield{#1}{\glstableotherfield}}

\glstableNameSingleFmt
    \newcommand{\glstableNameSingleFmt}[1]{%
        \glstableNameTarget{#1}%
        \ifglshasdesc{#1}%
        {%

```

```

\glstableNameSinglePostName
\glstableNameSingleSuppl
{%
  \ifglshassymbol{#1}%
    {\glstableSymbolFmt{#1}\glstableNameSingleSymSep}%
    {}%
  \ifdefempty\glstableotherfield{%
    {%
      \GlsXtrIfFieldUndef{\glstableotherfield}{#1}{}%
      {\glstableOther{#1}\glstableOtherSep}%
    }%
    \glstableDescFmt{#1}%
  }%
}%
{%
  \ifglshassymbol{#1}%
    {%
      \glstableNameSinglePostName
      \glstableNameSingleSuppl{\glstableSymbolFmt{#1}}%
      \ifdefempty\glstableotherfield{%
        {%
          \GlsXtrIfFieldUndef{\glstableotherfield}{#1}{}%
          {\glstableNameSingleSymSep\glstableOther{#1}}%
        }%
      }%
    }%
    {%
      \ifdefempty\glstableotherfield{%
        {%
          \glstableNameSinglePostName
          \GlsXtrIfFieldUndef{\glstableotherfield}{#1}{}%
          {\glstableOther{#1}}%
        }%
      }%
    }%
  }%
}

```

\glstableNameSingleSuppl

```
\newcommand{\glstableNameSingleSuppl}[1]{(#1)}
```

\glstableNameSinglePostName

```
\newcommand{\glstableNameSinglePostName}{ }
```

\glstableNameSingleSymSep

```
\newcommand{\glstableNameSingleSymSep}{ }
```

\glstableOtherSep

```
\newcommand{\glstableOtherSep}{, }
```

\glstableSubNameSingleFmt

```

\newcommand{\glstableSubNameSingleFmt}[1]{%
\glstableSubNameTarget{#1}%
\GlsXtrIfFieldUndef{desc}{#1}{}%
{%
\GlsXtrIfFieldUndef{symbol}{#1}{}%
{%
\glstableNameSinglePostSubName
\glstableNameSingleSubSuppl{\glstableSubSymbolFmt{#1}}%
}%
}%
{%
\glstableNameSinglePostSubName
\glstableNameSingleSubSuppl
{%
\GlsXtrIfFieldUndef{symbol}{#1}{}%
{\glstableSubSymbolFmt{#1}\glstableNameSingleSymSep}%
\glstableSubDescFmt{#1}%
}%
}%
}

\glstableNameSingleSubSuppl

\newcommand{\glstableNameSingleSubSuppl}[1]{#1}

\glstableNameSinglePostSubName

\newcommand{\glstableNameSinglePostSubName}{ }

\glstableNameNoDesc

\newcommand{\glstableNameNoDesc}[1]{%
\glstableNameTarget{#1}%
\ifdefempty\glstableotherfield{%
{%
\GlsXtrIfFieldUndef{\glstableotherfield}{#1}{}%
{\glstableOtherSep\glstableOther{#1}}%
}%
}%
}

\glstableSubNameNoDesc

\newcommand{\glstableSubNameNoDesc}[1]{%
\glstableSubNameTarget{#1}%
\ifdefempty\glstableotherfield{%
{%
\GlsXtrIfFieldUndef{\glstableotherfield}{#1}{}%
{\glstableOtherSep\glstableOther{#1}}%
}%
}%
}

\glstableSymbolFmt

\newcommand{\glstableSymbolFmt}[1]{\glossentrysymbol{#1}}

```

```

\glstableSubSymbolFmt
    \newcommand{\glstableSubSymbolFmt}[1]{\glstableSymbolFmt{#1}}

\glstableSymbolNameTarget Where the symbol takes place of the name.
    \newcommand{\glstableSymbolNameTarget}[1]{%
        \glstarget{#1}{\glstableSymbolNameFmt{#1}}%
    }

\glstableSymbolNameFmt
    \newcommand{\glstableSymbolNameFmt}[1]{\glssentryitem{#1}\glossentrysymbol{#1}}

\glstableSubSymbolNameTarget Where the symbol takes place of the name.
    \newcommand{\glstableSubSymbolNameTarget}[1]{%
        \glstarget{#1}{\glstableSubSymbolNameFmt{#1}}%
    }

\glstableSubSymbolNameFmt
    \newcommand{\glstableSubSymbolNameFmt}[1]{\glssubentryitem{#1}}

\glstableDesc
    \newcommand{\glstableDesc}[1]{%
        \ifdefempty\glstableotherfield{%
            {%
                \GlsXtrIfFieldUndef{\glstableotherfield}{#1}{}%
                {\glstableOther{#1}\ifglshasdesc{#1}{\glstableOtherSep}{}}%
            }%
            \glstableDescFmt{#1}%
        }

\glstableDescFmt
    \newcommand{\glstableDescFmt}[1]{\glossentrydesc{#1}\glspostdescription}

\glstableSubDesc
    \newcommand{\glstableSubDesc}[1]{\glstableDesc{#1}}

\glstableSubDescFmt
    \newcommand{\glstableSubDescFmt}[1]{\glstableDescFmt{#1}}

\glstableOtherNoDesc
    \newcommand{\glstableOtherNoDesc}[1]{%
        \ifdefempty\glstableotherfield{%
            {%
                \GlsXtrIfFieldUndef{\glstableotherfield}{#1}{}%
                {\glstableOther{#1}}%
            }%
        }
    }

```

```

\glstableSubOtherNoDesc
    \newcommand{\glstableSubOtherNoDesc}[1]{%
        \glstableOtherNoDesc{#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

\glstableblockheader The column header, which may cover multiple columns. Redefined by block
style.
    \newcommand{\glstableblockheader}{}

\glstableblockalign The column alignment. The default assumes two columns per entry (name and
description) 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 the final 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’}{}%
        }
    }

```

`\glstablenuwstyle`

```
\newcommand{\glstablenuwstyle}[2]{%
  \ifcsdef{@glstable@style@#1}%
  {\PackageError{glossary-table}{style ‘#1’ already defined}{}}%
  {\csdef{@glstable@style@#1}{#2}}%
}
```

Provide some common layouts.

```
\glstablenuwstyle{name-desc}{%
```

2 columns per block (name, description).

```
\glstablecolspersblock=2\relax
```

Initialise length registers (need to calculate max name width if par align).

```
\renewcommand{\glstableinitlengthupdates}{%
  \ifKV@printglosstable@header
  \settowidth{\glstablenuwwidth}{%
    {\glstableHeaderFmt\glstablenuwheader}%
  }%
  \else
  \setlength{\glstablenuwwidth}{0pt}%
  \fi
  \setlength{\glstabledescwidth}{0pt}%
}%
```

Update width in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenuwwidth}{\glstableNameFmt{##1}}%
}%
```

Finally set the description width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
    - \glstablenuwwidth}%
  \ifdim\glstabledescwidth<0pt\relax
  \setlength{\glstablenuwwidth}{\dimexpr0.5\glstableblockwidth}%
  \setlength{\glstabledescwidth}{\glstablenuwwidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableNameTarget{##1} & \glstableDesc{##1}}%
```

How to format the entry’s children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameTarget{##1} & \glstableSubDesc{##1}}%
```

How to format the block’s header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstablenuwheader &
  \glstableHeaderFmt\glstabledescheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstablenamecolalign\glstabledesccolalign}%
}
```

Set the default style.

```
\glstablesetstyle{name-desc}
\glstableneverstyle{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}}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
\glstableSubNameSingleFmt{##1}}%
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
\glstableHeaderFmt\glstablenameheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstablenamecolalign}%
}

\glstableneverstyle{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
\settowidth{\glstablesymbolwidth}%
{\glstableHeaderFmt\glstablesymbolheader}%
\else
\setlength{\glstablesymbolwidth}{0pt}%
```

```

\fi
\setlength{\glstablenamewidth}{0pt}%
}%

```

Update widths in unsrt hook.

```

\renewcommand{\glstablelengthupdate}[1]{%
\glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolFmt{##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]{%
\glstableNameNoDesc{##1} & \glstableSymbolFmt{##1}}%

```

How to format the entry's children.

```

\renewcommand{\glstableblocksubentry}[1]{%
\glstableSubNameNoDesc{##1} & \glstableSubSymbolFmt{##1}}%

```

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}%
}

\glstableneverstyle{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
\settowidth{\glstablenamewidth}{%
\glstableHeaderFmt\glstablenameheader}%
\else
\setlength{\glstablenamewidth}{0pt}%
\fi
\setlength{\glstabledescwidth}{0pt}%
}%

```


Update width in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableNameFmt{##1}}%
}%
```

Finally set the description width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledeswidth}{\dimexpr\glstableblockwidth
    - \glstablenamewidth}%
  \ifdim\glstabledeswidth<0pt\relax
    \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstabledeswidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableDesc{##1} &
  \glstableNameTarget{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubDesc{##1} &
  \glstableSubNameTarget{##1}%
}%
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstabledesheader &
  \glstableHeaderFmt\glstablenameheader
}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstabledesccolalign\glstablenamecolalign}%
}

\glstableneverstyle{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
    \settowidth{\glstablesymbolwidth}{%
      {\glstableHeaderFmt\glstablesymbolheader}%
    }
  \else
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
  \setlength{\glstablenamewidth}{0pt}%
}%
```

Update widths in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolFmt{##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]{%
  \glstableSymbolFmt{##1} &
  \glstableNameNoDesc{##1}%
}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubSymbolFmt{##1} &
  \glstableSubNameNoDesc{##1}%
}%
```

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}%
}

\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
    \settowidth{\glstablenamewidth}{%
      {\glstableHeaderFmt\glstablenameheader}%
    }
    \settowidth{\glstablesymbolwidth}{%
      {\glstableHeaderFmt\glstablesymbolheader}%
    }
  \else
    \setlength{\glstablenamewidth}{0pt}%
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
}
```

```

\fi
\setlength{\glstabledescwidth}{0pt}%
}%

```

Update widths in unsrt hook.

```

\renewcommand{\glstablelengthupdate}[1]{%
\glstablemeasureandupdate{\glstabilenamewidth}{\glstableNameFmt{##1}}%
\glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolFmt{##1}}%
}%

```

Finally set the description width to the remaining available.

```

\renewcommand{\glstablefinishlengthupdates}{%
\setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
- \glstablesymbolwidth - \glstabilenamewidth}%
\ifdim\glstabledescwidth<0pt\relax
\setlength{\glstabilenamewidth}{\dimexpr0.5\glstableblockwidth
- 0.5\glstablesymbolwidth}%
\setlength{\glstabledescwidth}{\glstabilenamewidth}%
\fi
}%

```

How to format the top-level entry in the block.

```

\renewcommand{\glstableblockentry}[1]{%
\glstableNameTarget{##1} &
\glstableSymbolFmt{##1} &
\glstableDesc{##1}}%

```

How to format the entry's children.

```

\renewcommand{\glstableblocksubentry}[1]{%
\glstableSubNameTarget{##1} &
\glstableSubSymbolFmt{##1} &
\glstableSubDesc{##1}}%

```

How to format the block's header row, if required.

```

\renewcommand{\glstableblockheader}{%
\glstableHeaderFmt\glstabilenameheader &
\glstableHeaderFmt\glstablesymbolheader &
\glstableHeaderFmt\glstabledescheader}%

```

Set the block's column alignments.

```

\renewcommand{\glstableblockalign}{%
\glstabilenamecolalign\glstablesymbolcolalign\glstabledesccolalign}%
}

\glstabilenewstyle{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

```

```

\settowidth{\glstablenamewidth}%
  {\glstableHeaderFmt\glstablenameheader}%
\settowidth{\glstablesymbolwidth}%
  {\glstableHeaderFmt\glstablesymbolheader}%
\else
  \setlength{\glstablenamewidth}{0pt}%
  \setlength{\glstablesymbolwidth}{0pt}%
\fi
\setlength{\glstabledescwidth}{0pt}%
}%

```

Update widths in unsrt hook.

```

\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableNameFmt{##1}}%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolFmt{##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} &
  \glstableDesc{##1} &
  \glstableSymbolFmt{##1}%
}%

```

How to format the entry's children.

```

\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameTarget{##1} &
  \glstableSubDesc{##1} &
  \glstableSubSymbolFmt{##1}%
}%

```

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}%
}

```

As name-desc but the other field is put in the description column.

```
\glstablenustyle{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
    \settowidth{\glstablenamewidth}{%
      {\glstableHeaderFmt\glstablenameheader}%
    }%
  \else
    \setlength{\glstablenamewidth}{0pt}%
  \fi
  \setlength{\glstabledescwidth}{0pt}%
}%
```

Update width in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableNameFmt{##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} & \glstableOtherNoDesc{##1}}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubNameTarget{##1} & \glstableSubOtherNoDesc{##1}}%
```

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\glstabledesccolalign}%
}
```

```
\glstablenustyle{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
    \settowidth{\glstablenamewidth}{%
      {\glstableHeaderFmt\glstablenameheader}%
    }%
  \else
    \setlength{\glstablenamewidth}{0pt}%
  \fi
  \setlength{\glstabledesccolwidth}{0pt}%
}%
```

Update width in unsrt hook.

```
\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablenamewidth}{\glstableNameFmt{##1}}%
}%
```

Finally set the description width to the remaining available.

```
\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledesccolwidth}{\dimexpr\glstableblockwidth
    - \glstablenamewidth}%
  \ifdim\glstabledesccolwidth<0pt\relax
    \setlength{\glstablenamewidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstabledesccolwidth}{\glstablenamewidth}%
  \fi
}%
```

How to format the top-level entry in the block.

```
\renewcommand{\glstableblockentry}[1]{%
  \glstableOtherNoDesc{##1} &
  \glstableNameTarget{##1}}%
```

How to format the entry's children.

```
\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubOtherNoDesc{##1} &
  \glstableSubNameTarget{##1}}%
```

How to format the block's header row, if required.

```
\renewcommand{\glstableblockheader}{%
  \glstableHeaderFmt\glstableotherheader &
  \glstableHeaderFmt\glstablenameheader}%
```

Set the block's column alignments.

```
\renewcommand{\glstableblockalign}{\glstabledesccolalign\glstablenamecolalign}%
}
```

As name-other but the use the symbol in place of the name.

```
\glstablenustyle{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
\settowidth{\glstablesymbolwidth}{%
  {\glstableHeaderFmt\glstablesymbolheader}%
\else
\setlength{\glstablesymbolwidth}{0pt}%
\fi
\setlength{\glstabledescwidth}{0pt}%
}%

```

Update width in unsrt hook.

```

\renewcommand{\glstablelengthupdate}[1]{%
  \glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolNameFmt{##1}}%
}%

```

Finally set the description width to the remaining available.

```

\renewcommand{\glstablefinishlengthupdates}{%
  \setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
    - \glstablesymbolwidth}%
  \ifdim\glstabledescwidth<0pt\relax
    \setlength{\glstablesymbolwidth}{\dimexpr0.5\glstableblockwidth}%
    \setlength{\glstabledescwidth}{\glstablesymbolwidth}%
  \fi
}%

```

How to format the top-level entry in the block.

```

\renewcommand{\glstableblockentry}[1]{%
  \glstableSymbolNameTarget{##1} & \glstableOtherNoDesc{##1}}%

```

How to format the entry's children.

```

\renewcommand{\glstableblocksubentry}[1]{%
  \glstableSubSymbolNameTarget{##1} & \glstableSubOtherNoDesc{##1}}%

```

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\glstabledesccolalign}%
}

```

```

\glstableneverstyle{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
    \settowidth{\glstablesymbolwidth}{%
      {\glstableHeaderFmt\glstablesymbolheader}%
    }%
  \else
    \setlength{\glstablesymbolwidth}{0pt}%
  \fi
}

```

```

\fi
\setlength{\glstabledescwidth}{0pt}%
}%

```

Update width in unsrt hook.

```

\renewcommand{\glstablelengthupdate}[1]{%
\glstablemeasureandupdate{\glstablesymbolwidth}{\glstableSymbolNameFmt{##1}}%
}%

```

Finally set the description width to the remaining available.

```

\renewcommand{\glstablefinishlengthupdates}{%
\setlength{\glstabledescwidth}{\dimexpr\glstableblockwidth
- \glstablesymbolwidth}%
\ifdim\glstabledescwidth<0pt\relax
\setlength{\glstablesymbolwidth}{\dimexpr0.5\glstableblockwidth}%
\setlength{\glstabledescwidth}{\glstablesymbolwidth}%
\fi
}%

```

How to format the top-level entry in the block.

```

\renewcommand{\glstableblockentry}[1]{%
\glstableOtherNoDesc{##1} & \glstableSymbolNameTarget{##1}}%

```

How to format the entry's children.

```

\renewcommand{\glstableblocksubentry}[1]{%
\glstableSubOtherNoDesc{##1} &
\glstableSubSymbolNameTarget{##1}}%

```

How to format the block's header row, if required.

```

\renewcommand{\glstableblockheader}{%
\glstableHeaderFmt\glstableotherheader &
\glstableHeaderFmt\glstablesymbolheader}%

```

Set the block's column alignments.

```

\renewcommand{\glstableblockalign}{\glstabledesccolalign\glstablesymbolcolalign}%
}

```

```

\glstablecaption{\glstablecaption{<toc title>}{<title>}{<label>}}

```

```

\newcommand{\glstablecaption}[3]{%
\caption[#1]{#3#2}%
}

```

`\glstablepostnextcaption`

```

\newcommand{\glstablepostnextcaption}{ Cont./}

```

```

\glstabilenextcaption{\glstabilenextcaption{<toc title>}{<title>}}

```

```

\newcommand{\glstabilenextcaption}[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}}
```

`\glstablerowspan`

```
\glstablerowspan{\<text>}
```

```
\newcommand{\glstablerowspan}[1]{%
\multicolumn{\glstabletotalcols}{c}{\parbox{\LTcapwidth}{#1}}%
}
```

`\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}%
{%
\glstableifpar{\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

\glstableifmeasuring{\langle true \rangle}{\langle false \rangle}
\newcommand{\glstableifmeasuring}[2]{#2}

\glstable@stepentry
\newcommand{\glstable@stepentry}[1]{%
    \ifglstentrycounter
    \stepcounter{glossaryentry}%
    \fi
}

\glstable@stepsubentry
\newcommand{\glstable@stepsubentry}[1]{%
    \ifglssubentrycounter
    \stepcounter{glossarysubentry}%
    \fi
}

\glstablemeasureandupdate{\langle len reg \rangle}{\langle text \rangle}
\newcommand{\glstablemeasureandupdate}[2]{%
    \let\glstableifmeasuring\@firstoftwo
    Disable unset and indexing
    \let\@glunset\@glxtrbuffer@unset
    \let\@gls@org@noindex\ifKV@glslink@noindex
    \KV@glslink@noindextrue

```

Disable targets and links

```
\let\@glstable@org@target\glstarget
\let\glstarget\@secondoftwo
\let\@glstable@org@link\@glslink
\let\@glslink\@secondoftwo
```

Disable ref step and label

```
\let\glstable@org@stepentry\glsstepentry
\let\glsstepentry\glstable@stepentry
\let\glstable@org@stepsubentry\glsstepsubentry
\let\glsstepsubentry\glstable@stepsubentry
```

Measure.

```
\settowidth{\dimen@}{#2}%
```

Update if wider.

```
\ifdim\dimen@>#1\relax
\setlength{#1}{\dimen@}%
\fi
```

Restore

```
\let\ifKV@glslink@noindex\@gls@org@noindex
\let\@glsunset\@glxtr@unset
\let\glstarget\@glstable@org@target
\let\@glslink\@glstable@org@link
\let\glsstepentry\glstable@org@stepentry
\let\glsstepsubentry\glstable@org@stepsubentry
\let\glstable@ifmeasuring\@secondoftwo
}
```

`\glstable@ifhaspreamble`

```
\newcommand{\glstable@ifhaspreamble}[2]{%
\ifdefempty\glossarypreamble
{#2}%
{%
\ifx\@glstable@defaultpreamble\glossarypreamble
\ifcvoid{\@glossarypreamble@\currentglossary}{#2}{#1}%
\else
#1%
\fi
}%
}
```

Need the type, preamble and postamble.

```
\define@key{printglosstable}{type}{\renewcommand{\@gls@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@key{printglosstable}{init}{\renewcommand{\glstable@init}{#1}}
```

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}[true]%
{%
\ifnum\@glstable@groups@n=0\relax
\let\glstable@groupheading\@gobble
\glstr@printgloss@groupsfalse
}else
\let\glstable@groupheading\glstablegroupheading
\glstr@printgloss@groupstrue
\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]{%
\glstablefinishrow
\multicolumn{\glstabletotalcols}{c}{%
\glstrgetgrouptitle{#1}{\glstrcurrentgrptitle}%
\glstableGroupHeaderFmt\glstrcurrentgrptitle
}%
\\*%
}
```

```
\@glstable@defaultpreamble
```

```
\let\@glstable@defaultpreamble\glossarypreamble
```

```
\glstableGroupHeaderFmt
```

```
\newcommand{\glstableGroupHeaderFmt}{\glstableHeaderFmt}
```

```
\printunsrttable
```

```
\newrobustcmd*{\printunsrttable}[1][ ]{%
\bgroup
```

Initialise glossary type.

```
\def\@glo@type{\glstabledefaulttype}%
```

Initialise title.

```
\def\glossarytitle{%
\ifcsdef{\@glo@type@\@glo@type @title}%
{\csuse{\@glo@type@\@glo@type @title}}%
{\glossaryname}%
}%
\def\glossarytoctitle{\glossarytitle}%
```

Initialise preamble.

```
\let\glossarypreamble\@glstable@defaultpreamble
```

Set table keys.

```
\setkeys*{printglosstable}{#1}%
\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.

```
\glstableifpar
{%
\glstableblockwidth=\dimexpr\linewidth/\glstableblockperrowcount
-\glstablecolsperblock\tabcolsep\relax
\glstableinitlengthupdates
}%
```

If no vertical line required between blocks, `\glstablefinishrow` can simply do a new line.

```
\ifdefempty\glstable@blockalignsep{\def\glstable@finishrow{}}{}}%
```

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{\tabularnewline}%
\ifKV@printglosstable@rules
\appto\glstable@header{\midrule}%
\fi
\fi
\protected@edef\glstable@begin{%
\noexpand\begin{longtable}{\expandonce\glstable@alignment}%
\noexpand\glstablecaption
{\noexpand\glossarytoctitle}{\noexpand\glossarytitle}%
{\noexpand\@glossaryseclabel}%
\noexpand\tabularnewline
}%
Add preamble if set.
\glstable@ifhaspreamble
{%
\eappto\glstable@begin{\noexpand\glstablefirsthead
{\expandonce\glossarypreamble}%
\noexpand\tabularnewline
\expandonce\glstable@header
\noexpand\endfirsthead
\noexpand\glstabl-nextcaption
{\noexpand\glossarytoctitle}{\noexpand\glossarytitle}%
{\expandonce\glossarypreamble}%
}%
}%
{%
\eappto\glstable@begin{%
\expandonce\glstable@header
\noexpand\endfirsthead
\noexpand\glstabl-nextcaption
{\noexpand\glossarytoctitle}{\noexpand\glossarytitle}%
}%
}%
\eappto\glstable@begin{%
\noexpand\tabularnewline
\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{\expandonce\glossarypostamble}%
\noexpand\tabularnewline
\noexpand\endfoot
\noexpand\bottomrule
\noexpand\glstablelastfoot{\expandonce\glossarypostamble}%
\noexpand\tabularnewline
\noexpand\endlastfoot
}%
\else
\eappto\glstable@begin{%
\noexpand\glstablefoot{\expandonce\glossarypostamble}%
\noexpand\tabularnewline
\noexpand\endfoot
\noexpand\glstablelastfoot{\expandonce\glossarypostamble}%
\noexpand\tabularnewline
\noexpand\endlastfoot
}%
\fi
}%

```

Initialisation hook.

```
\glstable@init
```

Set up filtering.

```
\let\printunsrtglossaryentryprocesshook\glstable@filter
```

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][{}]{}
```

The glossary will be empty on the first L^AT_EX run as the entries won't be defined until bib2gls has selected them.

```

\glstrifemptyglossary{\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\@glxtr@tmp{\noexpand\setkeys{printgloss}{\expandonce\glstable@opts}}%
\@glxtr@tmp
\glstableifpar{\glstablefinishlengthupdates}%
\global\glstablecurrentblockindex=0\relax

```

```

        \glstable@begin
        \glstablefinishrow
        \end{longtable}%
    }%
    {%
        \expandafter\printunsrtglossary\expandafter[\glstable@opts]\relax
    }%
\egroup
}

\glstable@child
\newcommand{\glstable@child}[1]{%
    \ifdefempty\glstable@dochildren{%
        {\appto\glstable@dochildren{\glstableblocksubentrysep}}%
        \appto\glstable@dochildren{\glstableblocksubentry{#1}}%
    }
}

\glstableChildEntries
\newcommand{\glstableChildEntries}[1]{%
    \def\glstable@dochildren{%
        \GlsXtrIfHasNonZeroChildCount*{#1}%
        {%
            \glstrfieldforlistloop{#1}{childlist}{\glstable@child}%
            \preto\glstable@dochildren{%
                \glstablePreChildren
                \begin{glstabsubentries}%
            }%
            \appto\glstable@dochildren{\end{glstabsubentries}}%
        }%
    }%
    \glstable@dochildren
}

\glstable@n@amps
\newcommand{\glstable@n@amps}[1]{%
    \ifnum#1>0\relax
        \count@=0\relax
        \loop
            \advance\count@ by 1\relax
            \appto\glstable@endblock{&}%
        \ifnum\count@<#1
            \repeat
    \fi
}

```

TODO: this needs to do nothing if it occurs are the start of a new row. The position of `\tabularnewline` may need changing within `\printunsrttable` and the table style.


```

\glstablefinishrow
    \newcommand{\glstablefinishrow}{%
        \glstable@finishrow
        \tabularnewline
    }

\glstable@finishrow
    \newcommand{\glstable@finishrow}{%
        \def\glstable@endblock{}%
        \@glstable@finishrow
        \glstable@endblock
    }

\@glstable@finishrow
    \newcommand{\@glstable@finishrow}{%
        \ifnum\glstablecurrentblockindex>0
            \ifnum\glstablecurrentblockindex<\glstableblockperrowcount
                \expandafter\glstable@n&\expandafter{%
                    \numexpr
                        (\glstableblockperrowcount-\glstablecurrentblockindex)
                        * (\glstablecolsperblock -1)
                }%
            \fi
            \global\glstablecurrentblockindex=0
        \fi
    }

table
    \newglossarystyle{table}%
    {%
        \renewenvironment{theglossary}%
        {%
            \glstableifpar{\glstablefinishlengthupdates}%
            \global\glstablecurrentblockindex=0\relax
            \glstable@begin
        }
        {%
            \glstablefinishrow
            \end{longtable}%
        }%
        \renewcommand*{\glossaryheader}{}%
        \renewcommand*{\glsgroupheading}[1]{\glstable@groupheading{##1}}%
        \renewcommand*{\glssubgroupheading}[4]{}%
        \renewcommand*{\glsgroupskip}{}%
        \renewcommand*{\glossentry}[2]{%
            \global\advance\glstablecurrentblockindex by 1\relax
            \ifnum\glstableblockperrowcount=\glstablecurrentblockindex
                \gdef\glstable@endblock{%
                    \global\glstablecurrentblockindex=0\relax

```

```

        \tabularnewline
      }%
    \else
      \gdef\glstable@endblock{&}%
    \fi
    \glstableblockentry{##1}%
    \glstableChildEntries{##1}%
    \glstable@endblock
  }%
  \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{\glstr@dooption}[1]{\setupglossaries{#1}}%
  \let\glstr@declareoption\@gls@declareoption
}
{%
  \newcommand{\glstr@dooption}[1]{%
    \PassOptionsToPackage{#1}{glossaries}%
  }%
  \PassOptionsToPackage{toc}{glossaries}
  \PassOptionsToPackage{nopostdot}{glossaries}
  \PassOptionsToPackage{noredefwarn}{glossaries}
  \@ifpackageloaded{polyglossia}%
  {}%
  {%
    \@ifpackageloaded{babel}%
    {\PassOptionsToPackage{translate=babel}{glossaries}}%
    {}%
  }%
  \newcommand*\@glstr@declareoption[2]{%
    \DeclareOptionX{#1}{#2}%
    \DeclareOption{#1}{#2}%
  }
}
\newcommand*\glstrundefaction[2]{%
  \@glstrundeftag\PackageError{glossaries-extra}{#1}{#2}%
}
\newcommand*\glstr@warnonexistsordo[1]{%

```

```

\newcommand*{\glxstrundeftag}{??}
\newcommand*{\@glxstrundeftag}{}
\newcommand*{\@glxstr@warn@undefaction}[2]{%
  \@glxstrundeftag\GlossariesExtraWarning{#1}%
}
\newcommand*{\@glxstr@err@undefaction}[2]{%
  \@glxstrundeftag\PackageError{glossaries-extra}{#1}{#2}%
}
\newcommand*{\@glxstr@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*{\@glxstr@redef@forlgsentries}{}
\newcommand*{\@glxstr@do@redef@forlgsentries}{%
  \renewcommand*{\forlgsentries}[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}%
[\glxstr@undefaction@val\glxstr@undefaction@nr]%
{warn,error}%
{%
  \ifcase\glxstr@undefaction@nr\relax
    \let\glxstrundefaction\@glxstr@warn@undefaction
    \let\glxstr@warnonexistsordo\@glxstr@warn@onexistsordo
    \let\@glxstr@redef@forlgsentries\@glxstr@do@redef@forlgsentries
  \or
    \let\glxstrundefaction\@glxstr@err@undefaction
    \let\glxstr@warnonexistsordo\@gobble
    \let\@glxstr@redef@forlgsentries\relax
  \fi
}
\newcommand*{\@glxstr@record}[3]{}
\newcommand*{\glxstr@recordsee}[2]{}
\newcommand*{\@glxstr@defaultnumberformat}{\glsnumberformat}%
\newcommand*{\GlsXtrSetDefaultNumberFormat}[1]{%
  \renewcommand*{\@glxstr@defaultnumberformat}{#1}%
}

```

```

}%
\newcommand*{\@glxtr@do@record@wrglossary}[1]{%
\beginingroup
\ifKV@glslink@noindex
\else
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\let\glslabel\@gls@label
\glswriteentry{#1}%
{%
\ifdefempty{\@glxtr@thevalue}%
{%
\ifx\@glxtr@org@theHvalue\@glxtr@theHvalue
\else
\let\theHglentrycounter\@glxtr@theHvalue
\fi
\glxtr@saveentrycounter
\let\@do@wrglossary\@glxtr@dorecord
}%
{%
\let\theHglentrycounter\@glxtr@thevalue
\let\theHglentrycounter\@glxtr@theHvalue
\let\@do@wrglossary\@glxtr@dorecordnodefer
}%
\ifx\@glxtr@record@setting\@glxtr@record@setting@alsoindex
\glxtr@do@wrglossary{#1}%
\else
\@glxtr@wrglossmark
\glxtr@inc@wrglossaryctr{#1}%
\@do@wrglossary
\fi
}%
\fi
\endgroup
}
\newcommand*{\glxtr@do@alsoindex@wrglossary}[1]{%
\glxtr@do@wrglossary{#1}%
\@glxtr@do@record@wrglossary{#1}%
}
\newcommand*{\@glxtr@record}[3]{%
\protected@edef\@gls@label{\glsdetoklabel{#2}}%
\let\glslabel\@gls@label
\ifglentryexists{#2}{%
{%
\@glxtr@wrglossmark
\beginingroup
\let\@glsnumberformat\@glxtr@defaultnumberformat
\def\@glxtr@thevalue{%
\def\@glxtr@theHvalue{\@glxtr@thevalue}%
\let\@glxtr@org@theHvalue\@glxtr@theHvalue
\let\@gls@counter\glscounter

```

```

\if@glxtr@equations
  \@glxtr@use@equation@counter
\fi
\@gls@setdefault@glslink@opts
\csuse{@glxtr@#3@prekeys}%
\setkeys{#3}{#1}%
\glxtr@do@autoadd{#3}%
\csuse{@glxtr@#3@postkeys}%
\glxtr@inc@wrglossaryctr{#2}%
\ifKV@glslink@noindex
\else
  \glswriteentry{#2}%
  {%
    \ifdefempty{\@glxtr@thevalue}%
    {%
      \ifx\@glxtr@org@theHvalue\@glxtr@theHvalue
      \else
        \let\theHglentrycounter\@glxtr@theHvalue
      \fi
      \glxtr@saveentrycounter
      \let\@do@wrglossary\@glxtr@dorecord
    }%
    {%
      \let\theHglentrycounter\@glxtr@thevalue
      \let\theHglentrycounter\@glxtr@theHvalue
      \let\@do@wrglossary\@glxtr@dorecordnodefer
    }%
    \ifx\@glxtr@record@setting\@glxtr@record@setting@alsoindex
      \glxtr@do@wrglossary{#2}%
    \else
      \@do@wrglossary
    \fi
  }%
\fi
\endgroup
}%
}
\newcommand{\@glxtr@glslink@prekeys}{\glslinkpresetkeys}
\newcommand{\@glxtr@glslink@postkeys}{\glslinkpostsetkeys}
\newcommand{\@glxtr@glossadd@prekeys}{\glsaddpresetkeys}
\newcommand{\@glxtr@glossadd@postkeys}{\glsaddpostsetkeys}
\newcommand*{\@glxtr@dorecord{%
  \global\let\@glsrecordlocref\theHglentrycounter
  \let\@glxtr@orgprefix\@glo@counterprefix
  \ifx\theHglentrycounter\theHglentrycounter
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@glxtr@theentrycounter{\theHglentrycounter}%
    \protected@edef\@glxtr@theHentrycounter{\theHglentrycounter}%
    \@onelevel@sanitize\@glxtr@theentrycounter

```

```

\@onelevel@sanitize\@glxtr@theHentrycounter
\protected@edef\@do@gl@getcounterprefix{\noexpand\@gl@getcounterprefix
  {\@glxtr@theentrycounter}{\@glxtr@theHentrycounter}%
}%
\@do@gl@getcounterprefix
\fi
\ifx\@glxtr@record@setting\@glxtr@record@setting@nameref
\@glxtr@do@nameref@record
  {\@gl@label}{\@glo@counterprefix}{\@gl@counter}{\@gl@numberformat}%
  {\@gl@recordlocref}%
\else
\protected@write\@auxout{}{\string\glxtr@record
  {\@gl@label}{\@glo@counterprefix}{\@gl@counter}{\@gl@numberformat}%
  {\@gl@recordlocref}}%
\fi
\@glxtr@counterrecordhook
\let\@glo@counterprefix\@glxtr@orgprefix
}
\newcommand*\@glxtr@dorecordnodefer{%
\ifx\theglsentrycounter\theHglentrycounter
\ifx\@glxtr@record@setting\@glxtr@record@setting@nameref
\@glxtr@do@nameref@record
  {\@gl@label}{\@gl@counter}{\@gl@numberformat}%
  {\theglsentrycounter}%
\else
\protected@write\@auxout{}{\string\glxtr@record
  {\@gl@label}{\@gl@counter}{\@gl@numberformat}%
  {\theglsentrycounter}}%
\fi
\else
\edef\@do@gl@getcounterprefix{\noexpand\@gl@getcounterprefix
  {\theglsentrycounter}{\theHglentrycounter}%
}%
\@do@gl@getcounterprefix
\ifx\@glxtr@record@setting\@glxtr@record@setting@nameref
\@glxtr@do@nameref@record
  {\@gl@label}{\@glo@counterprefix}{\@gl@counter}%
  {\@gl@numberformat}{\theglsentrycounter}%
\else
\protected@write\@auxout{}{\string\glxtr@record
  {\@gl@label}{\@glo@counterprefix}{\@gl@counter}{\@gl@numberformat}%
  {\theglsentrycounter}}%
\fi
\fi
\@glxtr@counterrecordhook
}
\newcommand{\@glxtr@ifnum@mmode}[2]{%
\ifmmode
\ifst@rred
#2%

```

```

\else
\if@display #1\else #2\fi
\fi
\else
#2%
\fi
}
\newcommand*{\@glxtr@do@nameref@record}[5]{%
\gls@ifnotmeasuring
{%
\protected@write\@auxout{}\string\glxtr@record@nameref
{#1}{#2}{#3}{#4}{#5}%
{\csuse{@currentlabelname}}{\csuse{@currentHref}}%
{\theHglentrycounter}}%
}%
}
\newcommand*{\@glxtr@recordcounter}{%
\glxtr@noop@recordcounter
}
\newcommand*{\@glxtr@noop@recordcounter}[1]{%
\PackageError{glossaries-extra}{\string\GlsXtrRecordCounter\space
requires record=only or record=hybrid package option}{}}%
}
\newcommand*{\@glxtr@op@recordcounter}[1]{%
\protected@eappto\@glxtr@counterrecordhook{\noexpand\@glxtr@docounterrecord{#1}}%
}
\newcommand*{\@glxtr@recordsee}[2]{%
\@glxtrwrglossmark
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\protected@write\@auxout{}\string\glxtr@recordsee{#1}{\@gls@xref}}%
}
\newcommand{\printunsrtglossaryunit}{%
\print@noop@unsrtglossaryunit
}
\newcommand*{\glxtr@setup@record}{\let\@do@wrglossary\glxtr@do@wrglossary}
\newcommand*{\glxtr@indexonly@saveentrycounter}{%
\ifKV@glslink@noindex
\else
\glxtr@saveentrycounter
\fi
}
\newcommand*{\glxtr@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@noprntgloss{%
    \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=\@glxtr@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*{\@glxtr@record@only@setup}{%
\def\glxtr@setup@record{%
  \@glxtr@autoseeindexfalse
  \let\@do@seeglossary\@glxtr@recordsee
  \let\@glxtr@record\@glxtr@record
  \let\@do@wrglossary\@glxtr@do@record@wrglossary
  \let\@glxtr@saveentrycounter\relax
  \let\glxtrundefaction\@glxtr@warn@undefaction
  \let\glxtr@warnonexistsordo\@glxtr@warn@onexistsordo
  \glxtr@addloclistfield
  \renewcommand*{\@glxtr@autoindexcrossrefs}{}%
  \let\@glxtr@recordcounter\@glxtr@op@recordcounter
  \def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
  \def\glxtrsetaliasnoindex{}%
  \ifdef\@glxtr@setupsort@none{\@glxtr@setupsort@none}{}%
  \def\glxtrNoGlossaryWarning{\@glxtr@record@noglossarywarning}%
  \RequirePackage{glossaries-extra-bib2gls}[v1.48]%
}%
}
\define@choicekey{glossaries-extra.sty}{record}
[{\@glxtr@record@setting\glxtr@record@nr}%
{off,only,alsoindex,nameref,hybrid}%
{only}%
{%
  \ifcase\glxtr@record@nr\relax
  \def\glxtr@setup@record{%
    \renewcommand*{\@do@seeglossary}{\@glxtr@doseeglossary}%
    \renewcommand*{\@glxtr@record}[3]{}%
    \let\@do@wrglossary\glxtr@do@do@wrglossary
    \let\@glxtr@saveentrycounter\glxtr@indexonly@saveentrycounter
    \let\glxtrundefaction\@glxtr@err@undefaction
    \let\glxtr@warnonexistsordo\@gobble
    \let\@glxtr@recordcounter\@glxtr@noop@recordcounter
    \def\printunsrtglossaryunit{\print@noop@unsrtglossaryunit}%
    \undef\glxtrsetaliasnoindex
  }%
  \or
  \@glxtr@record@only@setup
  \or
  \def\glxtr@setup@record{%
    \renewcommand*{\@glxtr@record@setting@alsoindex}{alsoindex}%
    \renewcommand*{\@do@seeglossary}{\@glxtr@dosee@alsoindex@glossary}%

```

```

\let\@glxstr@record\@glxstr@record
\let\@do@wrglossary\glxstr@do@alsoindex@wrglossary
\let\@glx@saveentrycounter\glxstr@indexonly@saveentrycounter
\let\glxstrundefaction\@glxstr@warn@undefaction
\let\glxstr@warnonexistssordo\@glxstr@warn@onexistssordo
\glxstr@addloclistfield
\let\@glxstr@recordcounter\@glxstr@op@recordcounter
\def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
\undef\glxstrsetaliasnoindex
}%
\or
\@glxstr@record@only@setup
\ifundef\hyperlink
{\GlossariesExtraWarning{You have requested record=nameref but
the document doesn't support hyperlinks}}%
}%
\or
\def\glxstr@setup@record{%
\renewcommand*{\@glxstr@record@setting@alsoindex}{hybrid}%
\renewcommand*{\@do@seeglossary}{\@glxstr@dosee@alsoindex@glossary}%
\let\@glxstr@record\@glxstr@record
\let\@do@wrglossary\glxstr@do@alsoindex@wrglossary
\let\@glx@saveentrycounter\glxstr@indexonly@saveentrycounter
\let\glxstrundefaction\@glxstr@warn@undefaction
\let\glxstr@warnonexistssordo\@glxstr@warn@onexistssordo
\glxstr@addloclistfield
\let\@glxstr@recordcounter\@glxstr@op@recordcounter
\def\printunsrtglossaryunit{\print@op@unsrtglossaryunit}%
\undef\glxstrsetaliasnoindex
}%
\fi
}
\newcommand*{\@glxstr@docdefval}{0}
\newcommand*{\if@glxstrdocdef}{\ifnum\@glxstr@docdefval>0 }
\newcommand*{\@glxstrdocdeftrue}{\def\@glxstr@docdefval{1}}
\newcommand*{\@glxstrdocdeffalse}{\def\@glxstr@docdefval{0}}
\define@choicekey{glossaries-extra.sty}{docdef}
[\@glxstr@docdefsetting\@glxstr@docdefval]%
{false,true,restricted,atom}[true]%
{%
\ifnum\@glxstr@docdefval>1\relax
\renewcommand*{\@glxdoifexistssorwarn}{\glxdoifexists}%
\else
\renewcommand*{\@glxdoifexistssorwarn}{\glxdoifexistssorwarn}%
\fi
}
\newcommand*{\if@glxstrdocdefrestricted}{\ifnum\@glxstr@docdefval>1 }
\newcommand*{\@glxdoifexistssorwarn}{\glxdoifexistssorwarn}
\define@boolkey{glossaries-extra.sty}[\@glxstr]{indexcrossrefs}[true]{%
\if@glxstrindexcrossrefs

```

```

\else
  \renewcommand*{\@glsxtr@autoindexcrossrefs}{}%
\fi
}
\@glsxtr@indexcrossrefsfalse
\newcommand*{\@glsxtr@autoindexcrossrefs}{\@glsxtr@indexcrossrefstrue}
\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@doooption{nowarn}%
}
\newcommand*{\@glsxtr@defpostpunc}{%
\@glsxtr@declareoption{postdot}{%
  \glsxtr@doooption{nopostdot=false}%
  \renewcommand*{\@glsxtr@defpostpunc}{%
    \renewcommand*{\glspostdescription}{%
      \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi}%
    }%
}
}

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

```

```

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

\newcommand*{\@glstr@doabbreviationsdef}{%
  \ifpackage{babel}%
  {\providecommand{\abbreviationsname}{\acronymname}}%
  {\providecommand{\abbreviationsname}{Abbreviations}}%
  \newglossary[glg-abr]{abbreviations}{gls-abr}{glo-abr}{\abbreviationsname}%
  \renewcommand*{\glstrabbrvtype}{abbreviations}%
  \newcommand*{\printabbreviations}[1][1]{%
    \printglossary[type=\glstrabbrvtype,##1]%
  }%
  \disable@keys{glossaries-extra.sty}{abbreviations}%
  \ifglsacronym
  \else
  \renewcommand*{\acronymtype}{\glstrabbrvtype}%
  \fi
}%
\@glstr@declareoption{abbreviations}{%
  \let\@glstr@abbreviationsdef\@glstr@doabbreviationsdef
}

```

```

\newcommand*{\GlsXtrDefineAbbreviationShortcuts}{%
  \newcommand*{\ab}{\cgl{s}}%
  \newcommand*{\abp}{\cgl{s}pl}%
  \newcommand*{\as}{\gl{s}xtrshort}%
  \newcommand*{\asp}{\gl{s}xtrshortpl}%
  \newcommand*{\al}{\gl{s}xtrlong}%
  \newcommand*{\alp}{\gl{s}xtrlongpl}%
  \newcommand*{\af}{\gl{s}xtrfull}%
  \newcommand*{\afp}{\gl{s}xtrfullpl}%
  \newcommand*{\Ab}{\cGl{s}}%
  \newcommand*{\Abp}{\cGl{s}pl}%
  \newcommand*{\As}{\Gl{s}xtrshort}%
  \newcommand*{\Asp}{\Gl{s}xtrshortpl}%
  \newcommand*{\Al}{\Gl{s}xtrlong}%
  \newcommand*{\Alp}{\Gl{s}xtrlongpl}%
  \newcommand*{\Af}{\Gl{s}xtrfull}%
  \newcommand*{\Afp}{\Gl{s}xtrfullpl}%
  \newcommand*{\AB}{\cGL{S}}%
  \newcommand*{\ABP}{\cGL{S}pl}%
  \newcommand*{\AS}{\GL{S}xtrshort}%
  \newcommand*{\ASP}{\GL{S}xtrshortpl}%
  \newcommand*{\AL}{\GL{S}xtrlong}%
  \newcommand*{\ALP}{\GL{S}xtrlongpl}%
  \newcommand*{\AF}{\GL{S}xtrfull}%
  \newcommand*{\AFP}{\GL{S}xtrfullpl}%
  \providecommand*{\newabbr}{\newabbreviation}%
  \let\GlsXtrDefineAbbreviationShortcuts\relax
}

\newcommand*{\GlsXtrDefineAcShortcuts}{%
  \newcommand*{\ac}{\cgl{s}}%
  \newcommand*{\acp}{\cgl{s}pl}%
  \newcommand*{\acs}{\gl{s}xtrshort}%
  \newcommand*{\acsp}{\gl{s}xtrshortpl}%
  \newcommand*{\acl}{\gl{s}xtrlong}%
  \newcommand*{\aclp}{\gl{s}xtrlongpl}%
  \newcommand*{\acf}{\gl{s}xtrfull}%
  \newcommand*{\acfp}{\gl{s}xtrfullpl}%
  \newcommand*{\Ac}{\cGl{s}}%
  \newcommand*{\Acp}{\cGl{s}pl}%
  \newcommand*{\Acs}{\Gl{s}xtrshort}%
  \newcommand*{\Acsp}{\Gl{s}xtrshortpl}%
  \newcommand*{\Acl}{\Gl{s}xtrlong}%
  \newcommand*{\Aclp}{\Gl{s}xtrlongpl}%
  \newcommand*{\Acf}{\Gl{s}xtrfull}%
  \newcommand*{\Acfp}{\Gl{s}xtrfullpl}%
  \newcommand*{\AC}{\cGL{S}}%
  \newcommand*{\ACP}{\cGL{S}pl}%
  \newcommand*{\ACS}{\GL{S}xtrshort}%
  \newcommand*{\ACSP}{\GL{S}xtrshortpl}%
  \newcommand*{\ACL}{\GL{S}xtrlong}%

```

```

\newcommand*{\ACLP}{\GLSxtrlongpl}%
\newcommand*{\ACF}{\GLSxtrfull}%
\newcommand*{\ACFP}{\GLSxtrfullpl}%
\providecommand*{\newabbr}{\newabbreviation}%
\let\GlsXtrDefineAcShortcuts\relax
}
\newcommand*{\GlsXtrDefineOtherShortcuts}{%
\newcommand*{\newentry}{\newglossaryentry}%
\ifdef\printsymbols
{%
\newcommand*{\newsym}{\glxtrnewsymbol}%
}%
\ifdef\printnumbers
{%
\newcommand*{\newnum}{\glxtrnewnumber}%
}%
\let\GlsXtrDefineOtherShortcuts\relax
}
\newcommand*{\@glxtr@setupshortcuts}{%
\newcommand*{\@glxtr@shortcutsval}{\ifglacrshortcuts acro\else none\fi}%
\define@choicekey{glossaries-extra.sty}{shortcuts}{%
[\@glxtr@shortcutsval\@glxtr@shortcutsnr]}%
{acronyms,acro,abbreviations,abbr,other,all,true,ac,none,false}[true]{%
\ifcase\@glxtr@shortcutsnr\relax % acronyms
\renewcommand*{\@glxtr@setupshortcuts}{%
\glacrshortcutstrue
\DefineAcronymSynonyms
}%
\or % acro
\renewcommand*{\@glxtr@setupshortcuts}{%
\glacrshortcutstrue
\DefineAcronymSynonyms
}%
\or % abbreviations
\renewcommand*{\@glxtr@setupshortcuts}{%
\GlsXtrDefineAbbreviationShortcuts
}%
\or % abbr
\renewcommand*{\@glxtr@setupshortcuts}{%
\GlsXtrDefineAbbreviationShortcuts
}%
\or % other
\renewcommand*{\@glxtr@setupshortcuts}{%
\GlsXtrDefineOtherShortcuts
}%
\or % all
\renewcommand*{\@glxtr@setupshortcuts}{%
\glacrshortcutstrue
\GlsXtrDefineAcShortcuts
\GlsXtrDefineAbbreviationShortcuts
}

```

```

        \GlsXtrDefineOtherShortcuts
    }%
\or % true
    \renewcommand*{\@glxtr@setupshortcuts}{%
        \glsacrshortcutstrue
        \GlsXtrDefineAcShortcuts
        \GlsXtrDefineAbbreviationShortcuts
        \GlsXtrDefineOtherShortcuts
    }%
\or % ac
    \renewcommand*{\@glxtr@setupshortcuts}{%
        \glsacrshortcutstrue
        \GlsXtrDefineAcShortcuts
    }%
\else % none, false
    \renewcommand*{\@glxtr@setupshortcuts}{}%
\fi
}
\newcommand*{\@glxtr@doaccsupp}{}
\@glxtr@declareoption{accsupp}{%
    \renewcommand*{\@glxtr@doaccsupp}{\RequirePackage{glossaries-accsupp}}}
\newcommand*{\@glxtr@doloadprefix}{}
\@glxtr@declareoption{prefix}{%
    \renewcommand*{\@glxtr@doloadprefix}{\RequirePackage{glossaries-prefix}}}
\newcommand{\glxtrNoGlossaryWarning}[1]{%
    \GlossariesExtraWarning{Glossary ‘#1’ is missing}%
    \@glxtr@defaultnoglossarywarning{#1}%
}
\define@choicekey{glossaries-extra.sty}{nomissingglstext}
[{\@glxtr@nomissingglstextval\@glxtr@nomissingglstextnr}%
{true,false}[true]{%
    \ifcase\@glxtr@nomissingglstextnr\relax % true
        \renewcommand{\glxtrNoGlossaryWarning}[1]{\null}%
    \else % false
        \renewcommand{\glxtrNoGlossaryWarning}[1]{%
            \@glxtr@defaultnoglossarywarning{#1}%
        }%
    \fi
}
\newcommand*{\@glxtr@redefstyles}{}
\define@key{glossaries-extra.sty}{stylemods}[default]{%
    \ifstrequal{#1}{default}%
    {%
        \renewcommand*{\@glxtr@redefstyles}{%
            \RequirePackage{glossaries-extra-stylemods}}%
    }%
    {%
        \ifstrequal{#1}{all}%
        {%
            \renewcommand*{\@glxtr@redefstyles}{%

```

```

\PassOptionsToPackage{all}{glossaries-extra-stylemods}%
\RequirePackage{glossaries-extra-stylemods}%
}%
}%
{%
\renewcommand*{\@glxtr@redefstyles}{}%
\@for\@glxtr@tmp:=#1\do{%
\IfFileExists{glossary-\@glxtr@tmp.sty}%
{%
\eappto\@glxtr@redefstyles{%
\noexpand\RequirePackage{glossary-\@glxtr@tmp}}%
}%
{%
\PackageError{glossaries-extra}%
{Glossaries style package ‘glossary-\@glxtr@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\@glxtr@redefstyles{\RequirePackage{glossaries-extra-stylemods}[=v1.48]}%
}
}%
}
\newcommand*{\@glxtr@do@style}{%
\define@key{glossaries-extra.sty}{style}{%
\renewcommand*{\@glxtr@do@style}{%
\setkeys{glossaries.sty}{style={#1}}%
\setglossarystyle{#1}%
}%
}%
}
\newcommand*{\@glxtr@inc@wrglossaryctr}[1]{%
\newcommand*{\GlsXtrInternalLocationHyperlink}[3]{%
\glxtrhyperlink{#1#2#3}{#3}%
}
\newcommand*{\@glxtr@wrglossary@locationhyperlink}[3]{%
\pageref{wrglossary.#3}%
}
}
\@glxtr@declareoption{indexcounter}{%
\glxtr@dooption{counter=wrglossary}%
\ifundef\c@wrglossary
{%
\newcounter{wrglossary}%
\renewcommand{\thewrglossary}{\arabic{wrglossary}}%
}%
}%
\newcommand*{\@glxtr@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@doption{debug=true}%
\renewcommand*{\@glsxtrwrglossmark}{}%
\or % false
\glsxtr@doption{debug=false}%
\renewcommand*{\@glsxtrwrglossmark}{}%
\let\@glsxtr@doshowtarget\@secondoftwo
\or % showtargets
\glsxtr@doption{debug=showtargets}%
\def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
\or % showwrgloss
\glsxtr@doption{debug=true}%
\renewcommand*{\@glsxtrwrglossmark}{\glsxtrwrglossmark}%
\or % all
\glsxtr@doption{debug=showtargets,debug=showaccsupp}%
\renewcommand*{\@glsxtrwrglossmark}{\glsxtrwrglossmark}%
\def\@glsxtr@doshowtarget{\@glsxtrshowtargetleft}%
\or % showaccsupp
\glsxtr@doption{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]%
{left,right,innerleft,innerright,annoteleft,annoteright}%

```

```

{%
\glxtr@doooption{debug=showtargets}%
\ifcase\@glxtr@doshowtargetsnr\relax
\def\@glxtr@doshowtarget{\@glxtr@showtargetleft}%
\def\glxtr@showtargetouter{\glsshowtargetouter}%
\def\glxtr@showtargetinner{\glsshowtargetinner}%
\let\@glxtr@showtargetmark\empty
\or
\def\@glxtr@doshowtarget{\@glxtr@showtargetright}%
\def\glxtr@showtargetouter{\glsshowtargetouter}%
\def\glxtr@showtargetinner{\glsshowtargetinner}%
\let\@glxtr@showtargetmark\empty
\or
\def\@glxtr@doshowtarget{\@glxtr@showtargetleft}%
\def\glxtr@showtargetouter{\glxtr@showtargetinner}%
\def\glxtr@showtargetinner{\glsshowtargetinnersymleft}%
\let\@glxtr@showtargetmark\empty
\or
\def\@glxtr@doshowtarget{\@glxtr@showtargetright}%
\def\glxtr@showtargetouter{\glxtr@showtargetinner}%
\def\glxtr@showtargetinner{\glsshowtargetinnersymright}%
\let\@glxtr@showtargetmark\empty
\or
\def\@glxtr@doshowtarget{\@glxtr@showtargetleft}%
\def\glxtr@showtargetouter{\glxtr@showtargetinner}%
\def\glxtr@showtargetinner{\glsshowtargetinnersymleft}%
\def\@glxtr@showtargetmark{\@glsshowtargetmarkfmt\glxtr@showtargetsymbolright}%
\or
\def\@glxtr@doshowtarget{\@glxtr@showtargetright}%
\def\glxtr@showtargetouter{\glxtr@showtargetinner}%
\def\glxtr@showtargetinner{\glsshowtargetinnersymright}%
\def\@glxtr@showtargetmark{\@glsshowtargetmarkfmt\glxtr@showtargetsymbolleft}%
\fi
}
\DeclareOptionX*{%
\expandafter\glxtr@doooption\expandafter{\CurrentOption}}
\ProcessOptionsX
\RequirePackage{glossaries}
\@glxtr@doaccsupp
\@glxtr@doloadprefix
\@glxtr@defpostpunc
\def\glsdoshowtarget{\@glxtr@doshowtarget}
\newcommand{\glxtr@showtargetsymbolright}{\tiny$\triangleleft$}%
\newcommand{\glxtr@showtargetsymbolleft}{\tiny$\triangleright$}%
\providecommand*{\glsshowtargetinner}[1]{\glsshowtargetfont [1]}
\providecommand*{\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*{\glsshowtargetinnersymleft}[1]{%
\glsshowtargetinner{#1}\allowbreak\glxtr@showtargetsymbolleft}
\newcommand*{\glsshowtargetinnersymright}[1]{%
\glxtr@showtargetsymbolright\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]{%
  \glsdofexists{#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@glxtr@autoseeindex\@glxtr@org@gloautosee\fi}%
}%
{}
\renewcommand*{\gls@checkseeallowed}{%
  \if@glxtr@autoseeindex\@gls@see@noindex\fi
}
\@glxtr@abbreviationsdef
\let\@glxtr@abbreviationsdef\relax
\@glxtr@setupshortcuts
\@glxtr@redef@for@gl@sentries
\renewcommand{\glxtr@doption}[1]{\setupglossaries{#1}}%
\disable@keys{glossaries-extra.sty}{accsupp}
\newcommand*{\glossariesextrasetup}[1]{%
  \let\glxtr@setup@record\relax
  \let\@glxtr@setupshortcuts\relax
  \let\@glxtr@redef@for@gl@sentries\relax
  \let\@glxtr@doloadprefix\relax
  \setkeys{glossaries-extra.sty}{#1}%
  \@glxtr@abbreviationsdef
  \let\@glxtr@abbreviationsdef\relax
  \@glxtr@setupshortcuts
  \glxtr@setup@record
  \@glxtr@redef@for@gl@sentries
  \@glxtr@doloadprefix
}
\let\glxtr@org@@do@wrglossary\@do@wrglossary
\newcommand*{\glxtr@@do@wrglossary}[1]{%
  \@glxtrwrglossmark
  \glxtr@inc@wrglossaryctr{#1}%
  \glxtr@org@@do@wrglossary{#1}%
}
\let\glxtr@saveentrycounter\@gls@saveentrycounter
\let\@gls@saveentrycounter\glxtr@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]{%
\ifglstryentryexists{#1}%
{\ifbool{glo@\glsdetoklabel{#1}@flag}{#3}{#2}}%
{#2}%
}
}
\ifdef\s@ifglossaryexists
{}
{
\renewcommand{\ifglossaryexists}{%
\@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
\ifcsundef{glo@type@#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][\@glsxtrfmt]{#1}{#2}{#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@glstdetoklabel}{#1}@#2}{#3}%
}
\newcommand*\glsxtrfieldlistgadd[3]{%

```

```

\listcsadd{glo@glstetoklabel{#1}@#2}{#3}%
}
\newcommand*{\glstrfieldlistadd}[3]{%
\listcsadd{glo@glstetoklabel{#1}@#2}{#3}%
}
\newcommand*{\glstrfieldlistxadd}[3]{%
\listcsxadd{glo@glstetoklabel{#1}@#2}{#3}%
}
\newcommand*{\glstrfielddolistloop}[2]{%
\dolistcsloop{glo@glstetoklabel{#1}@#2}%
}
\newcommand*{\glstrfieldforlistloop}[3]{%
\forlistcsloop{#3}{glo@glstetoklabel{#1}@#2}%
}
\newrobustcmd*{\glstrfieldformatlist}[2]{%
\beginingroup
\def\@dtl@formatlist@itemsep{%
\def\@dtl@formatlist@lastitem{%
\def\@dtl@formatlist@prelastitem{%
\def\@dtl@formatlist@prelastitemsep{%
\forlistcsloop{\@dtl@formatlist@handler}{glo@glstetoklabel{#1}@#2}%
\@dtl@formatlist@prelastitem\@dtl@formatlist@lastitem
\endgroup
}
\newcommand*{\glstrfieldifinlist}[5]{%
\ifinlistcs{#3}{glo@glstetoklabel{#1}@#2}{#4}{#5}%
}
\newcommand*{\glstrfieldxifinlist}[5]{%
\xifinlistcs{#3}{glo@glstetoklabel{#1}@#2}{#4}{#5}%
}
\newcommand*{\glstrforcsvfield}{%
\@ifstar\s@glstrforcsvfield\@glstrforcsvfield
}
\newcommand*{\@glstrforcsvfield}[3]{%
\@glstrifhasfield{#2}{#1}%
{%
\let\glstrendfor\@endfortrue
\@for\@glstr@label:=\glscurrentfieldvalue\do
{\expandafter#3\expandafter{\@glstr@label}}}%
}%
}
\newcommand*{\s@glstrforcsvfield}[3]{%
\s@glstrifhasfield{#2}{#1}%
{%
\let\glstrendfor\@endfortrue
\@for\@glstr@label:=\glscurrentfieldvalue\do
{\expandafter#3\expandafter{\@glstr@label}}}%
}%
}
\newrobustcmd*{\glstrfieldformatcsvlist}[2]{%

```



```

\@glstrifhasfield{#2}{#1}%
{\@dtlformatlist\glscurrentfieldvalue}%
}%
}
\newcommand*{\GlsXtrIfFieldValueInCsvList}{%
  \ifstar\s@GlsXtrIfFieldValueInCsvList\@GlsXtrIfFieldValueInCsvList
}
\newcommand*{\@GlsXtrIfFieldValueInCsvList}[5]{%
  \@glstrifhasfield{#2}{#1}%
  {%
    \expandafter\DTLifinlist\expandafter{\glscurrentfieldvalue}%
    {#3}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\s@GlsXtrIfFieldValueInCsvList}[5]{%
  \s@glstrifhasfield{#2}{#1}%
  {%
    \expandafter\DTLifinlist\expandafter{\glscurrentfieldvalue}%
    {#3}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\GlsXtrIfValueInFieldCsvList}{%
  \ifstar\s@GlsXtrIfValueInFieldCsvList\@GlsXtrIfValueInFieldCsvList
}
\newcommand*{\@GlsXtrIfValueInFieldCsvList}[5]{%
  \@glstrifhasfield{#2}{#1}%
  {%
    \DTLifinlist{#3}{\glscurrentfieldvalue}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\s@GlsXtrIfValueInFieldCsvList}[5]{%
  \s@glstrifhasfield{#2}{#1}%
  {%
    \DTLifinlist{#3}{\glscurrentfieldvalue}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\xGlsXtrIfValueInFieldCsvList}{%
  \ifstar\s@xGlsXtrIfValueInFieldCsvList\@xGlsXtrIfValueInFieldCsvList
}
\newcommand*{\@xGlsXtrIfValueInFieldCsvList}[5]{%
  \@glstrifhasfield{#2}{#1}%
  {%
    \protected@edef\@gls@tmp{#3}%
    \expandafter\DTLifinlist\expandafter{\@gls@tmp}{\glscurrentfieldvalue}{#4}{#5}%
  }%
  {#5}%
}

```

```

}
\newcommand*{\s@xGlsXtrIfValueInFieldCsvList}[5]{%
\s@glxstrifhasfield{#2}{#1}%
{%
\protected@edef\@gls@tmp{#3}%
\expandafter\DTLifinlist\expandafter{\@gls@tmp}{\glscurrentfieldvalue}{#4}{#5}%
}%
{#5}%
}
\newrobustcmd{\glxstrifhasfield}{%
\@ifstar{\s@glxstrifhasfield}{\@glxstrifhasfield}%
}
\newcommand{\@glxstrifhasfield}[4]{%
{\s@glxstrifhasfield{#1}{#2}{#3}{#4}}%
}
\newcommand{\s@glxstrifhasfield}[4]{%
\letcs{\glscurrentfieldvalue}{glo@\glsetoklabel{#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@\glsetoklabel{#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*{\glxtrentryparentname}[1]{%
  \ifcsdef{glo@\glstdetoklabel{#1}@parent}%
    {\csuse{glo@\csuse{glo@\glstdetoklabel{#1}@parent}@name}}%
  }%
}
\newcommand*{\glxtrdeffield}[2]{\csdef{glo@\glstdetoklabel{#1}@#2}}
\newcommand*{\glxtrredefield}[2]{\protected@csdef{glo@\glstdetoklabel{#1}@#2}}
\newcommand*{\glxtrapptocsvfield}[3]{%
  \ifcsdef{glo@\glstdetoklabel{#1}@#2}%
    {\csappto{glo@\glstdetoklabel{#1}@#2}{, #3}}%
    {\csdef{glo@\glstdetoklabel{#1}@#2}{#3}}%
}
\newcommand*{\glxtrsetfieldifexists}[3]{\glstdoifexists{#1}{#3}}
\newrobustcmd*{\GlsXtrSetField}[3]{%
  \glxtrsetfieldifexists{#1}{#2}%
  {\csdef{glo@\glstdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\GlsXtrLetField}[3]{%
  \glxtrsetfieldifexists{#1}{#2}%
  {\cslet{glo@\glstdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\csGlsXtrLetField}[3]{%
  \glxtrsetfieldifexists{#1}{#2}%
  {\csletcs{glo@\glstdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\GlsXtrLetFieldToField}[4]{%
  \glxtrsetfieldifexists{#1}{#2}%
  {\csletcs{glo@\glstdetoklabel{#1}@#2}{glo@\glstdetoklabel{#3}@#4}}%
}
\newrobustcmd*{\gGlsXtrSetField}[3]{%
  \glxtrsetfieldifexists{#1}{#2}%
  {\csgdef{glo@\glstdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\xGlsXtrSetField}[3]{%
  \glxtrsetfieldifexists{#1}{#2}%
  {\protected@csxdef{glo@\glstdetoklabel{#1}@#2}{#3}}%
}
\newrobustcmd*{\eGlsXtrSetField}[3]{%
  \glxtrsetfieldifexists{#1}{#2}%
  {\protected@csedef{glo@\glstdetoklabel{#1}@#2}{#3}}%
}
\newcommand*{\GlsXtrIfFieldEqStr}{%
  \@ifstar\sGlsXtrIfFieldEqStr\@GlsXtrIfFieldEqStr
}
\newrobustcmd*{\@GlsXtrIfFieldEqStr}[5]{%
  \@glxtrifhasfield{#1}{#2}%
  {%
    \ifdefstring{\glscurrentfieldvalue}{#3}{#4}{#5}%
  }%
  {#5}%
}

```

```

}
\newrobustcmd*{\s@GlsXtrIfFieldEqStr}[5]{%
  \s@glstrifhasfield{#1}{#2}%
  {%
    \ifdefstring{\glscurrentfieldvalue}{#3}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\GlsXtrIfFieldEqXpStr}{%
  \ifstar\s@GlsXtrIfFieldEqXpStr\@GlsXtrIfFieldEqXpStr
}
\newrobustcmd*{\@GlsXtrIfFieldEqXpStr}[5]{%
  \@glstrifhasfield{#1}{#2}%
  {%
    \protected@edef\@glstmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@glstmp}{#4}{#5}%
  }%
  {#5}%
}
\newrobustcmd*{\s@GlsXtrIfFieldEqXpStr}[5]{%
  \s@glstrifhasfield{#1}{#2}%
  {%
    \protected@edef\@glstmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@glstmp}{#4}{#5}%
  }%
  {#5}%
}
\newcommand*{\GlsXtrIfXpFieldEqXpStr}{%
  \ifstar\s@GlsXtrIfXpFieldEqXpStr\@GlsXtrIfXpFieldEqXpStr
}
\newrobustcmd*{\@GlsXtrIfXpFieldEqXpStr}[5]{%
  \@glstrifhasfield{#1}{#2}%
  {%
    \protected@edef\@glstmp{\glscurrentfieldvalue}%
    \let\glscurrentfieldvalue\@glstmp
    \protected@edef\@glstmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@glstmp}{#4}{#5}%
  }%
  {#5}%
}
\newrobustcmd*{\s@GlsXtrIfXpFieldEqXpStr}[5]{%
  \s@glstrifhasfield{#1}{#2}%
  {%
    \protected@edef\@glstmp{\glscurrentfieldvalue}%
    \let\glscurrentfieldvalue\@glstmp
    \protected@edef\@glstmp{#3}%
    \ifdefequal{\glscurrentfieldvalue}{\@glstmp}{#4}{#5}%
  }%
  {#5}%
}
}

```

```

\ifdef\foreignlanguage
{
  \ifdef\GetTrackedDialectFromLanguageTag
  {
    \newcommand{\GlsXtrForeignText}[2]{%
      \let\@glstr@org@currentfieldvalue\glscurrentfieldvalue
      \glstrifhasfield{\GlsXtrForeignTextField}{#1}%
      {%
        \expandafter\GetTrackedDialectFromLanguageTag\expandafter
        {\glscurrentfieldvalue}{\@glstr@dialect}%
        \let\@glstr@locale\glscurrentfieldvalue
        \let\glscurrentfieldvalue\@glstr@org@currentfieldvalue
        \ifdefempty\@glstr@dialect
        {%
          \ifundef\TrackedDialectClosestSubMatch
          {%
            \GlossariesExtraWarning{Can't obtain dialect label
              (tracklang v1.3.6+ required)}%
          }%
          {\let\@glstr@dialect\TrackedDialectClosestSubMatch}%
        }%
      }%
    }%
    \ifdefempty\@glstr@dialect
    {%
    }%
    {%
      \ifcsundef{captions\@glstr@dialect}{}%
      {%
        \IfTrackedDialectHasMapping{\@glstr@dialect}%
        {%
          \edef\@glstr@dialect{%
            \GetTrackedDialectToMapping{\@glstr@dialect}}%
          \ifcsundef{captions\@glstr@dialect}{}%
          {%
            \ifcsundef{captions\@tracklang@lang}{}%
            {%
              \let\@glstr@dialect\@tracklang@lang
            }%
          }%
        }%
      }%
      {%
        \ifcsundef{captions\@tracklang@lang}{}%
        {%
          \let\@glstr@dialect\@tracklang@lang
        }%
      }%
    }%
    \ifdefempty\@glstr@dialect
    {%

```

```

        \GlsXtrUnknownDialectWarning{\@glstr@locale}{\@tracklang@lang}%
        #2%
    }%
    {\foreignlanguage{\@glstr@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*{\glstrpageref}[1]{%
        \ifglentrycounter
            \pageref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
        \else
            \ifglssubentrycounter
                \pageref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
            \else
                \gls{#1}%
            \fi
        \fi
    }
}%
{%
    \newcommand*{\glstrpageref}[1]{%
        \ifglentrycounter
            \pageref{glentry-\glsdetoklabel{#1}}%
        \else
            \ifglssubentrycounter
                \pageref{glentry-\glsdetoklabel{#1}}%
            \else
                \gls{#1}%
            \fi
        \fi
    }
}

```

```

}%
\newcommand{\apptoglossarypreamble}[2][\glsdefaultttype]{%
  \ifcsdef{glolist@#1}%
  {%
    \ifcsundef{@glossarypreamble@#1}%
    {\csdef{@glossarypreamble@#1}{}}%
    {}%
    \csappto{@glossarypreamble@#1}{#2}%
  }%
  {%
    \GlossariesExtraWarning{Glossary ‘#1’ is not defined}%
  }%
}
\newcommand{\preglossarypreamble}[2][\glsdefaultttype]{%
  \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}%
    }
    \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*{\glxtr@s@newignoredglossary}[1]{%
  \ifcsdef{glolist@#1}
  {%
    \glxtrundefaction{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}%
    {%
      \defglentryfmt[#1]{\glentryfmt}%
    }%
    {}%
  }%
}
}
\glsifusetranslator
{%
  \renewcommand*{\glsettoctitle}[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*{\glsettoctitle}[1]{%
    \ifcsdef{@glotype@#1@title}%
    {\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}%
    {\def\glossarytoctitle{\glossarytitle}}%
  }
}
\newcommand{\provideignoredglossary}{%
  \@ifstar\glxtr@s@provideignoredglossary\glxtr@provideignoredglossary
}
\newcommand*{\glxtr@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}%
  {%
    \defglentryfmt[#1]{\glentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \protected@eappto\@gls@nohyperlist{, #1}%
  }%
}%
}
\newcommand*{\glxtr@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}%
    {%
      \defglentryfmt[#1]{\glentryfmt}%
    }%
    {}%
  }%
}
\newcommand*{\glxtr@copytoglossary}[2]{%
  \glsdoifexists{#1}%
  {%
    \ifcsdef{glolist@#2}
    {%
      \protected@eappto{glolist@#2}{#1,}%
    }%
    {}%
  }%
}

```

```

        \glstrundefaction{Glossary type ‘#2’ doesn’t exist}{}%
    }%
}%
}
\renewcommand{\glsdoifexists}[2]{%
    \ifglentryexists{#1}{#2}%
    {%
        \protected@edef\glslabel{\glsdetoklabel{#1}}%
        \glstrundefaction{Glossary entry ‘\glslabel’
            has not been defined}{You need to define a glossary entry before
            you can reference it.}%
    }%
}
\renewcommand{\glsdoifnoexists}[2]{%
    \ifglentryexists{#1}{%
        \glstrundefaction{Glossary entry ‘\glsdetoklabel{#1}’
            has already been defined}{}}{#2}%
    }
\ifdef\glsdoifexistsordo
{%
    \renewcommand{\glsdoifexistsordo}[3]{%
        \ifglentryexists{#1}{#2}%
        {%
            \glstrundefaction{Glossary entry ‘\glsdetoklabel{#1}’
                has not been defined}{You need to define a glossary entry
                before you can use it.}%
            #3%
        }%
    }%
}
{%
    \glstr@warnonexistsordo\glsdoifexistsordo
    \newcommand{\glsdoifexistsordo}[3]{%
        \ifglentryexists{#1}{#2}%
        {%
            \glstrundefaction{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}%
        {%
            \glstrundefaction{Glossary type ‘#1’ already exists}{}%
            #3%
        }%
    }%
}

```

```

        {#2}%
    }%
}
{%
\glstr@warnonexistsordo\doifglossarynoexistsordo
\newcommand{\doifglossarynoexistsordo}[3]{%
    \ifglossaryexists*{#1}%
    {%
        \glstrundefaction{Glossary type ‘#1’ already exists}{}%
        #3%
    }%
    {#2}%
}%
}

\appto\@newglossaryentryposthook{%
    \ifdefvoid\@glo@see
    {\csxdef{glo@\@glo@label @see}{}}%
    {%
        \csxdef{glo@\@glo@label @see}{\@glo@see}%
        \ifglstr@autoseeindex
        \@glstr@autoindexcrossrefs
        \fi
    }%
}
\appto\@gls@keymap{,{see}{see}}
\newcommand*{\glstr@usesee}[1]{%
    \glstoifexists{#1}%
    {%
        \letcs{\@glo@see}{glo\@glstoklabel{#1}@see}%
        \ifdefempty\@glo@see
        {}%
        {%
            \expandafter\glstr@usesee\@glo@see\@end@glstr@usesee
        }%
    }%
}
\newcommand*{\glstr@usesee}[1][\@seenname]{%
    \@glstr@usesee{#1}%
}
\def\@glstr@usesee[#1]#2\@end@glstr@usesee{%
    \glstr@useseeformat{#1}{#2}%
}
\newcommand*{\glstr@useseeformat}[2]{%
    \glseeformat{#1}{#2}%
}
\renewcommand*{\glseeitemformat}[1]{%
    \ifglshashshort{#1}{\glsfmttext{#1}}{\glsfmtname{#1}}%
}
\newcommand*{\glstr@hiername}[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[\seeealsoname]{#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}{\mglssseeitem{#1}}{\glsseeitem{#1}}%
}
\newcommand*{\@glsxtr@seefirstitem}[1]{%
  \glsxtrifmulti{#1}{\mglssseefirstitem{#1}}{\glsseefirstitem{#1}}%
}

```

```

}
\newcommand*{\mglseeitem}[1]{%
  \mglename[all={noindex},setup={hyper=allmain}]{#1}%
}
\newcommand*{\mglseefirstitem}{\mglseeitem}
\newcommand*{\glseefirstitem}{\glseeitem}
\newcommand*{\glseeelastoxfordsep}{\glseeelastsep}
\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\glxtruseealsoformat\glsoopenbrace\string"
      :close \string"\glsclosebrace\string")}%
  }
  \appto\@xdylocationclassorder{\space\string"seealso\string"}
  \newrobustcmd*{\glxtrindexseealso}[2]{%
    \ifx\glxtr@record@setting\glxtr@record@setting@alsoindex
      \@glxtr@recordsee{#1}{#2}%
    \fi
    \glsoifexists{#1}%
    {%
      \@@glxtrwrglossmark
      \def\@glx@xref{#2}%
      \@onelevel@sanitize\@glx@xref
      \@glx@checkmkidxchars\@glx@xref
      \glsglossary{\csname glo@#1@type\endcsname}{%
        (indexentry
          :tkey (\csname glo@#1@index\endcsname)
          :xref (\string"\@glx@xref\string")
          :attr \string"seealso\string"
        )
      }%
    }%
  }
}
{
  \newrobustcmd*{\glxtrindexseealso}{\glsee[\seealsoname]}
}
\ifdef\glset@xr@key
{
  \define@key{glossentry}{alias}{%
    \glset@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}%
\ifglstr@autoindex
\@glstr@autoindexcrossrefs
\fi
}%
\ifdefvoid\@glo@alias
{\csxdef{glo@\@glo@label @alias}{}}}%
{%
\csxdef{glo@\@glo@label @alias}{\@glo@alias}%
}%
}
\newcommand*{\glstralias}[1]{\@gls@entry@field{#1}{alias}}
\newcommand*{\glstrseealsolabels}[1]{\@gls@entry@field{#1}{seealso}}
\appto\@glo@autohook{%
\ifdefvoid\@glo@alias
{%
\ifdefvoid\@glo@seealso
{}}%
{%
\protected@edef\@do@glsee{\noexpand\glstrindexseealso
{\@glo@label}{\@glo@seealso}}%
\@do@glsee
}%
}%
{%
\ifdefvoid\@glo@see
{%
\protected@edef\@do@glsee{\noexpand\glsee{\@glo@label}{\@glo@alias}}%
\@do@glsee
}%
{}}%
}%
}%
}
{
\glsaddstoragekey*{alias}{\glstralias}
\glsaddstoragekey*{seealso}{\glstrseealsolabels}
\appto\@newglossaryentryposthook{%
\ifcsvoid{glo@\@glo@label @alias}%
{%
\ifcsvoid{glo@\@glo@label @seealso}%
{}}%
}
}

```

```

    {%
      \protected@edef\do@glsssee{\noexpand\glxtrindexseealso
        {\@glo@label}{\csuse{glo@\@glo@label @seealso}}}%
      \@do@glsssee
    }%
  }%
  {%
    \ifdefvoid\@glo@see
    {%
      \protected@edef\do@glsssee{\noexpand\glsssee
        {\@glo@label}{\csuse{glo@\@glo@label @alias}}}%
      \@do@glsssee
    }%
  }%
}
}
\AtEndDocument{\ifglxtrindexcrossrefs\glxtraddallcrossrefs\fi}
\newcommand*\glxtraddallcrossrefs{%
  \forallglossaries{\@glo@type}%
  {%
    \forglsentries[\@glo@type]{\@glo@label}%
    {%
      \ifglssused{\@glo@label}%
      {\expandafter\@glxtr@addunusedxrefs\expandafter{\@glo@label}}{%}%
    }%
  }%
}
}
\newcommand*\glxtr@addunusedxrefs[1]{%
  \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@see}%
  \ifdefvoid\@glo@see
  {}%
  {%
    \expandafter\glxtr@addunused\@glo@see\end@glxtr@addunused
  }%
  \letcs{\@glo@see}{glo@\glsdetoklabel{#1}@seealso}%
  \ifdefvoid\@glo@see
  {}%
  {%
    \expandafter\glxtr@addunused\@glo@see\end@glxtr@addunused
  }%
}
\newcommand*\glxtr@addunused[1][1]{%
  \@glxtr@addunused
}
\def\@glxtr@addunused#1\end@glxtr@addunused{%
  \@for\@glxtr@label:=#1\do
  {%
    \glxtrifmulti\@glxtr@label
  }%
}

```

```

\letcs\@glxstr@labellist{\@gls@combined@\@glxstr@label @list}%
\@for\@glxstr@multilabel:=\@glxstr@labellist\do
{\@glxstr@addunused\@glxstr@multilabel\@end@glxstr@addunused}%
}%
{%
\ifglsused{\@glxstr@label}{}%
{%
\glsadd[format=glxstrunusedformat]{\@glxstr@label}%
\glsunset{\@glxstr@label}%
\expandafter\@glxstr@addunusedxrefs\expandafter{\@glxstr@label}%
}%
}%
}%
}
\newcommand*{\glxstrunusedformat}[1]{\unskip}
\ifdef\gls@begindocdefs
{%
\renewcommand*{\gls@begindocdefs}{%
\ifnum\@glxstr@docdefval=1\relax
\@gls@enablesavenonnumberlist
\edef\@gls@restoreat{%
\noexpand\catcode'\noexpand\@=\number\catcode'\@ \relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{\@glxstr@restoreat}{}%
\@gls@restoreat
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
\else
\ifnum\@glxstr@docdefval=3\relax
\@gls@enablesavenonnumberlist
\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\@glxstr@docdefval=3\relax
\PackageError{glossaries-extra}{Package option
'docdef=\@glxstr@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}%
}

```

```

\let\glstr@orgmakenoidxglossaries\makenoidxglossaries
\renewcommand{\makenoidxglossaries}{%
\@domakeglossaries
{%
\ifdefequal\@glstr@record@setting\@glstr@record@setting@off
{%
\glstr@orgmakenoidxglossaries
\renewcommand{\@do@seeglossary}[2]{%
\@glstrwrglossmark
\protected@edef\@gls@label{\glsdetoklabel{##1}}%
\protected@write\@auxout{}{%
\string\@gls@reference
{\csname glo@\@gls@label @type\endcsname}%
{\@gls@label}%
}%
\string\glsseeformat##2}%
}%
}%
\if@glstrdocdefrestricted
\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
\@glstrdocdeffalse
\fi
\disable@keys{glossaries-extra}{docdef}%
}%
{%
\PackageError{glossaries-extra}{\string\makenoidxglossaries\space
not permitted\MessageBreak
with record=\@glstr@record@setting\space package option}%
{You may only use \string\makenoidxglossaries\space with the
record=off option}%
}%
}%
}
\renewcommand*{\@gls@defdocnewglossaryentry}{%
\ifcase\@glstr@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\@glxtr@optlist
\GlossariesExtraWarning{The options '\@glxtr@optlist' have
been ignored for entry '##2' as it has already been defined}%
}
\renewcommand\@printglossary[2]{%
\def\@glxtr@printglossopts{##1}%
\@glxtr@orgprintglossary{##1}{##2}%
\def\@glxtr{\@glxtr@disabledflycommand\glxtr}%
\def\@glxtrpl{\@glxtr@disabledflycommand\glxtrpl}%
\def\@Glsxtr{\@glxtr@disabledflycommand\Glsxtr}%
\def\@Glsxtrpl{\@glxtr@disabledflycommand\Glsxtrpl}%
}
\newcommand*\@glxtr@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}%
\@glxtr@disabledflycommand
}%
\newcommand*\@glxtr@disabledflycommand[2][{}]{##2}
\let\GlsXtrEnableOnTheFly\relax
}
\@onlypreamble\GlsXtrEnableOnTheFly
\newcommand*\@glxtr@current@style{\@glossary@default@style}
\renewcommand*\@setglossarystyle[1]{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries-extra}{Glossary style '#1' undefined}{}%
}%
{%
\csname @glsstyle@#1\endcsname
\protected@edef\@glxtr@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\@glstrpreloctag\@glstrpreloctag
    \let\@glstrpostloctag\@glstrpostloctag
    \renewcommand*{\@glstr@pagetag}{#1}%
    \renewcommand*{\@glstr@pagetag}{#2}%
    \renewcommand*{\@glstr@savepreloctag}[2]{%
      \csgdef{\@glstr@preloctag##1}{##2}%
    }%
    \renewcommand*{\@glstr@doloctag}{%
      \ifcsundef{\@glstr@preloctag\@glscurrententrylabel}%
      {%
        \GlossariesWarning{Missing pre-location tag for ‘\@glscurrententrylabel’.
          Rerun required}%
      }%
      {%
        \csuse{\@glstr@preloctag\@glscurrententrylabel}%
      }%
    }%
  }
}
\@onlypreamble\GlsXtrEnablePreLocationTag
\newcommand*{\@glstrpreloctag}{%
  \let\@glstr@org@delimN\delimN
  \let\@glstr@org@delimR\delimR
  \let\@glstr@org@glsglignore\glsglignore
  \gdef\@glstr@thisloctag{\@glstr@pagetag}%
  \renewcommand*{\delimN}{%
    \gdef\@glstr@thisloctag{\@glstr@pagetag}%
    \@glstr@org@delimN}%
  \renewcommand*{\delimR}{%
    \gdef\@glstr@thisloctag{\@glstr@pagetag}%
    \@glstr@org@delimR}%
  \renewcommand*{\glsglignore}[1]{%
    \gdef\@glstr@thisloctag{\relax}%
    \@glstr@org@glsglignore{##1}}%
  \@glstr@doloctag
}
\newcommand*{\@glstrpreloctag}{%
\newcommand*{\@glstr@pagetag}{%
\newcommand*{\@glstr@pagetag}{%
\newcommand*{\@glstrpostloctag}{%
  \let\delimN\@glstr@org@delimN
  \let\delimR\@glstr@org@delimR
  \let\glsglignore\@glstr@org@glsglignore
  \protected@write\@auxout{%
    {\string\@glstr@savepreloctag\@glscurrententrylabel}{\@glstr@thisloctag}}%

```

```

}
\newcommand*{\@glstrpostloctag}{%
\newcommand*{\@glstrsavepreloctag}[2]{%
\protected@write\@auxout{}{%
\string\providecommand\string\@glstrsavepreloctag[2]{}%
\newcommand*{\@glstrdoloctag}{%
\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{%
\@glstrpreloctag
\GlsXtrFormatLocationList{##1}%
\@glstrpostloctag
\gls@save@numberlist{##1}}%
\fi
}%
}
\renewcommand*{\glsentryfmt}{%
\ifglshasshort{\glslabel}{\glssetabbrvfmt{\glscategory{\glslabel}}}{%
\glsifregular{\glslabel}%
{\glstrregularfont{\glsentryfmt}}%
}%
\ifglshasshort{\glslabel}%
{\glstrabbreviationfont{\glstrgenabbrvfmt}}%
{\glstrregularfont{\glsentryfmt}}%
}%
}
\newcommand*{\glstrregularfont}[1]{#1}
\newcommand*{\glstrabbreviationfont}[1]{#1}
\renewcommand{\@gls@field@link}[4][1]{%
\@glstr@record{#2}{#3}{\glslink}%
\glsdoifexists{#3}%
{%
\let\glstrorg@ifKV@glslink@hyper@ifKV@glslink@hyper
\@gls@save@glslocal
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glscustomtext{#4}%
\@glstr@field@linkdefs
#1%
\@gls@link[#2]{#3}{#4}%
\let@ifKV@glslink@hyper\glstrorg@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\gls caps case\@firstofthree
    \def\gls custom text{#3}%
    \def\gls insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link{#1}{#2}{\@glo@text}%
    \@gls@do@gls@unset{#2}%
  }%
  \gls post link hook
}
\renewcommand*{\@gls@@link}[3][{}]{%
  \@glsxtr@record{#1}{#2}{glslink}%
  \glsdoifexists sordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

```

```

\def\glscustomtext{#3}%
\@glstr@field@linkdefs
\@gl@link[#1]{#2}{#3}%
}%
{%
\glstextformat{#3}%
}%
\glspostlinkhook
}
\newcommand*{\glstr@init@wrgloss}{%
\gl@if@attribute{\glsl@label}{wrgloss}{after}%
{%
\glstr@init@wrgloss@before@false
}%
{%
\glstr@init@wrgloss@before@true
}%
}
\newif\ifglstr@init@wrgloss@before
\glstr@init@wrgloss@before@true
\define@choicekey{glsl@link}{wrgloss}%
[\@glstr@wrgloss@val\@glstr@wrgloss@nr]%
{before,after}%
{%
\ifcase\@glstr@wrgloss@nr\relax
\glstr@init@wrgloss@before@true
\or
\glstr@init@wrgloss@before@false
\fi
}
\define@key{glsl@link}{thevalue}{\def\@glstr@thevalue{#1}}
\define@key{glsl@link}{theHvalue}{\def\@glstr@theHvalue{#1}}
\define@boolkey{glsl@link}[glstr@]{hyperoutside}[true]{%
\glstr@hyperoutside@true
}
\define@key{glsl@link}{textformat}{%
\ifcsdef{#1}
{%
\letcs{\@glstr@local@textformat}{#1}%
}%
{%
\PackageError{glossaries-extra}{Unknown control sequence name ‘#1’}{}%
}%
}
\define@key{glsl@link}{prefix}{\def\glolink@prefix{#1}}
\newcommand*{\glstr@init@hyperoutside}{%
\gl@if@attribute{\glsl@label}{hyperoutside}{false}%
{%
\glstr@hyperoutside@false
}%
{%

```

```

\glstr@hyperoutsidetrue
}%
}
\newcommand*\glstr@inc@linkcount{}\}
\newcommand*\glslinkpresetkeys{}\}
\newrobustcmd*\GlsXtrExpandedFmt}[2]{%
\protected@edef\glstr@tmp{#2}%
\expandafter#1\expandafter{\glstr@tmp}%
}
\newcommand*\glstr@use@equation@counter{%
\glstr@ifnum@mmode{\def\gls@counter{equation}}{}\}
}
\newcommand*\glstr@do@autoadd}[1]{%
\newcommand*\GlsXtrAutoAddOnFormat}[3][\glslabel]{%
\renewcommand*\glstr@do@autoadd}[1]{%
\begingroup
\protected@edef\glstr@do@autoadd{%
\noexpand\ifstrequal{##1}{\glslink}%
{%
\noexpand\DTLifinlist{\glsnumberformat}{#2}{\noexpand\glsadd[format={\glsnumberformat},%
}%
}%
}%
\glstr@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\glstr@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\glstr@org\glslink@prefix\glslink@prefix
\let\glstr@local@textformat\relax
\def\glstr@thevalue{}%
\def\glstr@theHvalue{\glstr@thevalue}%
\glstr@initwrgloss
\glstr@inithyperoutside
\gls@setdefault\glslink@opts
\glstr@inc@linkcount
\if\glstr@equations
\glstr@use@equation@counter
\fi
\do\glsdisablehyperinlist

```

```

\do@gls@link@checkfirsthyper
\glslinkpresetkeys
\setkeys{glslink}{#1}%
\glstr@do@autoadd{glslink}%
\glslinkpostsetkeys
\ifdefempty{\@glstr@thevalue}%
{%
  \@gls@saveentrycounter
}%
{%
  \let\theglsentrycounter\@glstr@thevalue
  \def\theHglentrycounter{\@glstr@theHvalue}%
}%
\@gls@setsort{\glslabel}%
\ifx\@glstr@local@textformat\relax
  \glshasattribute{\glslabel}{textformat}%
  {%
    \protected@edef\@glstr@attrval{\glsgetattribute{\glslabel}{textformat}}%
    \ifcsdef{\@glstr@attrval}%
    {%
      \letcs{\@glstr@textformat}{\@glstr@attrval}%
    }%
    {%
      \GlossariesExtraWarning{Unknown control sequence name
        '\@glstr@attrval' supplied in textformat attribute
        for entry '\glslabel'. Reverting to default \string\glstextformat}%
      \let\@glstr@textformat\glstextformat
    }%
  }%
  {%
    \let\@glstr@textformat\glstextformat
  }%
\else
  \let\@glstr@textformat\@glstr@local@textformat
\fi
\glslinkwrcontent
{%
  \ifglstr@initwrglossbefore
    \@do@wrglossary{#2}%
  \fi
  \ifKV@glslink@hyper
    \ifglstr@hyperoutside
      \@glslink{\glolinkprefix\glslabel}{\@glstr@textformat{#3}}%
    \else
      \@glslink{\@glstr@textformat{\@glslink{\glolinkprefix\glslabel}{#3}}}%
    \fi
  \else
    \ifglstr@hyperoutside
      \glsdonohyperlink{\glolinkprefix\glslabel}{\@glstr@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\theHglentrycounter{\@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]{%
  \ifgl Sentryexists{#1}%
  {%
    \ifglshashshort{#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{\GLSaccentstext{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@Glstext@#1#2[#3]{%
  \glsxtrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@secondofthree]{#1}{#2}%
  {\@gls@field@font{\Glsaccentstext{#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]{%
  \glstrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glstrifwasfirstuse\@firstoftwo
   \let\glsifplural\@firstoftwo
   \let\glscapscase\@secondofthree
   \glstrchecknohyperfirst{#2}%
  ]%
  {#1}{#2}{\@gls@field@font{\Glsaccessfirstplural{#2}#3}}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \glstrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glstrifwasfirstuse\@firstoftwo
   \let\glsifplural\@firstoftwo
   \let\glscapscase\@thirdofthree
   \glstrchecknohyperfirst{#2}%
  ]%
  {#1}{#2}%
  {\@gls@field@font{\GLSaccessfirstplural{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glsname@#1#2[#3]{%
  \glstrassignfieldfont{#2}%
  \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccessname{#2}#3}}%
}
\def\@Glsname@#1#2[#3]{%
  \glstrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]{#1}{#2}%
  {\@gls@field@font{\Glsaccessname{#2}#3}}%
}
\def\@GLSname@#1#2[#3]{%
  \glstrassignfieldfont{#2}%
  \@gls@field@link[\let\glscapscase\@thirdoftwo]%
  {#1}{#2}%
  {\@gls@field@font{\GLSaccessname{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@glsdesc@#1#2[#3]{%
  \glstrassignfieldfont{#2}%
  \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccessdesc{#2}#3}}%
}
\def\@Glsdesc@#1#2[#3]{%
  \glstrassignfieldfont{#2}%
  \@gls@field@link
  [\let\glscapscase\@secondoftwo]{#1}{#2}%
  {\@gls@field@font{\Glsaccessdesc{#2}#3}}%
}
\def\@GLSdesc@#1#2[#3]{%
  \glstrassignfieldfont{#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\@glsymbol@#1#2[#3]{%
    \glsxtrassignfieldfont{#2}%
    \@gls@field@link{#1}{#2}{\@gls@field@font{\glsaccesssymbol{#2}#3}}%
}
\def\@GLsymbol@#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\@glsymbolplural@#1#2[#3]{%
    \glsxtrassignfieldfont{#2}%
    \@gls@field@link
    [\let\glscapscase\@secondoftwo
    \let\glsifplural\@firstoftwo
    ]{#1}{#2}{\@gls@field@font{\glsaccesssymbolplural{#2}#3}}%
}
\def\@GLsymbolplural@#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]{%
\glstrassignfieldfont{#2}%
\@gls@field@link
[\let\glscapscase\@thirdoftwo
\let\glsifplural\@firstoftwo
]%
{#1}{#2}%
{\@gls@field@font{\Glsaccesssymbolplural{#2}\mfirstucMakeUppercase{#3}}}%
}
\def\@Glsuseri@#1#2[#3]{%
\glstrassignfieldfont{#2}%
\@gls@field@link
[\let\glscapscase\@secondoftwo]{#1}{#2}%
{\@gls@field@font{\Glsentryuseri{#2}#3}}%
}
\def\@GLSuseri@#1#2[#3]{%
\glstrassignfieldfont{#2}%
\@gls@field@link[\let\glscapscase\@thirdoftwo]%
{#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}}%
}
\def\@Glsuserii@#1#2[#3]{%
\glstrassignfieldfont{#2}%
\@gls@field@link
[\let\glscapscase\@secondoftwo]%
{#1}{#2}{\@gls@field@font{\Glsentryuserii{#2}#3}}%
}
\def\@GLSuserii@#1#2[#3]{%
\glstrassignfieldfont{#2}%
\@gls@field@link[\let\glscapscase\@thirdoftwo]%
{#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}}%
}
\def\@Glsuseriii@#1#2[#3]{%
\glstrassignfieldfont{#2}%
\@gls@field@link
[\let\glscapscase\@secondoftwo]%
{#1}{#2}{\@gls@field@font{\Glsentryuseriii{#2}#3}}%
}
\def\@GLSuseriii@#1#2[#3]{%
\glstrassignfieldfont{#2}%
\@gls@field@link[\let\glscapscase\@thirdoftwo]%
{#1}{#2}{\@gls@field@font{\mfirstucMakeUppercase{\glsentryuseriii{#2}#3}}}%
}
\def\@Glsuseriv@#1#2[#3]{%
\glstrassignfieldfont{#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\@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\@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}}}%
}
}
\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]{%
\@glstr@base@acrcmd\Acrshort\Glsxtrshort
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glxtrifwasfirstuse\@secondoftwo
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsaccessshort{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\@glstr@base@acrcmd\ACRshort\GLSxtrshort
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glxtrifwasfirstuse\@secondoftwo
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\mfirstucMakeUppercase{\acronymfont{\Glsaccessshort{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrshortpl#1#2[#3]{%
\@glstr@base@acrcmd\acrshortpl\glxtrshortpl
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glxtrifwasfirstuse\@secondoftwo
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsaccessshortpl{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

```

```

\def\@Acrshortpl#1#2[#3]{%
  \@glstr@base@acrcmd\Acrshortpl\Glsxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\gl@ifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccessshortpl{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRshortpl#1#2[#3]{%
  \@glstr@base@acrcmd\ACRshortpl\GLSxtrshortpl
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\gl@ifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\Glsaccessshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \@glstr@base@acrcmd\acrlong\glxtrlong
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\gl@ifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccesslong{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
  \@glstr@base@acrcmd\Acrlong\Glsxtrlong

```

```

\glsdoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \let\glxtrifwasfirstuse\@secondoftwo
  \let\gl@ifplural\@secondoftwo
  \let\glscapscase\@secondofthree
  \let\glinsert\@empty
  \def\glscustomtext{%
    \acronymfont{\Glsaccesslong{#2}}#3%
  }%
  \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
  \@glxtr@base@acrcmd\ACRlong\GLSxtrlong
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\gl@ifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\Glsaccesslong{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@acrlongpl#1#2[#3]{%
  \@glxtr@base@acrcmd\acrlongpl\glxtrlongpl
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\gl@ifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsaccesslongpl{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrlongpl#1#2[#3]{%
  \@glxtr@base@acrcmd\Acrlongpl\GLSxtrlongpl
  \glsdoifexists{#2}%
  {%

```



```

\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\gl@xtrifwasfirstuse\@secondoftwo
\let\gl@sifplural\@firstoftwo
\let\gl@scapscase\@secondofthree
\let\gl@insert\@empty
\def\gl@customtext{%
  \acronymfont{\Glsaccesslongpl{#2}}{#3}%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\gl@postlinkhook
}
\def\@ACRlongpl#1#2[#3]{%
  \@gl@xtr@base@acrcmd\ACRlongpl\GLSxtrlongpl
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\gl@xtrifwasfirstuse\@secondoftwo
    \let\gl@sifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@insert\@empty
    \def\gl@customtext{%
      \mfirstucMakeUppercase{\acronymfont{\Glsaccesslongpl{#2}}{#3}}%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \gl@postlinkhook
}
\def\@acrfull#1#2[#3]{%
  \@gl@xtr@base@acrcmd\acrfull\gl@xtrfull
  \acrfullfmt{#1}{#2}{#3}%
}
\def\@Acrfull#1#2[#3]{%
  \@gl@xtr@base@acrcmd\Acrfull\Glsxtrfull
  \Acrfullfmt{#1}{#2}{#3}%
}
\def\@ACRfull#1#2[#3]{%
  \@gl@xtr@base@acrcmd\ACRfull\GLSxtrfull
  \ACRfullfmt{#1}{#2}{#3}%
}
\def\@acrfullpl#1#2[#3]{%
  \@gl@xtr@base@acrcmd\acrfullpl\gl@xtrfullpl
  \acrfullplfmt{#1}{#2}{#3}%
}
\def\@Acrfullpl#1#2[#3]{%
  \@gl@xtr@base@acrcmd\Acrfullpl\Glsxtrfullpl
  \Acrfullplfmt{#1}{#2}{#3}%
}
\def\@ACRfullpl#1#2[#3]{%
  \@gl@xtr@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\gls@scaps@case\@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}{%
    \ifglslused{\glslabel}%
        {\let\glsxtrifwasfirstuse\@secondoftwo}
        {\let\glsxtrifwasfirstuse\@firstoftwo}%
    \protected@edef\glscategorylabel{\glscategory{\glslabel}}%
    \ifglslused{\glslabel}%
    {%
        \glsifcategoryattribute{\glscategorylabel}{nohypernext}{true}%
        {\KV@glslink@hyperfalse}{}%
    }%
    {%
        \glsifcategoryattribute{\glscategorylabel}{nohyperfirst}{true}%
        {\KV@glslink@hyperfalse}{}%
    }%
    \glslinkcheckfirsthyperhook
}
\ifdef\do@glstdisablehyperinlist

```

```

{%
  \let\@glstr@do@gl:disablehyperinlist\do@gl:disablehyperinlist
  \renewcommand*{\do@gl:disablehyperinlist}{%
    \@glstr@do@gl:disablehyperinlist
    \gl@ifattribute{\glslabel}{nohyper}{true}{\KV@glslink@hyperfalse}{}%
  }
}
{}
\define@boolkey{glslink}{noindex}[true]{}
\KV@glslink@noindexfalse
\providecommand*{\@gl:save@gl:local}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\providecommand*{\@gl:restore@gl:local}{%
  \ifKV@glslink@local
    \let\@gl:do@gl:sunset\gl:localunset
  \else
    \let\@gl:do@gl:sunset\gl:sunset
  \fi
}
\providecommand*{\@gl:do@gl:sunset}[1]{\gl:sunset{#1}}
\ifdef\@gl:setdefault@glslink@opts
{
  \renewcommand*{\@gl:setdefault@glslink@opts}{%
    \KV@glslink@noindexfalse
    \@glstrsetaliasnoindex
  }
}
{
  \newcommand*{\@gl:setdefault@glslink@opts}{%
    \KV@glslink@noindexfalse
    \@glstrsetaliasnoindex
  }
}
\preto\do@gl:disablehyperinlist{\@gl:setdefault@glslink@opts}
}
\providecommand*{\glstrsetaliasnoindex}{%
  \KV@glslink@noindextrue
}
\newcommand*{\@glstrsetaliasnoindex}{%
  \ifcvoid{glo@\gl:detoklabel{\glslabel}@alias}%
  {}%
}
{%
  \let\glstrindexaliased\@glstrindexaliased
  \glstrsetaliasnoindex
  \let\glstrindexaliased\@no@glstrindexaliased
}%
}
\newcommand{\@glstrindexaliased}{%
  \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}{%
\glsxtrdoautoindexname{\@gls@label}{dualindex}%
}

```

```

}
\newcommand*{\glstrdowrglossaryhook}[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][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\@glstr@record@setting\@glstr@record@setting@off
{}%
{%
\protected@write\@auxout{}{\string\providecommand{\string\@glstr@altmodifier}[1]{}}%
\protected@write\@auxout{}{\string\@glstr@altmodifier{#1}}%
}%
}
\let\glstr@org@dohyperlink\glsdohyperlink
\ifdef\glsnahyperlink
{
\renewcommand*{\glsnahyperlink}[3][\@glo@type]{%
\protected@edef\gls@grplabel{#2}\protected@edef\@gls@grptitle{#3}%
{%
\let\glstrdohyperlink\glstr@org@dohyperlink
\@glslink{\glsnahyperlinkname{#1}{#2}}{#3}%
}%
}%
}

```

```

}
{}
\ifdef\@gls@navhypertarget
{}
{%
\renewcommand*\glsnavhypertarget{\protect\@gls@navhypertarget}
\newcommand*\@gls@navhypertarget[3][\@gls@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@hypergroup@list{#1}%
{%
\letcs\@gls@list{\@gls@hypergroup@list{#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@hypergroup@prerun{%
\GlossariesWarningNoLine{Navigation panel
has changed. Rerun LaTeX}}%
}%
}%
}%
{%
\GlossariesWarningNoLine{Navigation panel
for glossary type ‘#1’^^Jmissing group ‘#2’}%
\gdef\gls@hypergroup@prerun{%
\GlossariesWarningNoLine{Navigation panel
has changed. Rerun LaTeX}}%
}%
}%
}
{}
\newcommand*\glsxtrdohyperlink[2]{%
\gls@hasattribute{\glslabel}{targeturl}%
{%
\gls@hasattribute{\glslabel}{targetname}%
{%
\gls@hasattribute{\glslabel}{targetcategory}%
{%
\hyperref{\gls@getattribute{\glslabel}{targeturl}}%
{\gls@getattribute{\glslabel}{targetcategory}}%
{\gls@getattribute{\glslabel}{targetname}}%
{\glsxtrprotectlinks#2}}%
}%
}%
}%

```

```

    {%
      \hyperref{\glsgetattribute{\glslabel}{targeturl}}{%
        {%
          {\glsgetattribute{\glslabel}{targetname}}%
          {\glsxtrprotectlinks#2}}%
        }%
      }%
    {%
      \href{\glsgetattribute{\glslabel}{targeturl}}{%
        {\glsxtrprotectlinks#2}}%
      }%
    }%
    {%
      \glsfieldfetch{\glslabel}{alias}{\gloaliaslabel}%
      \ifvoid\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\@gls@target\@secondoftwo
}
\renewcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \def\@glslink{\glsxtrdohyperlink}%
}

```



```

\def\@glstarget{\glsdohypertarget}%
}
\def\glndonohyperlink#1#2{{\glxtrprotectlinks #2}}
\ifcsundef{hyperlink}%
{%
\def\@glslink{\glndonohyperlink}
}%
{%
\def\@glslink{\glxtrdohyperlink}
}
\newcommand*{\glxtrprotectlinks}{%
\KV@glslink@hyperfalse
\KV@glslink@noindextrue
\let\@gls@\@glxtr@p@text@
\let\@Gls@\@Glsxtr@p@text@
\let\@GLS@\@GLSxtr@p@text@
\let\@glspl@\@glxtr@p@plural@
\let\@Glspl@\@Glsxtr@p@plural@
\let\@GLSpl@\@GLSxtr@p@plural@
\let\@glxtrshort\@glxtr@p@short@
\let\@Glsxtrshort\@Glsxtr@p@short@
\let\@GLSxtrshort\@GLSxtr@p@short@
\let\@glxtrlong\@glxtr@p@long@
\let\@Glsxtrlong\@Glsxtr@p@long@
\let\@GLSxtrlong\@GLSxtr@p@long@
\let\@glxtrshortpl\@glxtr@p@shortpl@
\let\@Glsxtrshortpl\@Glsxtr@p@shortpl@
\let\@GLSxtrshortpl\@GLSxtr@p@shortpl@
\let\@glxtrlongpl\@glxtr@p@longpl@
\let\@Glsxtrlongpl\@Glsxtr@p@longpl@
\let\@GLSxtrlongpl\@GLSxtr@p@longpl@
\let\@acrshort\@glxtr@p@acrshort@
\let\@Acrshort\@Glsxtr@p@acrshort@
\let\@ACRshort\@GLSxtr@p@acrshort@
\let\@acrshortpl\@glxtr@p@acrshortpl@
\let\@Acrshortpl\@Glsxtr@p@acrshortpl@
\let\@ACRshortpl\@GLSxtr@p@acrshortpl@
\let\@acrlong\@glxtr@p@acrlong@
\let\@Acrlong\@Glsxtr@p@acrlong@
\let\@ACRlong\@GLSxtr@p@acrlong@
\let\@acrlongpl\@glxtr@p@acrlongpl@
\let\@Acrlongpl\@Glsxtr@p@acrlongpl@
\let\@ACRlongpl\@GLSxtr@p@acrlongpl@
}
\def\@glxtr@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\@glxtr@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\@glxstr@p@short@#1#2[#3]{%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\glsabbrvfont{\glsentryshort{#2}}#3%
}%
}
\def\@Glsxtr@p@short@#1#2[#3]{%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\glsabbrvfont{\Glsentryshort{#2}}#3%
}%
}
\def\@GLSxtr@p@short@#1#2[#3]{%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\mfirstucMakeUppercase{\glsabbrvfont{\glsentryshort{#2}}#3}%
}%
}
\def\@glxstr@p@shortpl@#1#2[#3]{%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\glsabbrvfont{\glsentryshortpl{#2}}#3%
}%
}
\def\@Glsxtr@p@shortpl@#1#2[#3]{%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\glsabbrvfont{\Glsentryshortpl{#2}}#3%
}%
}
\def\@GLSxtr@p@shortpl@#1#2[#3]{%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\mfirstucMakeUppercase{\glsabbrvfont{\glsentryshortpl{#2}}#3}%
}%
}
\def\@glxstr@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\@glxstr@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\@glxstr@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\@glxstr@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{\glentryshortpl{#2}}#3}}}
\def\@glxtr@p@acrlong@#1#2[#3]{\glentrylong{#2}#3}
\def\@Glsxtr@p@acrlong@#1#2[#3]{\Glsentrylong{#2}#3}
\def\@GLSxtr@p@acrlong@#1#2[#3]{%
  {\mfirstucMakeUppercase{\glentrylong{#2}#3}}}
\def\@glxtr@p@acrlongpl@#1#2[#3]{\glentrylongpl{#2}#3}
\def\@Glsxtr@p@acrlongpl@#1#2[#3]{\Glsentrylongpl{#2}#3}
\def\@GLSxtr@p@acrlongpl@#1#2[#3]{%
  {\mfirstucMakeUppercase{\glentrylongpl{#2}#3}}}
\newcommand*{\@glxtrp@opt}{hyper=false,noindex}
\newcommand*{\glxtrsetpopts}[1]{%
  \renewcommand*{\@glxtrp@opt}{#1}%
}
\newcommand*{\glossxtrsetpopts}{%
  \glxtrsetpopts{noindex}%
}
\newrobustcmd*{\@@glxtrp}[2]{%
  {%
    \let\glspostlinkhook\relax
    \csname#1\expandafter\endcsname\expandafter[\@glxtrp@opt]{#2}[]%
  }%
}
\newrobustcmd*{\@glxtrp}[2]{%
  \ifcsdef{gls#1}%
  {%
    \@glxtrp{gls#1}{#2}%
  }%
  {%
    \ifcsdef{glsxtr#1}%
    {%
      \@glxtrp{glsxtr#1}{#2}%
    }%
    {%
      \PackageError{glossaries-extra}{‘#1’ not recognised by
        \string\glxtrp}{}%
    }%
  }%
}
\newrobustcmd*{\@Glsxtrp}[2]{%
  \ifcsdef{Gls#1}%
  {%
    \@glxtrp{Gls#1}{#2}%
  }%
  {%
    \ifcsdef{Glsxtr#1}%
    {%
      \@glxtrp{Glsxtr#1}{#2}%
    }%
    {%

```

```

        \PackageError{glossaries-extra}{‘#1’ not recognised by
        \string\Glsxtrp}{}%
    }%
}%
}
\newrobustcmd*{\@GLSxtrp}[2]{%
    \ifcsdef{GLS#1}%
    {%
        \@@glxtrp{GLS#1}{#2}%
    }%
    {%
        \ifcsdef{GLSxtr#1}%
        {%
            \@@glxtrp{GLSxtr#1}{#2}%
        }%
        {%
            \PackageError{glossaries-extra}{‘#1’ not recognised by
            \string\GLSxtrp}{}%
        }%
    }%
}%
}
\newrobustcmd*{\@glxtr@headentry@p}[2]{%
    \glsifattribute{#1}{headuc}{true}%
    {%
        \mfirstucMakeUppercase{\@gls@entry@field{#1}{#2}}%
    }%
    {%
        \@gls@entry@field{#1}{#2}%
    }%
}%
}
\ifdef\texorpdfstring
{
    \newcommand{\glxtrp}[2]{%
        \protect\NoCaseChange
        {%
            \protect\texorpdfstring
            {%
                \protect\glxtrifinmark
                {%
                    \ifcsdef{glxtrhead#1}%
                    {%
                        {\protect\csuse{glxtrhead#1}{#2}}%
                    }%
                    {%
                        \glxtr@headentry@p{#2}{#1}%
                    }%
                }%
            }%
            {%
                \@glxtrp{#1}{#2}%
            }%
        }%
    }%
}

```

```

    }%
    {%
    \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}%
    }%
  }%
}

```

```

        {%
        \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*{\@glxtrbuffer@nodup@unset}[1]{%
  \expandafter\ifinlist\expandafter{#1}{\@glxtr@unset@buffer}{}%
  {\listxadd\@glxtr@unset@buffer{#1}}%
}
\newcommand*{\GlsXtrStopUnsetBuffering}{%
  \ifstar\s@GlsXtrStopUnsetBuffering\@GlsXtrStopUnsetBuffering
}
\newcommand*{\@GlsXtrStopUnsetBuffering}{%
  \let\@glsunset\@glxtr@unset
  \forlistloop\@glsunset\@glxtr@unset@buffer
  \let\@glxtr@unset@buffer\@glxtr@org@unset@buffer
}
\newcommand*{\s@GlsXtrStopUnsetBuffering}{%
  \forlistloop\@glslounset\@glxtr@unset@buffer
  \let\@glsunset\@glxtr@unset
}
\newcommand*{\GlsXtrDiscardUnsetBuffering}{%
  \let\@glsunset\@glxtr@unset
  \let\@glxtr@unset@buffer\@glxtr@org@unset@buffer
}
\newcommand*{\GlsXtrForUnsetBufferedList}[1]{%
  \forlistloop#1\@glxtr@unset@buffer
}
\renewcommand*{\@glslounset}[1]{%
  \@glslounset{#1}%
  \glxtrpostlocalunset{#1}%
}%
\newcommand*{\glxtrpostlocalunset}[1]{}
\renewcommand*{\@glxreset}[1]{%
  \@glxreset{#1}%
  \glxtrpostreset{#1}%
}%
\newcommand*{\glxtrpostreset}[1]{}
\renewcommand*{\@glslreset}[1]{%
  \@glslreset{#1}%
  \glxtrpostlocalreset{#1}%
}%
\newcommand*{\glxtrpostlocalreset}[1]{}
\newcommand*{\glslreseteach}[1]{%
  \glsl@ifnotmeasuring
  {%
    \@for\@glsl@thislabel:=#1\do{%
      \glsl@ifexists{\@glsl@thislabel}%
      {%
        \@glslreset{\@glsl@thislabel}%
      }%
    }%
  }%
}%
}

```



```

\newcommand*{\glslocalunseteach}[1]{%
  \gls@ifnotmeasuring
  {%
    \@for\@gls@thislabel:=#1\do{%
      \glsdoifexists{\@gls@thislabel}%
      {%
        \@glslocalunset{\@gls@thislabel}%
      }%
    }%
  }%
}

\newcommand*{\GlsXtrEnableEntryCounting}[2]{%
  \glsenableentrycount
  \renewcommand*{\gls}{\cgl}%
  \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}}%
}%

```

```

\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}{%
    \gls@hasattribute{\@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]{%
\gls@hasattribute{#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}%
    {%

```

```

        \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\mfirstuc\MakeUppercase\expandafter{\cglSformat{#1}{#2}}%
}
\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\mfirstuc\MakeUppercase\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@glsggetattribute{#1}{unitcount}.%
  \csuse{glsxtr@theunit@glsggetattribute{#1}{unitcount}}%
}
\newcommand*\@glsxtr@previousunitcount[1]{%
  glo@glsdetoklabel{#1}@prevunit@glsggetattribute{#1}{unitcount}.%
  \csuse{glsxtr@theunit@glsggetattribute{#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}%
        {\glsggetattribute{#1}{unitcount}.%
          \csuse{glsxtr@theunit@glsggetattribute{#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}%
        {\glsggetattribute{#1}{unitcount}.%
          \csuse{glsxtr@theunit@glsggetattribute{#1}{unitcount}}}%
      }%
    }%
    {%
      \csedef{\@glsxtr@csname}%
        {\number\numexpr\csname\@glsxtr@csname\endcsname+1}%
      }%
  }%
  {}%
}
\newcommand*\@glsxtr@currunitcount}[2]{%

```

```

\ifcsundef
{glo@glstdetoklabel{#1}@currunit@#2}%
{0}%
{\csuse{glo@glstdetoklabel{#1}@currunit@#2}}%
}%
\newcommand*{\glstr@prevunitcount}[2]{%
\ifcsundef
{glo@glstdetoklabel{#1}@prevunit@#2}%
{0}%
{\csuse{glo@glstdetoklabel{#1}@prevunit@#2}}%
}%
\newcommand*{\glsenableentryunitcount}{%
\appto@newglossaryentry@defcounters{\@newglossaryentry@defunitcounters}%
\renewcommand*{\glstdocnewglossaryentry}{%
\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*{\glstrycurrcount}[1]{%
\@glstr@currunitcount{##1}{\glstrgetattribute{##1}{unitcount}.%
\csuse{glstr@theunit@glstrgetattribute{##1}{unitcount}}}%
}%
\newcommand*{\glstryprevcount}[1]{%
\@glstr@prevunitcount{##1}{\glstrgetattribute{##1}{unitcount}.%
\csuse{glstr@theunit@glstrgetattribute{##1}{unitcount}}}%
}%
\newcommand*{\glstryprevtotalcount}[1]{%
\ifcsundef{glo@glstdetoklabel{##1}@prevunittotal}%
{0}%
{%
\number\csuse{glo@glstdetoklabel{##1}@prevunittotal}
}%
}%
\newcommand*{\glstryprevmaxcount}[1]{%
\ifcsundef{glo@glstdetoklabel{##1}@prevunitmax}%
{0}%
{%
\number\csuse{glo@glstdetoklabel{##1}@prevunitmax}
}%
}%
\let\@glstr@entryunitcount@org@unset\glstrpostunset
\renewcommand*{\glstrpostunset}[1]{%
\@glstr@entryunitcount@org@unset{##1}%
\@glstr@increment@currunitcount{##1}%
}%
\let\@glstr@entryunitcount@org@localunset\glstrpostlocalunset

```

```

\renewcommand*{\glxstrpostlocalunset}[1]{%
  \@glxstr@entryunitcount@org@localunset{##1}%
  \@glx@local@increment@currunitcount{##1}%
}%
\let\@glxstr@entryunitcount@org@reset\glxstrpostreset
\renewcommand*{\glxstrpostreset}[1]{%
  \glshasattribute{##1}{unitcount}%
  {%
    \protected@edef\@glxstr@csname{\@glxstr@currentunitcount{##1}}%
    \ifcsundef{\@glxstr@csname}%
    {}%
    {\csgdef{\@glxstr@csname}{0}}%
  }%
  {}%
}%
\let\@glxstr@entryunitcount@org@localreset\glxstrpostlocalreset
\renewcommand*{\glxstrpostlocalreset}[1]{%
  \@glxstr@entryunitcount@org@localreset{##1}%
  \glshasattribute{##1}{unitcount}%
  {%
    \protected@edef\@glxstr@csname{\@glxstr@currentunitcount{##1}}%
    \ifcsundef{\@glxstr@csname}%
    {}%
    {\csdef{\@glxstr@csname}{0}}%
  }%
  {}%
}%
\let\@cgls@\@cgls@
\let\@cglspl@\@cglspl@
\let\@cGls@\@cGls@
\let\@cGlspl@\@cGlspl@
\let\@cGLS@\@cGLS@
\let\@cGLSpl@\@cGLSpl@
\AtEndDocument{\@glx@write@entryunitcounts}%
\renewcommand*{\@glx@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\cuse{glo@glsdetoklabel{##1}@prevunittotal}+##2}%
    }%
    \ifcsundef{glo@glsdetoklabel{##1}@prevunitmax}%
    {\csgdef{glo@glsdetoklabel{##1}@prevunitmax}{##2}}%
    {%
      \ifnum\cuse{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}{%
    \gls@hasattribute{\@glsentry}{unitcount}%
    {%
      \ifglsused{\@glsentry}%
      {%
        \forlistcsloop
        {\@gls@write@entryunitcounts@do}%
        {\gls@{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}{\cglsl}%
  \renewcommand*{\Gls}{\cGls}%
  \renewcommand*{\glspl}{\cglspl}%
  \renewcommand*{\Glspl}{\cGlspl}%
  \renewcommand*{\GLS}{\cGLS}%

```

```

\renewcommand*{\GLSpl}{\cGLSpl}%
\@glxtr@setentryunitcountunsetattr{#1}{#2}{#3}%
\let\GlsXtrEnableEntryUnitCounting\@glxtr@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*{\@glxtr@setentryunitcountunsetattr}[3]{%
  \@for\@glxtr@cat:=#1\do
  {%
    \ifdefempty{\@glxtr@cat}{}%
    {%
      \glsssetcategoryattribute{\@glxtr@cat}{entrycount}{#2}%
      \glsssetcategoryattribute{\@glxtr@cat}{unitcount}{#3}%
    }%
  }%
}
\renewcommand*{\SetGenericNewAcronym}{%
  \ifdefequal\@addtoacronymlists\@glxtr@org@addtoacronymlists
  {%
    {%
      \GlossariesWarning{\string\SetGenericNewAcronym\space used
        without restoring base acronym functions with
        \string\RestoreAcronyms}%
    }%
    \let\@Gls@entryname\@Gls@acrentryname
    \renewcommand{\newacronym}[4][{}]{%
      \ifdefempty{\@glssacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
    }%
    \glsskeylisttok{##1}%
    \glsslabeltok{##2}%
    \glssshorttok{##3}%
    \glsslongtok{##4}%
    \newacronymhook
    \protected@edef\@do@newglossaryentry{%
      \noexpand\newglossaryentry{\the\glsslabeltok}%
      {%
        type=\acronymtype,%
        name={\expandonce{\acronymentry{##2}}},%
        sort={\acronymsort{\the\glssshorttok}{\the\glsslongtok}},%
        text={\the\glssshorttok},%
        short={\the\glssshorttok},%
        shortplural={\the\glssshorttok\noexpand\acrpluralsuffix},%
        long={\the\glsslongtok},%

```

```

        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@addtoacronymlists\@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]{%

```

```

\@glxtr@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\@glxtr@acronymlists\@glsacronymlists
  \let\@glsacronymlists\@empty
  \let\@addtoacronymlists\@gobble
  \let\SetAcronymLists\@gobble
  \let\@glxtr@base@acrcmd\@glxtr@base@acrcmd@warn
  \renewcommand*\newacronym}[4][{}]{%
    \glxtr@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}%
    \@glxtr@org@newacronymstyle{##1}%
  }%
}
\MakeAcronymsAbbreviations
\newcommand*\RestoreAcronyms{%
  \let\@glsacronymlists\@glxtr@acronymlists
  \let\@addtoacronymlists\@glxtr@org@addtoacronymlists
  \let\SetAcronymLists\@glxtr@org@setacronymlists
  \let\@glxtr@base@acrcmd\@gobbletwo
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gls@type}%
  }%
  \SetGenericNewAcronym
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acronymfont}[1]{##1}%
  \let\setacronymstyle\@glxtr@org@setacronymstyle
  \let\newacronymstyle\@glxtr@org@newacronymstyle
  \renewcommand*\@gls@link@checkfirsthyper{%
    \ifglused{\glslabel}%
    {\let\glxtrifwasfirstuse\@secondoftwo}
    {\let\glxtrifwasfirstuse\@firstoftwo}%
    \@glxtr@org@checkfirsthyper
  }

```

```

}
\glsssetcategoryattribute{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\@gls@write{\newwrite\@gls@write}{}%
\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@noprntglossary\@makeglossaries@warn@noprntglossary
\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\@glxtr@mixed@assign@sortkey
\renewcommand*{\@printgloss@setsort}{%
    \expandafter\@glxtr@gettype\expandafter,\@glxtr@printglossopts,%
    type=\glsdefaulttype,\@end@glxtr@gettype
\def\@glo@sorttype{\@glo@default@sorttype}%
}%
\ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
        \for\@gls@type:=\@glxtr@reg@glosslist\do{%
            \ifdefempty{\@gls@type}{\@gls@automake{\@gls@type}}%
        }%
    }%
\fi
\ifdef\@glo@check@sortallowed{\@glo@check@sortallowed\makeglossaries}{}%
\fi
}%
}%
}%
}
\ifdef\@printgloss@checkexists
{\newcommand{\glxtr@printgloss@checkexists}{\@printgloss@checkexists}}
{\newcommand{\glxtr@printgloss@checkexists}[2]{#2}}
\newcommand{\@glxtr@orgprintglossary}[2]{%
    \def\@glo@type{\glsdefaulttype}%
    \def\glossarytitle{%
        \ifcsdef{\@glo@type \@title}%
        {\csuse{\@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@dotoc@title{\gls@settoc@title{\@glo@type}}%
    \let\@org@glossaryentrynumbers\glossaryentrynumbers
    \bgroup
        \@printgloss@setsort
        \setkeys{printgloss}{#1}%
        \ifx\glossarytitle\org@glossarytitle
        \else

```



```

\cslet{@glo@type@{@glo@type @title}{\glossarytitle}%
\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers@glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\glsxtractivatenopost
\gls@dotoc@title
\@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\@glxstr@nopostpunc@postdesc{%
  \@glxstr@org@postdescription
  \let\@glxstr@nopostpunc@postdesc\@empty
  \let\glxstrrestorepostpunc\@empty
}%
}
\newcommand*{\glxstrrestorepostpunc}{}
\renewcommand{\@printglossary}[2]{%
  \def\@glxstr@printglossopts{#1}%
  \@glxstr@orgprintglossary{#1}{#2}%
}
\define@choicekey{printgloss}{target}
[{\@glxstr@printglossval\@glxstr@printglossnr}]%
{true,false}[true]%
{%
  \ifcase\@glxstr@printglossnr
    \def\@glstarget{\@glsdohypertarget}%
  \else
    \let\@glstarget\@secondoftwo
  \fi
}
\newcommand{\@glxstr@hypernameprefix}{}
\define@key{printgloss}{targetnameprefix}{%
  \renewcommand{\@glxstr@hypernameprefix}{#1}%
}
\define@key{printgloss}{prefix}{%
  \renewcommand{\@glolinkprefix}{#1}%
}
\define@key{printgloss}{label}{%
  \glxstrsetglossarylabel{#1}%
}
\newcommand{\glxstrsetglossarylabel}[1]{%
  \renewcommand*{\@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{#1}%
  }%
}
\newcount\@glxstr@leveloffset
\define@key{printgloss}{leveloffset}{%
  \@glxstr@assign@leveloffset#1\relax
}
\newcommand*{\@glxstr@assign@leveloffset}{%
  \@ifnextchar+{\p@glxstr@assign@leveloffset}{\np@glxstr@assign@leveloffset}%
}
\newcommand*{\p@glxstr@assign@leveloffset}[1]{%
  \@ifnextchar+{\pp@glxstr@assign@leveloffset}{\np@glxstr@assign@leveloffset}%
}
\def\np@glxstr@assign@leveloffset#1\relax{\@glxstr@leveloffset=#1\relax}
\def\pp@glxstr@assign@leveloffset#1\relax{\advance\@glxstr@leveloffset by #1\relax}
\define@boolkey{printgloss}[glxstr@printgloss@]{groups}[true]{}

```

```

\glstr@printgloss@groupstrue
\let\@glstr@org@glsdohypertarget\glsdohypertarget
\renewcommand{\glsdohypertarget}[2]{%
  \@glstr@org@glsdohypertarget{\@glstrhypernameprefix#1}{#2}%
}
\ifx\@glstarget\@glstr@org@glsdohypertarget
  \def\@glstarget{\glsdohypertarget}%
\fi
\newcommand{\@glstr@do@org@target}[2]{%
  {%
    \let\glsdohypertarget\@glstr@org@glsdohypertarget
    \@glstarget{#1}{#2}%
  }%
}
\newcommand*{\glstr@makeglossaries}[1]{%
\def\@glstr@gettype#1,type=#2,#3\@end@glstr@gettype{%
  \def\@glo@type{#2}%
}
\newcommand\@glstr@mixed@assign@sortkey[1]{%
  \protected@edef\@glo@type{\@glo@type}%
  \expandafter\DTLifinlist\expandafter{\@glo@type}{\@glstr@reg@glosslist}%
  {%
    \@glo@no@assign@sortkey{#1}%
  }%
  {%
    \@glo@assign@sortkey{#1}%
  }%
}%
\let\@glstr@idx@displaynumberlist\glstdisplaynumberlist
\newcommand*{\@glstr@noidx@displaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glstdetoklabel{#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
  }%
  {%
    \glstrundeftag
    \glsoifexists{#1}%
  }%
}

```

```

    {%
      \GlossariesWarning{Missing location list for ‘#1’. Either
        a rerun is required or you haven’t referenced the entry.}%
    }%
  }%
}%

\newcommand*{\@glxtr@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}%
  }%
  {%
    \glxtrundeftag
    \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*{\@glxtr@noidx@entrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    \glxtrundeftag
    \glsdoifexists{#1}%
    {%
      \GlossariesWarning{Missing location list for ‘#1’. Either
        a rerun is required or you haven’t referenced the entry.}%
    }%
  }%
}%

\newcommand*{\@glxtr@idx@entrynumberlist}[1]{\glsentrynumberlist{#1}}
\renewcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \protected@edef\@glxtr@titlelabel{#1}%
  \ifdefvoid\@glxtr@titlelabel
  {}%
  {%
    \protected@edef\@glxtr@titlelabel{\csuse{glxtr@grouptitle@#1}}%
  }%
}

```

```

}%
\ifdefined{\@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
  }%
}

```

```

\@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{\@gls@ref@{\@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}{\@gls@ref@{\@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@glxtr@display@loc{#4}%
}
\def\@glsxtr@display@loc#1#2\end@glxtr@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*{\glxtrdisplaystartloc}[2]{%
  \protected@edef\glxtrlocrangefmt{#1}%
  \ifx\glxtrlocrangefmt\empty
    \def\glxtrlocrangefmt{glsnumberformat}%
  \fi
  \expandafter\glxtrdisplaysingleloc
  \expandafter{\glxtrlocrangefmt}{#2}%
}
\newcommand*{\glxtrdisplayendloc}[2]{%
  \protected@edef\@glxtr@tmp{#1}%
  \ifdefempty{\@glxtr@tmp}{\def\@glxtr@tmp{glsnumberformat}}{}%
  \ifx\glxtrlocrangefmt\@glxtr@tmp
  \else
    \GlossariesExtraWarning{Mismatched end location range
      (start=\glxtrlocrangefmt, end=\@glxtr@tmp)}%
  \fi
  \expandafter\glxtrdisplayendlochook\expandafter{\@glxtr@tmp}{#2}%
  \expandafter\glxtrdisplaysingleloc
  \expandafter{\glxtrlocrangefmt}{#2}%
  \def\glxtrlocrangefmt{}%
}
\newcommand*{\glxtrdisplayendlochook}[2]{}
\newcommand*{\glxtrlocrangefmt}{}
\renewcommand*{\setentrycounter}[2][{}]{%
  \def\glxtrcounterprefix{#1}%
  \ifx\glxtrcounterprefix\empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsetentrycounter{#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\glxtrlocationhyperlink\expandafter
      \glsetentrycounter\expandafter\@glo@counterprefix\expandafter{\the\toks@}%
    \fi
  \else
    \@gls@ReturnAfterFi{%
      \@gls@removespaces#2\@nil
    }%
  \fi
}
\newcommand*{\glxtrlocationhyperlink}[3]{%

```

```

\ifdefvoid\glstrsupplocationurl
{%
  \GlsXtrInternalLocationHyperlink{#1}{#2}{#3}%
}%
{%
  \hyperref{\glstrsupplocationurl}{#1#2#3}{#3}%
}%
}
\newcommand*{\glstrsupphypernumber}[1]{%
{%
  \glshasattribute{\glscurrententrylabel}{externallocation}%
  {%
    \def\glstrsupplocationurl{%
      \glsgotattribute{\glscurrententrylabel}{externallocation}}%
    }%
    {%
      \def\glstrsupplocationurl{%
        }%
      }%
      \glshypernumber{#1}%
    }%
  }%
}
\renewcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type\@glo@type @in\endcsname}%
  {}%
  {\glstrNoGlossaryWarning{\@glo@type}}%
  \ifglxindy
    \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
  \ifglxindy 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 \ifglxindy 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=@glxtr@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{nomissinggls} package option.%
}
\newcommand*{\@glxtr@defaultnoglossarywarning}[1]{%
  \glossarysection[\glossarytoctitle]{\glossarytitle}
  \GlsXtrNoGlsWarningHead{#1}{\jobname.\csname @glotype@\@glo@type @in\endcsname}
  \par
  \glxtrifemptyglossary{#1}%
  {%
    \GlsXtrNoGlsWarningEmptyStart\space
    \ifthenelse{\equal{#1}{main}}{\GlsXtrNoGlsWarningEmptyMain\par
    \medskip
    \noindent\texttt{\string\usepackage[nomain\ifglsacronym ,acronym\fi]}%
    \glsoopenbrace 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]%
        \glsoopenbrace glossaries-extra\glsclosebrace}
        \medskip
      }%
    }%

    \ifdefequal\makeglossaries\@no@makeglossaries

```

```

        {%
        \GlsXtrNoGlsWarningMisMatch
        }%
        {%
        \GlsXtrNoGlsWarningBuildInfo
        }%
    \fi
}%
{%
    \GlsXtrNoGlsWarningNoOut
    {\jobname.\csname @glotype@ \@glo@type @out\endcsname}%
}%
\par
\GlsXtrNoGlsWarningTail
}
\newcommand*{\@glxtr@record@noglossarywarning}[1]{%
    \GlossariesExtraWarning{\string\printglossary\space doesn't work\MessageBreak
with record=\@glxtr@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*{\glxtrresourcefile}[2][{}]{%
    \disable@keys{glossaries-extra.sty}{record}%
    \glxtr@writefields
    \ifdefempty\GlsXtrDefaultResourceOptions
    {%
        \protected@write\@auxout{\glxtrresourceinit}%
        {\string\glxtr@resource{#1}{#2}}%
    }%
    {%
        \protected@write\@auxout{\glxtrresourceinit}%
        {\string\glxtr@resource{\GlsXtrDefaultResourceOptions,#1}{#2}}%
    }%
    \let\@glxtr@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
        \@glxtr@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\bibgls@hrefchar
{}%
{%
\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\glxtr@langtag{\CurrentTrackedLanguageTag}}%
}%
{}%
\protected@write\@auxout{}\{\string\glxtr@pluralsuffixes
  {\glspluralsuffix}{\abbrvpluralsuffix}{\acrpluralsuffix}%
  {\glxtrabbrvpluralsuffix}}%
\ifdef\inputencodingname
{%
  \protected@write\@auxout{}\{\string\glxtr@texencoding{\inputencodingname}}%
}%
{%
  \@ifpackageloaded{fontspec}%
  {\protected@write\@auxout{}\{\string\glxtr@texencoding{utf8}}}%
  {}%
}%
\protected@write\@auxout{}\{\string\glxtr@shortcutsval{\@glxtr@shortcutsval}}%
\AtBeginDocument
  {\protected@write\@auxout{}\{\string\glxtr@linkprefix{\glolinkprefix}}}%
\let\glxtr@writefields\relax
\ifglautomake
  \IfFileExists{\jobname.aux}%
  {\immediate\write18{\bib2gls \jobname}}{}%
  \ifx\@gls@doautomake\@gls@doautomake@err
    \let\@gls@doautomake\relax
  \fi
\fi
\@glxtr@if@record@only
{\ifdefstring{\glsorder}{letter}%
  {\GlossariesExtraWarningNoLine{Package option ‘order=letter’ isn’t
    supported with ‘record=\@glxtr@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*{\glxtr@record}[5]{%
\newcommand*{\glxtr@record@nameref}[8]{%
\newcommand*{\glxtr@counterrecord}[3]{%
  \glxtrfieldlistgadd{#1}{record.#2}{#3}%

```

```

}
\newcommand*{\@glxtr@counterrecordhook}{%
\newcommand*{\GlsXtrRecordCounter}[1]{%
  \@@glxtr@recordcounter{#1}%
}
\@onlypreamble\GlsXtrRecordCounter
\newcommand*{\@glxtr@docounterrecord}[1]{%
  \protected@write\@auxout{}\string\glxtr@counterrecord
    {\@glx@label}{#1}{\csuse{the#1}}}%
}
\newcommand*{\glxtrglossentry}[1]{%
  \glxtrtitleorpdforheading
    {\@glxtrglossentry{#1}}%
  {\glsentryname{#1}}%
  {\glxtrheadname{#1}}%
}
\newrobustcmd*{\@glxtrglossentry}[1]{%
  \glxtrtitleorpdforheading
    {%
      \glxdoifexists{#1}%
      {%
        \begingroup
          \protected@edef\glscurrententrylabel{\glxdetoklabel{#1}}%
          \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
          \ifglshasparent{#1}%
            {\GlsXtrStandaloneSubEntryItem{#1}}%
            {\glxentryitem{#1}}%
            {\GlsXtrStandaloneEntryName{#1}}%
          \endgroup
        }%
      }%
      {\glsentryname{#1}}%
      {\glxtrheadname{#1}}%
    }
}
\newcommand*{\GlsXtrStandaloneEntryName}[1]{%
  \glxtarget{#1}{\glxglossentryname{#1}}%
}
\newcommand{\GlsXtrStandaloneGlossaryType}{\glxentrytype{\glscurrententrylabel}}
\newcommand*{\GlsXtrStandaloneSubEntryItem}[1]{%
  \GlsXtrIfFieldEqNum{level}{#1}{1}{\glxsubentryitem{#1}}}%
}
\newcommand*{\glxtrglossentryother}[3]{%
  \ifstrempy{#1}%
  {%
    \ifcsdef{glxtrhead#3}%
    {%
      \glxtrtitleorpdforheading
        {\@glxtrglossentryother{#2}{#3}{#1}}%
        {\@glx@entry@field{#2}{#3}}%
        {\csuse{glxtrhead#3}{#2}}%
    }
  }
}

```

```

}%
{%
  \glstrtitleorpdforheading
  {\@glstrglossentryother{#2}{#3}{#1}}%
  {\@gls@entry@field{#2}{#3}}%
  {\@gls@entry@field{\NoCaseChange{#2}}{#3}}%
}%
}%
{%
  \glstrtitleorpdforheading
  {\@glstrglossentryother{#2}{#3}{#1}}%
  {\@gls@entry@field{#2}{#3}}%
  {#1}%
}%
}
\newrobustcmd*{\@glstrglossentryother}[3]{%
  \glstrtitleorpdforheading
  {%
    \glstoifexists{#1}%
    {%
      \begingroup
        \protected@edef\glscurrententrylabel{\glsetoklabel{#1}}%
        \protected@edef\currentglossary{\GlsXtrStandaloneGlossaryType}%
        \ifglshasparent{#1}%
          {\GlsXtrStandaloneSubEntryItem{#1}}%
          {\glentryitem{#1}}%
          \GlsXtrStandaloneEntryOther{#1}%
        \endgroup
      }%
    }%
    {\@gls@entry@field{#1}{#2}}%
    {#3}%
  }
}
\newcommand*{\GlsXtrStandaloneEntryOther}[2]{%
  \glstarget{#1}{\glossentrynameother{#1}{#2}}%
}
\ifdefined\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=\glstdefaulttype,#1}{\@print@unsrt@glossary}%
}

```



```

}
\newcommand*{\s@printunsrtglossary}[2][{}]{%
  \begin{group}
    #2%
    \@printglossary{type=\glsdefaulttype,#1}{\@print@unsrt@glossary}%
  \end{group}
}
\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 glist@\@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\@glxstr@doglossary{\end{theglossary}}%
  \printunsrtglossarypredoglossary
  \@glxstr@doglossary
}%
\glossarypostamble
}
\newcommand*{\printunsrtinnerglossary}[3][]{%
  \begingroup
  \def\@glxstr@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\@glxstr@printglossopts{#1}%
    \def\@glo@type{\glsdefaulttype}%
    \def\glossarytitle{\csname @glo@type\@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@dotoc@title{\glssettoctitle{\@glo@type}}%
    \let\@org@glossaryentrynumbers\glossaryentrynumbers
    \@printgloss@setsort
    \setkeys{printgloss}{#1}%
    \ifglossaryexists*\@glo@type}%
  {%
    \ifx\glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type\@glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary\@glo@type
  }%
}%

```

```

\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\let\nopostdesc\@nopostdesc
\gls@dotocitle
\@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
    }
  }
}

```

```

    {%
      \ifglstr@printgloss@groups
      \ifglshasparent{\glscurrententrylabel}{}%
      {%
        \@glstr@checkgroup\glscurrententrylabel
        \expandafter\appto\expandafter\@glstr@doglossary\expandafter
        {\@glstr@groupheading}%
      }%
    \fi
    \protected@eappto\@glstr@doglossary{%
      \noexpand\@printunsrt@glossary@handler{\glscurrententrylabel}}%
  }%
}%
\printunsrtglossarypredoglossary
\@glstr@doglossary
}%
}
\newcommand*{\printunsrtglossaryentryprocesshook}[1]{}
\newcommand*{\printunsrtglossaryskipentry}{%
  \PackageError{glossaries-extra}{\string\printunsrtglossaryskipentry\space
can only be used within \string\printunsrtglossaryentryprocesshook}{}%
}
\newcommand*{\@glstr@printunsrtglossaryskipentry}{%
  \let\glstr@process\@gobble
}
\newcommand*{\printunsrtglossarypredoglossary}{}
\newcommand{\@printunsrt@glossary@handler}[1]{%
  \protected@xdef\glscurrententrylabel{#1}%
  \printunsrtglossaryhandler\glscurrententrylabel
}
\newcommand{\printunsrtglossaryhandler}[1]{%
  \glstrunsrtdo{#1}%
}
\newrobustcmd*{\glstriflabelinlist}[4]{%
  \protected@edef\@glstr@doiflabelinlist{\noexpand\@gl@ifinlist{#1}{#2}}%
  \@glstr@doiflabelinlist{#3}{#4}%
}
\newcommand{\print@op@unsrtglossaryunit}[2][1]{%
  \s@printunsrtglossary[type=\gldefaulttype,#1]{%
    \printunsrtglossaryunitsetup{#2}%
  }%
}
\newcommand*{\printunsrtglossaryunitsetup}[1]{%
  \renewcommand{\printunsrtglossaryhandler}[1]{%
    \glstrfieldxifinlist{##1}{record.#1}{\csuse{the#1}}
    {\glstrunsrtdo{##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{\glspare\medskip\glspare}%
}
\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{\gls@groupskip}}%
\protected@eappto\@glsxtr@groupheading{%
\noexpand\gls@groupheading{\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}%
      }%
    }%
  }%
  {%
    \glossentry{#1}%
  }%
}

```

```

\newcount\@glstrnewgls@inner
\newcommand*{\@glstr@providenewgls}{%
  \protected@write\@auxout{}\string\providecommand{\string\@glstr@newglslike}[2]{}%
  \let\@glstr@providenewgls\relax
}
\newcommand{\glstridentifyglslike}[2]{%
  \ifdefequal\@glstr@record@setting\@glstr@record@setting@off
  {}%
  {%
    \@glstr@providenewgls
    \protected@write\@auxout{}\string\@glstr@newglslike{#1}{\string#2}}%
  }%
}
\newcommand*{\@glstrnewgls}[4]{%
  \ifdef{#3}%
  {%
    \PackageError{glossaries-extra}{Command \string#3\space already
defined}{}%
  }%
  {%
    \glstridentifyglslike{#2}{#3}%
    \ifcsdef{#4like@#2}%
    {%
      \advance\@glstrnewgls@inner by \@ne
      \def\@glstrnewgls@innercsname{#4like\number\@glstrnewgls@inner @#2}%
    }%
    {\def\@glstrnewgls@innercsname{#4like@#2}}%
    \expandafter\newrobustcmd\expandafter*\expandafter
      #3\expandafter{\expandafter\@glshyp@opt\csname\@glstrnewgls@innercsname\endcsname}%
    \ifstrempy{#1}%
    {%
      \expandafter\newcommand\expandafter*\csname\@glstrnewgls@innercsname\endcsname[2][{%
        \new@ifnextchar[%
          {\csname @#4@\endcsname{##1}{#2##2}}%
          {\csname @#4@\endcsname{##1}{#2##2}[]}%
        }%
      }%
    }%
    {%
      \expandafter\newcommand\expandafter*\csname\@glstrnewgls@innercsname\endcsname[2][{%
        \new@ifnextchar[%
          {\csname @#4@\endcsname{#1,##1}{#2##2}}%
          {\csname @#4@\endcsname{#1,##1}{#2##2}[]}%
        }%
      }%
    }%
  }%
}
\newrobustcmd*{\glstrnewgls}[3][{%
  \@glstrnewgls{#1}{#2}{#3}{gls}%
}
\newrobustcmd*{\glstrnewglslike}[6][{%

```

```

\@glstrnewgls{#1}{#2}{#3}{gls}%
\@glstrnewgls{#1}{#2}{#4}{glspl}%
\@glstrnewgls{#1}{#2}{#5}{Gls}%
\@glstrnewgls{#1}{#2}{#6}{Glspl}%
}
\newrobustcmd*\glstrnewGLSlike}[4][]{%
\@glstrnewgls{#1}{#2}{#3}{GLS}%
\@glstrnewgls{#1}{#2}{#4}{GLSpl}%
}
\newrobustcmd*\glstrnewrgls}[3][]{%
\@glstrnewgls{#1}{#2}{#3}{rgls}%
}
\newrobustcmd*\glstrnewrglslike}[6][]{%
\@glstrnewgls{#1}{#2}{#3}{rgls}%
\@glstrnewgls{#1}{#2}{#4}{rglspl}%
\@glstrnewgls{#1}{#2}{#5}{rGls}%
\@glstrnewgls{#1}{#2}{#6}{rGlspl}%
}
\newrobustcmd*\glstrnewrGLSlike}[4][]{%
\@glstrnewgls{#1}{#2}{#3}{rGLS}%
\@glstrnewgls{#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]{%
  \glsattribute{#1}{recordcount}%
  {%
    \ifnum\glsxtrrecordtriggervalue{#1}>\glsattribute{#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\theglentrycounter\@glsxtr@thevalue
    \def\theHglentrycounter{\@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]{%
    \glstriferecordtrigger{#2}%
    {%
        \@glstr@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]{%
    \glstriferecordtrigger{#2}%
    {%
        \@glstr@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]{%
    \glstriferecordtrigger{#2}%
    {%
        \@glstr@rglstrigger@record{#1}{#2}{\rGlsformat{#2}{#3}}%
    }%
    {%
        \@rGls@{#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][1]{%
  \let\glsxtr@inc@linkcount\@glsxtr@do@inc@linkcount
  \@for\@glsxtr@label:=#2\do
  {%
    \glssetcategoryattribute{\@glsxtr@label}{linkcount}{true}%
    \ifstrempy{#1}{}%
    {%
      \ifcsundef{c@#1}%

```

```

        {\@nocounterr{#1}}%
        {\glissetcategoryattribute{\@glxtr@label}{linkcountmaster}{#1}}%
    }%
}%
}
\@onlypreamble\GlsXtrEnableLinkCounting
\@ifpackageloaded{glossaries-accsupp}
{
    \newcommand*{\glsaccessname}[1]{%
        \glsnameaccessdisplay
        {%
            \glentryname{#1}%
        }%
        {#1}%
    }
    \newcommand*{\Glsaccessname}[1]{%
        \glsnameaccessdisplay
        {%
            \Glsentryname{#1}%
        }%
        {#1}%
    }
    \newcommand*{\GLSaccessname}[1]{%
        \glsnameaccessdisplay
        {%
            \mfirstucMakeUppercase{\glentryname{#1}}%
        }%
        {#1}%
    }
    \newcommand*{\glsaccesstext}[1]{%
        \glstextaccessdisplay
        {%
            \glentrytext{#1}%
        }%
        {#1}%
    }
    \newcommand*{\Glsaccesstext}[1]{%
        \glstextaccessdisplay
        {%
            \Glsentrytext{#1}%
        }%
        {#1}%
    }
    \newcommand*{\GLSaccesstext}[1]{%
        \glstextaccessdisplay
        {%
            \mfirstucMakeUppercase{\glentrytext{#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]{%
  \glssymbolpluralaccessdisplay

```

```

    {%
      \mfirstucMakeUppercase{\glentrysymbolplural{#1}}%
    }%
    {#1}%
  }
  \newcommand*{\glsaccessdesc}[1]{%
    \glsdescriptionaccessdisplay
    {%
      \glentrydesc{#1}%
    }%
    {#1}%
  }
  \newcommand*{\Glsaccessdesc}[1]{%
    \glsdescriptionaccessdisplay
    {%
      \Glsentrydesc{#1}%
    }%
    {#1}%
  }
  \newcommand*{\GLSaccessdesc}[1]{%
    \glsdescriptionaccessdisplay
    {%
      \mfirstucMakeUppercase{\glentrydesc{#1}}%
    }%
    {#1}%
  }
  \newcommand*{\glsaccessdescplural}[1]{%
    \glsdescriptionpluralaccessdisplay
    {%
      \glentrydescplural{#1}%
    }%
    {#1}%
  }
  \newcommand*{\Glsaccessdescplural}[1]{%
    \glsdescriptionpluralaccessdisplay
    {%
      \Glsentrydescplural{#1}%
    }%
    {#1}%
  }
  \newcommand*{\GLSaccessdescplural}[1]{%
    \glsdescriptionpluralaccessdisplay
    {%
      \mfirstucMakeUppercase{\glentrydescplural{#1}}%
    }%
    {#1}%
  }
  \newcommand*{\glsaccessshort}[1]{%
    \glsshortaccessdisplay
    {%

```



```

        \glentryshort{#1}%
    }%
    {#1}%
}
\newcommand*\Glsaccessshort}[1]{%
    \glsshortaccessdisplay
    {%
        \glentryshort{#1}%
    }%
    {#1}%
}
\newcommand*\Glsaccessshort}[1]{%
    \glsshortaccessdisplay
    {%
        \mfirstucMakeUppercase{\glentryshort{#1}}%
    }%
    {#1}%
}
\newcommand*\glssaccessshortpl}[1]{%
    \glsshortpluralaccessdisplay
    {%
        \glentryshortpl{#1}%
    }%
    {#1}%
}
\newcommand*\Glsaccessshortpl}[1]{%
    \glsshortpluralaccessdisplay
    {%
        \glentryshortpl{#1}%
    }%
    {#1}%
}
\newcommand*\Glsaccessshortpl}[1]{%
    \glsshortpluralaccessdisplay
    {%
        \mfirstucMakeUppercase{\glentryshortpl{#1}}%
    }%
    {#1}%
}
\newcommand*\glssaccesslong}[1]{%
    \glslongaccessdisplay{\glentrylong{#1}}{#1}%
}
\newcommand*\Glsaccesslong}[1]{%
    \glslongaccessdisplay{\Glsentrylong{#1}}{#1}%
}
\newcommand*\Glsaccesslong}[1]{%
    \glslongaccessdisplay
    {%
        \mfirstucMakeUppercase{\glentrylong{#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
  {%

```

```

\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*{\glxtrprovideaccsuppcmd}[2]{%
  \ifcsundef{glxtr#1#2accsup}%
  {\csdef{glxtr#1#2accsup}{\glsshortaccsup}}%
  {}%
}
\newcommand*{\glxtrAccSuppAbbrSetNoLongAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glssetcategoryattribute{#1}{firstshortaccess}{true}%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#1}{name}%
  \glxtrprovideaccsuppcmd{#1}{first}%
  \glxtrprovideaccsuppcmd{#1}{firstpl}%
  \glxtrprovideaccsuppcmd{#1}{text}%
  \glxtrprovideaccsuppcmd{#1}{plural}%
}
\newcommand*{\glxtrAccSuppAbbrSetFirstLongAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#1}{name}%
  \glxtrprovideaccsuppcmd{#1}{text}%
  \glxtrprovideaccsuppcmd{#1}{plural}%
}
\newcommand*{\glxtrAccSuppAbbrSetTextShortAttrs}[1]{%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#1}{text}%
  \glxtrprovideaccsuppcmd{#1}{plural}%
}
\newcommand*{\glxtrAccSuppAbbrSetNameShortAttrs}[1]{%
  \glssetcategoryattribute{#1}{nameshortaccess}{true}%
  \glxtrprovideaccsuppcmd{#1}{name}%
}
\newcommand*{\glxtrAccSuppAbbrSetNameLongAttrs}[1]{%
  \glssetcategoryattribute{#1}{firstshortaccess}{true}%
  \glssetcategoryattribute{#1}{textshortaccess}{true}%

```

```

\glxtrprovideaccsuppcmd{#1}{first}%
\glxtrprovideaccsuppcmd{#1}{firstpl}%
\glxtrprovideaccsuppcmd{#1}{text}%
\glxtrprovideaccsuppcmd{#1}{plural}%
}
}
{
\newcommand*{\glsaccessname}[1]{\glsentryname{#1}}
\newcommand*{\Glsaccessname}[1]{\Glsentryname{#1}}
\newcommand*{\GLSaccessname}[1]{%
\protect\mfirstucMakeUppercase{\glsentryname{#1}}}
\newcommand*{\glsaccessstext}[1]{\glsentrytext{#1}}
\newcommand*{\Glsaccessstext}[1]{\Glsentrytext{#1}}
\newcommand*{\GLSaccessstext}[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{\glentryshortpl{#1}}
\newcommand*{\glsaccesslong}[1]{\glentrylong{#1}}
\newcommand*{\Glsaccesslong}[1]{\Glsentrylong{#1}}
\newcommand*{\GLSaccesslong}[1]{%
\protect\mfirstucMakeUppercase{\glentrylong{#1}}}
\newcommand*{\glsaccesslongpl}[1]{\glentrylongpl{#1}}
\newcommand*{\Glsaccesslongpl}[1]{\Glsentrylongpl{#1}}
\newcommand*{\GLSaccesslongpl}[1]{%
\protect\mfirstucMakeUppercase{\glentrylongpl{#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@#1@#2@#3}{#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*{\glschascategoryattribute}[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]{%
  \ifglstryexists{#1}%
    {\glshascategoryattribute{\glscategory{#1}}{#2}{#3}{#4}}%
    {#4}%
}
\newcommand{\glusifcategoryattribute}[5]{%
  \ifcsundef{@glstr@categoryattr@#1@#2}%
    {#5}%
    {\ifcsstring{@glstr@categoryattr@#1@#2}{#3}{#4}{#5}}%
}
\newcommand{\glusifattribute}[5]{%
  \ifglstryexists{#1}%
    {\glusifcategoryattribute{\glscategory{#1}}{#2}{#3}{#4}{#5}}%
    {#5}%
}
\glissetcategoryattribute{general}{regular}{true}
\glissetcategoryattribute{acronym}{regular}{true}
\newcommand*{\glissetregularcategory}[1]{%
  \glissetcategoryattribute{#1}{regular}{true}%
}
\newcommand{\glusifregularcategory}[3]{%
  \glusifcategoryattribute{#1}{regular}{true}{#2}{#3}%
}
\newcommand{\glusifnotregularcategory}[3]{%
  \glusifcategoryattribute{#1}{regular}{false}{#2}{#3}%
}
\newcommand{\glusifregular}[3]{%
  \glusifregularcategory{\glscategory{#1}}{#2}{#3}%
}
\newcommand{\glusifnotregular}[3]{%
  \glusifnotregularcategory{\glscategory{#1}}{#2}{#3}%
}
\newcommand{\glsforeachincategory}[5][\@glo@types]{%
  \forallglossaries[#1]{#3}%
  {%
    \forglstryes[#3]{#4}%
    {%
      \glusifcategory{#4}{#2}{#5}{}%
    }%
  }%
}
\newcommand{\glsforeachwithattribute}[6][\@glo@types]{%
  \forallglossaries[#1]{#4}%
  {%
    \forglstryes[#4]{#5}%
    {%
      \glusifattribute{#5}{#2}{#3}{#6}{}%
    }%
  }%
}

```

```

}
\ifdef\newterm
{%
  \renewcommand*{\newterm}[2] [] {%
    \newglossaryentry{#2}%
      {type={index},category=index,name={#2},%
        description={\glstrpostdescription\nopostdesc},#1}%
  }
  \glsssetcategoryattribute{index}{regular}{true}
  \newcommand*{\glstrpostdescindex}{}
}
{}
\ifdef\printsymbols
{%
  \newcommand*{\glstrnewsymbol}[3] [] {%
    \newglossaryentry{#2}{name={#3},sort={#2},type=symbols,category=symbol,#1}%
  }
  \glsssetcategoryattribute{symbol}{regular}{true}
  \newcommand*{\glstrpostdescsymbol}{}
}
{}
\ifdef\printnumbers
{%
\ifdef\printnumbers
  \newcommand*{\glstrnewnumber}[3] [] {%
    \newglossaryentry{#2}{name={#3},sort={#2},type=numbers,category=number,#1}%
  }
  \glsssetcategoryattribute{number}{regular}{true}
  \newcommand*{\glstrpostdescnumber}{}
}
{}
\newcommand*{\glstrsetcategory}[2] {%
  \@for\@glstr@label:=#1\do
  {%
    \glsfieldxdef{\@glstr@label}{category}{#2}%
  }%
}
\newcommand*{\glstrsetcategoryforall}[2] {%
  \forallglossaries[#1]{\@glstr@type}{%
    \forglsentries[\@glstr@type]{\@glstr@label}%
    {%
      \glsfieldxdef{\@glstr@label}{category}{#2}%
    }%
  }%
}
\newcommand*{\glstrfieldtitlecase}[2] {%
  \expandafter\glstrfieldtitlecasesecs\expandafter
    {\cename glo@\glsdetoklabel{#1}@#2\endcename}%
}
\ifdef\glscapitalisewords

```

```

{
  \newcommand*{\glxtrfieldtitlecasecs}[1]{%
    \expandafter\glscapitalisewords\expandafter{#1}}
}
{
  \newcommand*{\glxtrfieldtitlecasecs}[1]{\xcapitalisewords{#1}}
}
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\glossentrydesc}[1]{%
    \glsoifexistsorwarn{#1}%
    {%
      \glissetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossdescfont}%
      {%
        \protected@edef\@glxtr@attrval{\glsggetattribute{#1}{glossdescfont}}%
        \ifcsdef{\@glxtr@attrval}%
        {%
          \letcs{\@glxtr@glossdescfont}{\@glxtr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
            '\@glxtr@attrval' supplied in glossdescfont attribute
            for entry '#1'. Ignoring}%
          \let\@glxtr@glossdescfont\@firstofone
        }%
      }%
      {\let\@glxtr@glossdescfont\@firstofone}%
      \glusifattribute{#1}{glossdesc}{firstuc}%
      {%
        \@glxtr@glossdescfont{\Glsaccessdesc{#1}}%
      }%
      {%
        \glusifattribute{#1}{glossdesc}{title}%
        {%
          \@glxtr@do@titlecaps@warn
          \glsdescriptionaccessdisplay
          {%
            \@glxtr@glossdescfont{\glxtrfieldtitlecase{#1}{desc}}%
          }%
          {#1}%
        }%
        {%
          \@glxtr@glossdescfont{\Glsaccessdesc{#1}}%
        }%
      }%
    }%
  }
}
{

```

```

\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsssetabbrvfmt{\glscategory{#1}}%
    \glshasattribute{#1}{glossdescfont}%
    {%
      \protected@edef\@glxstr@attrval{\glsggetattribute{#1}{glossdescfont}}%
      \ifcsdef{\@glxstr@attrval}%
      {%
        \letcs{\@glxstr@glossdescfont}{\@glxstr@attrval}%
      }%
      {%
        \GlossariesExtraWarning{Unknown control sequence name
          '\@glxstr@attrval' supplied in glossdescfont attribute
          for entry '#1'. Ignoring}%
        \let\@glxstr@glossdescfont\@firstofone
      }%
    }%
    {\let\@glxstr@glossdescfont\@firstofone}%
    \glusifattribute{#1}{glossdesc}{firstuc}%
    {%
      \@glxstr@glossdescfont{\Glsentrydesc{#1}}%
    }%
    {%
      \glusifattribute{#1}{glossdesc}{title}%
      {%
        \@glxstr@do@titlecaps@warn
        \@glxstr@glossdescfont{\glxstrfieldtitlecase{#1}{desc}}%
      }%
      {%
        \@glxstr@glossdescfont{\glsentrydesc{#1}}%
      }%
    }%
  }%
}
}
\@ifpackageloaded{glossaries-accsupp}
{
  \renewcommand*{\glossentryname}[1]{%
    \@glsdoifexistsorwarn{#1}%
    {%
      \glsssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glxstr@attrval{\glsggetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glxstr@attrval}%
        {%
          \letcs{\@glxstr@glossnamefont}{\@glxstr@attrval}%
        }%
        {%

```

```

\GlossariesExtraWarning{Unknown control sequence name
'\@glstr@attrval' supplied in glossnamefont attribute
for entry '#1'. Reverting to default \string\glstr}%
\let\@glstr@glossnamefont\glstr
}%
}%
{\let\@glstr@glossnamefont\glstr}%
\glstrattribute{#1}{glossname}{firstuc}%
{%
\glstraccessdisplay
{%
\@glstr@glossnamefont{\glstrentryname{#1}}%
}%
{#1}%
}%
{%
\glstrattribute{#1}{glossname}{title}%
{%
\@glstr@do@titlecaps@warn
\glstraccessdisplay
{%
\@glstr@glossnamefont{\glstrfieldtitlecase{#1}{name}}%
}%
{#1}%
}%
{%
\glstrattribute{#1}{glossname}{uc}%
{%
\glstraccessdisplay
{%
\letcs{\glo@name}{glo@\glstrdetoklabel{#1}@name}%
\@glstr@glossnamefont{\mfirstucMakeUppercase{\glo@name}}%
}%
{#1}%
}%
}%
{\letcs{\glo@name}{glo@\glstrdetoklabel{#1}@name}%
\glstraccessdisplay
{%
\expandafter\@glstr@glossnamefont\expandafter{\glo@name}%
}%
{#1}%
}%
}%
\glstrpostnamehook{#1}%
}%
}
}
{

```

```

\renewcommand*{\glossentryname}[1]{%
  \@glsdoifexistsorwarn{#1}%
  {%
    \glssetabbrvfmt{\glscategory{#1}}%
    \glsasattribute{#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}%
    {%

```

```

\glsssetabbrvfmt{\glscategory{#1}}%
\glshasattribute{#1}{glossnamefont}%
{%
  \protected@edef\@glxstr@attrval{\glsggetattribute{#1}{glossnamefont}}%
  \ifcsdef{\@glxstr@attrval}%
  {%
    \letcs{\@glxstr@glossnamefont}{\@glxstr@attrval}%
  }%
  {%
    \GlossariesExtraWarning{Unknown control sequence name
      '\@glxstr@attrval' supplied in glossnamefont attribute
      for entry '#1'. Reverting to default \string\glsnamefont}%
    \let\@glxstr@glossnamefont\glsnamefont
  }%
}%
{\let\@glxstr@glossnamefont\glsnamefont}%
\glsnameaccessdisplay
{%
  \@glxstr@glossnamefont{\Glsentryname{#1}}%
}%
{#1}%
\glxstrpostnamehook{#1}%
}%
}
}
{
  \renewcommand*{\Glossentryname}[1]{%
    \@glsdoidexistsorwarn{#1}%
    {%
      \glsssetabbrvfmt{\glscategory{#1}}%
      \glshasattribute{#1}{glossnamefont}%
      {%
        \protected@edef\@glxstr@attrval{\glsggetattribute{#1}{glossnamefont}}%
        \ifcsdef{\@glxstr@attrval}%
        {%
          \letcs{\@glxstr@glossnamefont}{\@glxstr@attrval}%
        }%
        {%
          \GlossariesExtraWarning{Unknown control sequence name
            '\@glxstr@attrval' supplied in glossnamefont attribute
            for entry '#1'. Reverting to default \string\glsnamefont}%
          \let\@glxstr@glossnamefont\glsnamefont
        }%
      }%
      {\let\@glxstr@glossnamefont\glsnamefont}%
      \@glxstr@glossnamefont{\Glsentryname{#1}}%
      \glxstrpostnamehook{#1}%
    }%
  }
}
}

```

```

\newcommand*{\glxtrpostnamehook}[1]{%
  \let\@glsnnumberformat\@glxtr@defaultnumberformat
  \glxtrdoautoindexname{#1}{indexname}%
  \glsextrapostnamehook{#1}%
  \csuse{glxtrpostname\glscategory{#1}}%
}
\newcommand*{\glsextrapostnamehook}[1]{}%
\newcommand*{\glstdefpostname}[2]{%
  \csdef{glxtrpostname#1}{#2}%
}
\@ifpackageloaded{glossaries-accsupp}
{
  \newcommand*{\glxtr@setaccessdisplay}[1]{%
    \ifcsdef{gl#1accessdisplay}%
    {\letcs\@glxtr@accessdisplay{gl#1accessdisplay}}%
    {%
      \protected@edef\@glsthisval{#1}%
      \@for\@glstmap:=\@glstkeymap\do{%
        \protected@edef\@thiskey{\expandafter\@secondoftwo\@glstmap}%
        \ifdefequal{\@thiskey}{\@glstthisval}%
        {%
          \protected@edef\@glstthisval{\expandafter\@firstoftwo\@glstmap}%
          \@endfortrue
        }%
      }%
    }%
    \ifcsdef{glst\@glstthisval accessdisplay}%
    {\letcs\@glxtr@accessdisplay{glst\@glstthisval accessdisplay}}%
    {\let\@glxtr@accessdisplay\@firstoftwo}%
  }%
}
}
{%
  \newcommand*{\glxtr@setaccessdisplay}[1]{%
    \let\@glxtr@accessdisplay\@firstoftwo
  }
}
\newrobustcmd*{\glossentrynameother}[2]{%
  \@glstdoifexistsorwarn{#1}%
  {%
    \glxtr@setaccessdisplay{#2}%
    \glstsetabbrvfmt{\glscategory{#1}}%
    \glsthasattribute{#1}{glossnamefont}%
    {%
      \protected@edef\@glxtr@attrval{\glstgetattribute{#1}{glossnamefont}}%
      \ifcsdef{\@glxtr@attrval}%
      {%
        \letcs{\@glxtr@glossnamefont}{\@glxtr@attrval}%
      }%
    }%
    {%
      \GlossariesExtraWarning{Unknown control sequence name

```



```

        '\@glxtr@attrval' supplied in glossnamefont attribute
        for entry '#1'. Reverting to default \string\glsnamefont}%
        \let\@glxtr@glossnamefont\glsnamefont
    }%
}%
{\let\@glxtr@glossnamefont\glsnamefont}%
\glsifattribute{#1}{glossname}{firstuc}%
{%
    \@glxtr@accessdisplay
    {\@glxtr@glossnamefont{\@Gls@entry@field{#1}{#2}}}%
    {#1}%
}%
{%
    \glsifattribute{#1}{glossname}{title}%
    {%
        \@glxtr@do@titlecaps@warn
        \@glxtr@accessdisplay
        {\@glxtr@glossnamefont{\glxtrfieldtitlecase{#1}{#2}}}%
        {#1}%
    }%
    {%
        \glsifattribute{#1}{glossname}{uc}%
        {%
            \letcs{\glo@name}{glo@\glsdetoklabel{#1}@#2}%
            \@glxtr@accessdisplay
            {\@glxtr@glossnamefont{\mfirstucMakeUppercase{\glo@name}}}%
            {#1}%
        }%
        {%
            \letcs{\glo@name}{glo@\glsdetoklabel{#1}@#2}%
            \@glxtr@accessdisplay
            {\expandafter\@glxtr@glossnamefont\expandafter{\glo@name}}%
            {#1}%
        }%
    }%
}%
\glxtrpostnamehook{#1}%
}%
}
\newif\ifglxtr@format@override
\glxtr@format@overridefalse
\@ifpackageloaded{hyperref}
{
    \ifHy@hyperindex
        \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
            \@glxtr@format@overridetrue
            \appto\theindex{\let\glshypernumber\@firstofone}%
        }
    \else
        \newcommand*{\GlsXtrEnableIndexFormatOverride}{%

```

```

        \@glxstr@format@overridetrue
        \appto\theindex{\let\glshypernumber\hyperpage}%
    }
\fi
}
{
    \newcommand*{\GlsXtrEnableIndexFormatOverride}{%
        \@glxstr@format@overridetrue
    }
}
\@onlypreamble\GlsXtrEnableIndexFormatOverride
\newcommand*{\glxstrdoautoindexname}[2]{%
    \glshasattribute{#1}{#2}%
    {%
        \@glxstr@autoindex@setname{#1}%
        \protected@edef\@glxstr@attrval{\glsggetattribute{#1}{#2}}%
        \if@glxstr@format@override
            \ifx\@glsnumberformat\@glxstr@defaultnumberformat
            \else
                \let\@glxstr@attrval\@glsnumberformat
            \fi
        \fi
        \ifdefstring{\@glxstr@attrval}{true}%
        {}%
        {\protected@eappto\@glo@name{\@glxstr@autoindex@encap\@glxstr@attrval}}%
        \expandafter\glxstrautoindex\expandafter{\@glo@name}%
    }%
    {}%
}
\newcommand*{\glxstrautoindex}{\index}
\newcommand*{\glxstrautoindexesc}{%
    \@glsc@checkmkidxchars\@glo@sort
    \@glxstr@autoindex@doextra@esc\@glo@sort
}
\newcommand*{\@glxstr@autoindex@setname}[1]{%
    \protected@edef\@glo@name{\glxstrautoindexentry{#1}}%
    \glxstrautoindexassignsort{\@glo@sort}{#1}%
    \glxstrautoindexesc
    \epreto\@glo@name{\@glo@sort\@glxstr@autoindex@at}%
}
\newcommand*{\glxstrautoindexentry}[1]{\string\glsentryname{#1}}
\newcommand*{\glxstrautoindexassignsort}[2]{%
    \glslentryfield{#1}{#2}{sort}%
}
\newcommand*{\@glxstr@autoindex@doextra@esc}[1]{%
    \ifx\@glxstr@autoindex@esc\@glsc@quotearch
    \else
        \def\@glsc@checkedmkidx{%
            \edef\@glxstr@checkspch{%
                \noexpand\@glxstr@autoindex@escquote\expandonce{#1}%
            }
        }
    \fi
}

```

```

        \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##2##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}%
    \fi
  \fi
}

```

```

\def\@glxtr@checkspch{#2#5#1\@nnil#1\@glxtr@endescspch}%
\fi
\fi
\@glxtr@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]{\glossentrysymbol{#1}}
\newrobustcmd*{\@glossentrysymbol}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\begingroup
\glssetabbrvfmt{\glscategory{#1}}%
\glshasattribute{#1}{glosssymbolfont}%
{%
\protected@edef\@glxtr@attrval{\glsggetattribute{#1}{glosssymbolfont}}%
\ifcsdef{\@glxtr@attrval}%
{%
\letcs{\@glxtr@glosssymbolfont}{\@glxtr@attrval}%
}%
{%
\GlossariesExtraWarning{Unknown control sequence name
'\@glxtr@attrval' supplied in glosssymbolfont attribute
for entry '#1'. Ignoring}%
\let\@glxtr@glosssymbolfont\@firstofone
}%
}%
{\let\@glxtr@glosssymbolfont\@firstofone}%
\@glxtr@glosssymbolfont{\glsaccesssymbol{#1}}%
\endgroup
}%
}
\renewcommand*{\Glossentrysymbol}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glssetabbrvfmt{\glscategory{#1}}%

```

```

\Glsaccesssymbol{#1}%
}%
}
\newcommand*\GlsXtrEnableInitialTagging}{%
\ifstar\s@glstr@enabletagging\@glstr@enabletagging
}
\@onlypreamble\GlsXtrEnableInitialTagging
\newcommand*\s@glstr@enabletagging}[2]{%
\undef#2%
\@glstr@enabletagging{#1}{#2}%
}
\newcommand*\@glstr@enabletagging}[2]{%
\@for\@glstr@cat:=#1\do
{%
\ifdefempty\@glstr@cat
}%
\glsssetcategoryattribute{\@glstr@cat}{tagging}{true}}%
}%
\newrobustcmd*#2[1]{##1}%
\def\@glstr@taggingscs{#2}%
\renewcommand*\@glstr@activate@initialtagging{%
\let#2\@glstr@tag
}%
\ifundef\@glspreglossaryhook
{\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*\@glstr@do@titlecaps@warn}{%
\GlossariesExtraWarning{mfirstuc.sty too old. Title Caps
support not available}%
\let\@glstr@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*{\@glxstr@do@titlecaps@warn}{}
\newcommand*\@glxstr@activate@initialtagging{}
\newrobustcmd*{\@glxstr@tag}[1]{%
    \glsifattribute{\glscurrententrylabel}{tagging}{true}%
    {\glxstrtagfont{#1}}{#1}%
}
\newcommand*{\glxstrtagfont}[1]{\underline{#1}}
\ifdef\@gls@preglossaryhook
{
    \renewcommand*{\@gls@preglossaryhook}{%
        \@glxstr@activate@initialtagging
        \ifundef\@glxstr@org@postdescription
        {%
            \let\@glxstr@org@postdescription\glspostdescription
            \renewcommand*{\glspostdescription}{%
                \ifglsentryexists{\glscurrententrylabel}%
                {%
                    \glxstrpostdescription
                    \@glxstr@org@postdescription
                }%
            }%
        }%
    }%
}
\glossxtrsetpopts
}%
}
{}
\newcommand*{\glxstrpostdescription}{%
    \csuse{glxstrpostdesc\glscategory{\glscurrententrylabel}}%
}
\newcommand*{\glxstrpostdescgeneral}{}
\newcommand*{\glxstrpostdescterm}{}
\newcommand*{\glxstrpostdescacronym}{}
\newcommand*{\glxstrpostdescabbreviation}{}
\newcommand*{\glsdefpostdesc}[2]{%
    \csdef{glxstrpostdesc#1}{#2}%
}
\renewcommand*{\glspostlinkhook}{%
    \ifglsentryexists{\glslabel}{\glxstrpostlinkhook}{}%
}
\newcommand*{\glxstrpostlinkhook}{%
    \glxstrdiscardperiod{\glslabel}%
    {\glxstrpostlinkendsentence}%
    {\glxstrifcustomdiscardperiod

```

```

    {\glxstrifperiod{\glxstrpostlinkendsentence}{\glxstrpostlink}}%
    {\glxstrpostlink}%
  }%
}
\newcommand*{\glxstrifcustomdiscardperiod}[2]{#2}
\newcommand*{\glxstrpostlink}{%
  \csuse{\glxstrpostlink\glscategory{\glslabel}}%
}
\newcommand*{\glsdefpostlink}[2]{%
  \ifthenelse{\equal{#1}{}}{%
    {\PackageError{glossaries-extra}
      {Invalid empty category label in \string\glsdefpostlink}{}}%
    {\csdef{\glxstrpostlink#1}{#2}}%
  }
}
\newcommand*{\glxstrpostlinkendsentence}{%
  \ifcsdef{\glxstrpostlink\glscategory{\glslabel}}
  {%
    \csuse{\glxstrpostlink\glscategory{\glslabel}}%
    .\spacefactor\sfcode'\. \relax
  }%
  {%
    \spacefactor\sfcode'\. \relax
  }%
}
}
\newcommand*{\glxstrpostlinkAddDescOnFirstUse}{%
  \glxstrifwasfirstuse{\space\glxstrparen{\glssaccessdesc{\glslabel}}}{}%
}
\newcommand*{\glxstrpostlinkAddSymbolOnFirstUse}{%
  \glxstrifwasfirstuse
  {%
    \ifglshassymbol{\glslabel}%
    {\space\glxstrparen{\glssaccesssymbol{\glslabel}}}%
    {}%
  }%
  {}%
}
}
\newcommand*{\glxstrpostlinkAddSymbolDescOnFirstUse}{%
  \glxstrifwasfirstuse
  {%
    \space\glxstrparen
    {%
      \ifglshassymbol{\glslabel}%
      {\glssaccesssymbol{\glslabel}, }%
      {}%
    }
    \glssaccessdesc{\glslabel}%
  }%
  }%
  {}%
}
}
\newcommand*{\glxstrdiscardperiod}[3]{%

```



```

\glxtrifwasfirstuse
{%
  \glusifattribute{#1}{retainfirstuseperiod}{true}%
  {#3}%
  {%
    \glusifattribute{#1}{discardperiod}{true}%
    {%
      \glusifplural
      {%
        \glusifattribute{#1}{pluraldiscardperiod}{true}%
        {\glxtrifperiod{#2}{#3}}%
        {#3}%
      }%
      {%
        \glxtrifperiod{#2}{#3}%
      }%
    }%
    {#3}%
  }%
}%
{%
  \glusifattribute{#1}{discardperiod}{true}%
  {%
    \glusifplural
    {%
      \glusifattribute{#1}{pluraldiscardperiod}{true}%
      {\glxtrifperiod{#2}{#3}}%
      {#3}%
    }%
    {%
      \glxtrifperiod{#2}{#3}%
    }%
  }%
  {#3}%
}%
}
\newcommand*{\glxtrifperiod}[1]{\new@ifnextchar.{\@firstoftwo{#1}}}
\newcommand*{\glxtr@punclist}{.,:;!}
\newcommand*{\glxtraddpunctuationmark}[1]{\appto\glxtr@punclist{#1}}
\newcommand*{\glxtrsetpunctuationmarks}[1]{\def\glxtr@punclist{#1}}
\newcommand*{\glxtrifnextpunc}[2]{%
  \def\reserved@a{#1}%
  \def\reserved@b{#2}%
  \futurelet\@glspunc@token\glxtr@ifnextpunc
}
\newcommand*{\glxtr@ifnextpunc}{%
  \glxtr@ifpunctoken{\@glspunc@token}{\let\reserved@b\reserved@a}{}%
  \reserved@b
}
\newcommand*{\glxtr@ifpunctoken}[1]{%

```

```

\expandafter\@glsxtr@ifpunctoken\expandafter#1\glsxtr@punctlist\@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\gls@categorylabel{#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]{%
  \glskeylistok{#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}%
{%
  \@glstr@markwordseps\@gls@short{#3}%
}%
{%
  \glsifcategoryattribute{\glscategorylabel}{insertdots}{true}%
  {%
    \@glstr@insertdots\@gls@short{#3}%
    \appto\@gls@short{\@}%
  }%
  {\def\@gls@short{#3}}%
}%
\glsifcategoryattribute{\glscategorylabel}{aposplural}{true}%
{%
  \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
    '\abrvpluralsuffix}%
}%
{%
  \glsifcategoryattribute{\glscategorylabel}{noshortplural}{true}%
  {%
    \let\@gls@shortpl\@gls@short
  }%
  {%
    \expandafter\def\expandafter\@gls@shortpl\expandafter{\@gls@short
      \abrvpluralsuffix}%
  }%
}%
\expandafter\glsshorttok\expandafter{\@gls@short}%
\glstrnewabrevpresetkeyhook{#1}{#2}{#3}%
\setkeys*{glstrabbrv}[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\@glstr@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=\glstrabbrvtype,%

```

```

        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@\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\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{\Glsxtrinlinefullformat{#2}{#3}}\%
    \glsxtrsetupfulldefs
    \@gls@link{#1}{#2}{\csname gls@\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}[]}}\%
}

```

```

        {\@GLSxtr@full{#1}{#2}[]}%
}
\def\@GLSxtr@full#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{\mfirstucMakeUppercase{\glxtrinlinefullformat{#2}{#3}}}%
    \glsxtrsetupfulldefs
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glxtrfullpl}{\@gl@hyp@opt\ns@glxtrfullpl}
\newcommand*\ns@glxtrfullpl[2][]{%
  \new@ifnextchar[{\@glxtr@fullpl{#1}{#2}}%
    {\@glxtr@fullpl{#1}{#2}[]}%
}
\def\@glxtr@fullpl#1#2[#3]{%
  \@glxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{\glxtrinlinefullplformat{#2}{#3}}%
    \glsxtrsetupfulldefs
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glsxtrfullpl}{\@gl@hyp@opt\ns@Glsxtrfullpl}
\newcommand*\ns@Glsxtrfullpl[2][]{%
  \new@ifnextchar[{\@Glsxtr@fullpl{#1}{#2}}%
    {\@Glsxtr@fullpl{#1}{#2}[]}%
}
\def\@Glsxtr@fullpl#1#2[#3]{%
  \@glxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \glssetabbrvfmt{\glscategory{#2}}%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty

```

```

\def\glscustomtext{\Glsxtrinlinefullplformat{#2}{#3}}%
\glstrsetupfulldefs
\@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{\glxtrinlinefullplformat{#2}{#3}}}%
\glstrsetupfulldefs
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glstrshort}{\@gls@hyp@opt\@ns@glstrshort}
\newcommand*\ns@glstrshort[2][]{%
\new@ifnextchar[{\@glstrshort{#1}{#2}}{\@glstrshort{#1}{#2}[]}%
}
\def\@glstrshort#1#2[#3]{%
\@glsxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\let\glstrifwasfirstuse\@secondoftwo
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glssabbrvfont{\glsaccessshort{#2}\ifglstrinsertinside#3\fi}%
\ifglstrinsertinside\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]{%
\@glxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glxtrifwasfirsttuse\@secondoftwo
\let\gl@ifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glinsert\@empty
\def\glscustomtext{%
\glabbrvfont{\Glsaccessshort{#2}\ifglxtrininsertinside#3\fi}%
\ifglxtrininsertinside\else#3\fi
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLSxtrshort}{\@gl@hyp@opt\@ns@GLSxtrshort}
\newcommand*{\ns@GLSxtrshort}[2] []{%
\new@ifnextchar[{\@GLSxtrshort{#1}{#2}}{\@GLSxtrshort{#1}{#2} []}%
}
\def\@GLSxtrshort#1#2[#3]{%
\@glxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
\glsssetabbrvfmt{\glscategory{#2}}%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glxtrifwasfirsttuse\@secondoftwo
\let\gl@ifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glinsert\@empty
\def\glscustomtext{%
\mfirstucMakeUppercase
{\glabbrvfont{\Glsaccessshort{#2}\ifglxtrininsertinside#3\fi}%
\ifglxtrininsertinside\else#3\fi
}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glxtrlong}{\@gl@hyp@opt\@ns@glxtrlong}
\newcommand*{\ns@glxtrlong}[2] []{%
\new@ifnextchar[{\@glxtrlong{#1}{#2}}{\@glxtrlong{#1}{#2} []}%
}
\def\@glxtrlong#1#2[#3]{%
\@glxtr@record{#1}{#2}{glslink}%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \let\glxtrifwasfirstuse\@secondoftwo
  \let\gl@ifplural\@secondoftwo
  \let\glscapscase\@firstofthree
  \let\glinsert\@empty
  \def\glscustomtext{%
    \glslongfont{\glaccesslong{#2}\ifglxtrininsertinside#3\fi}%
    \ifglxtrininsertinside\else#3\fi
  }%
  \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glsxtrlong}{\@gl@hyp@opt\@ns@Glsxtrlong}
\newcommand*{\ns@Glsxtrlong}[2][]{%
  \new@ifnextchar[{\@Glsxtrlong{#1}{#2}}{\@Glsxtrlong{#1}{#2} []}%
}
\def\@Glsxtrlong#1#2[#3]{%
  \@glxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\gl@ifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glinsert\@empty
    \def\glscustomtext{%
      \glslongfont{\Glsaccesslong{#2}\ifglxtrininsertinside#3\fi}%
      \ifglxtrininsertinside\else#3\fi
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSxtrlong}{\@gl@hyp@opt\@ns@GLSxtrlong}
\newcommand*{\ns@GLSxtrlong}[2][]{%
  \new@ifnextchar[{\@GLSxtrlong{#1}{#2}}{\@GLSxtrlong{#1}{#2} []}%
}
\def\@GLSxtrlong#1#2[#3]{%
  \@glxtr@record{#1}{#2}{glslink}%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \let\glxtrifwasfirstuse\@secondoftwo
    \let\gl@ifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glinsert\@empty
    \def\glscustomtext{%

```

```

\mfirstucMakeUppercase
{\glslongfont{\glsaccesslong{#2}\ifglxtrinsertinside#3\fi}%
\ifglxtrinsertinside\else#3\fi
}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glxtrshortpl}{\@gls@hyp@opt\@ns@glxtrshortpl}
\newcommand*{\ns@glxtrshortpl}[2][]{%
\new@ifnextchar[{\@glxtrshortpl{#1}{#2}}{\@glxtrshortpl{#1}{#2}[]}%
}
\def\@glxtrshortpl#1#2[#3]{%
\@glxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
\glssetabbrvfmt{\glscategory{#2}}%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\let\glxtrifwasfirsttuse\@secondoftwo
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsabbrvfont{\glsaccessshortpl{#2}\ifglxtrinsertinside#3\fi}%
\ifglxtrinsertinside\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]{%
\@glxtr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
\glssetabbrvfmt{\glscategory{#2}}%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\let\glxtrifwasfirsttuse\@secondoftwo
\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsabbrvfont{\Glsaccessshortpl{#2}\ifglxtrinsertinside#3\fi}%
\ifglxtrinsertinside\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{\gls@category{#2}}%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsxtrifwasfirstuse\@secondoftwo
    \let\glsifplural\@firstoftwo
    \let\gls@capscase\@thirdofthree
    \let\glsinsert\@empty
    \def\gls@customtext{%
      \mfirstucMakeUppercase
      {\gls@abbrvfont{\gls@accessshortpl{#2}\ifglsxtrininsertinside#3\fi}%
      \ifglsxtrininsertinside\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\gls@capscase\@firstofthree
    \let\glsinsert\@empty
    \def\gls@customtext{%
      \gls@longfont{\gls@accesslongpl{#2}\ifglsxtrininsertinside#3\fi}%
      \ifglsxtrininsertinside\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]{%
\@glstr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glstrifwasfirstuse\@secondoftwo
\let\gl@ifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glinsert\@empty
\def\glscustomtext{%
\glslongfont{\Glsaccesslongpl{#2}\ifglstrinsertinside#3\fi}%
\ifglstrinsertinside\else#3\fi
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLSxtrlongpl}{\@gl@hyp@opt\@ns@GLSxtrlongpl}
\newcommand*{\ns@GLSxtrlongpl}[2][]{%
\new@ifnextchar[{\@GLSxtrlongpl{#1}{#2}}{\@GLSxtrlongpl{#1}{#2}[]}%
}
\def\@GLSxtrlongpl#1#2[#3]{%
\@glstr@record{#1}{#2}{glslink}%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\let\glstrifwasfirstuse\@secondoftwo
\let\gl@ifplural\@firstoftwo
\let\glscapscase\@thirdofthree
\let\glinsert\@empty
\def\glscustomtext{%
\mfirstucMakeUppercase
{\glslongfont{\Glsaccesslongpl{#2}\ifglstrinsertinside#3\fi}%
\ifglstrinsertinside\else#3\fi
}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\glsetabbrvfmt}[1]{%
\ifcsdef{\gl@abbrv@current@#1}%
{\glstr@applyabbrvfmt{\csname @gl@abbrv@current@#1\endcsname}}%
{\glstr@applyabbrvfmt{\@gl@abbrv@current@abbreviation}}%
}
\newrobustcmd*{\gl@useabbrvfont}[2]{\@glsetabbrvfmt{#2}\gl@abbrvfont{#1}}
\newrobustcmd*{\gl@uselongfont}[2]{\@glsetabbrvfmt{#2}\glslongfont{#1}}
\newcommand*{\glstrgenabbrvfmt}{%

```

```

\ifdefempty\glscustomtext
{%
  \ifglused\glslabel
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \glxtrsubsequentplfmt{\glslabel}{\glsinsert}%
      }%
      {%
        \Glsxtrsubsequentplfmt{\glslabel}{\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glxtrsubsequentplfmt{\glslabel}{\glsinsert}}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \glxtrsubsequentfmt{\glslabel}{\glsinsert}%
    }%
    {%
      \Glsxtrsubsequentfmt{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\glxtrsubsequentfmt{\glslabel}{\glsinsert}}%
    }%
  }%
}
\glsifplural
{%
  \glscapscase
  {%
    \glxtrfullplformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Glsxtrfullplformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
    {\glxtrfullplformat{\glslabel}{\glsinsert}}%
  }%
}
  \glscapscase
  {%

```

```

        \glxtrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Glsxtrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \mfirstucMakeUppercase
        {\glxtrfullformat{\glslabel}{\glsinsert}}%
    }%
}
}%
}%
{%
    \glscustomtext
}%
}
\newcommand*{\glxtrsubsequentfmt}[2]{%
    \glsabbrvfont{\glsaccessshort{#1}\ifglxtrininsertinside #2\fi}%
    \ifglxtrininsertinside \else#2\fi
}
\let\glxtrdefaultsubsequentfmt\glxtrsubsequentfmt
\newcommand*{\glxtrsubsequentplfmt}[2]{%
    \glsabbrvfont{\glsaccessshortpl{#1}\ifglxtrininsertinside #2\fi}%
    \ifglxtrininsertinside \else#2\fi
}
\let\glxtrdefaultsubsequentplfmt\glxtrsubsequentplfmt
\newcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glsabbrvfont{\Glsaccessshort{#1}\ifglxtrininsertinside #2\fi}%
    \ifglxtrininsertinside \else#2\fi
}
\let\Glsxtrdefaultsubsequentfmt\Glsxtrsubsequentfmt
\newcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glsabbrvfont{\Glsaccessshortpl{#1}\ifglxtrininsertinside #2\fi}%
    \ifglxtrininsertinside \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\@glxtr@dostylewarn{}%
            \glsforeachincategory{#1}{\@gls@type}{\@gls@label}%
            {%
                \def\@glxtr@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
}%
\@glxtr@dostylewarn
\csdef{@glxtr@current@#1}{#2}%
\protected@edef\glxtr@categorylabel{#1}%
\glxtr@applyabbrvstyle{#2}%
}%
}
\newcommand*{\glxtr@applyabbrvstyle}[1]{%
    \csuse{@glxtr@dispstyle@setup@#1}%
    \csuse{@glxtr@dispstyle@fmts@#1}%
}
\newcommand*{\glxtr@applyabbrvfmt}[1]{%
    \csuse{@glxtr@dispstyle@fmts@#1}%
}
\newcommand*{\newabbreviationstyle}[3]{%
    \ifcsdef{@glxtr@dispstyle@setup@#1}
    {%
        \PackageError{glossaries-extra}{Abbreviation style '#1' already
        defined}{}%
    }%
    {%
        \csdef{@glxtr@dispstyle@setup@#1}{%
            \renewcommand*{\GlsXtrPostNewAbbreviation}{}%
            #2}%
        \csdef{@glxtr@dispstyle@fmts@#1}{%
            \renewcommand*{\glxtrinlinefullformat}{\glxtrfullformat}%
            \renewcommand*{\Glsxtrinlinefullformat}{\Glsxtrfullformat}%
            \renewcommand*{\glxtrinlinefullplformat}{\glxtrfullplformat}%
            \renewcommand*{\Glsxtrinlinefullplformat}{\Glsxtrfullplformat}%
            \let\glxtrsubsequentfmt\glxtrdefaultsubsequentfmt
            \let\glxtrsubsequentplfmt\glxtrdefaultsubsequentplfmt
            \let\Glsxtrsubsequentfmt\Glsxtrdefaultsubsequentfmt
            \let\Glsxtrsubsequentplfmt\Glsxtrdefaultsubsequentplfmt
            #3}%
        }%
    }
}
\newcommand*{\renewabbreviationstyle}[3]{%
    \ifcsundef{@glxtr@dispstyle@setup@#1}
    {%
        \PackageError{glossaries-extra}{Abbreviation style '#1' not defined}{}%
    }%
    {%
        \csdef{@glxtr@dispstyle@setup@#1}{%

```



```

\renewcommand*{\GlsXtrPostNewAbbreviation}{}%
#2}%
\csdef{@glsabbrv@dispstyle@fmts@#1}{%
\renewcommand*{\glxtrinlinefullformat}{\glxtrfullformat}%
\renewcommand*{\glxtrinlinefullplformat}{\glxtrfullplformat}%
\renewcommand*{\glxtrinlinefullplformat}{\glxtrfullplformat}%
\renewcommand*{\glxtrinlinefullplformat}{\glxtrfullplformat}%
#3}%
}%
}
\newcommand*{\letabbreviationstyle}[2]{%
\csletcs{@glsabbrv@dispstyle@setup@#1}{@glsabbrv@dispstyle@setup@#2}%
\csletcs{@glsabbrv@dispstyle@fmts@#1}{@glsabbrv@dispstyle@fmts@#2}%
}
\newcommand*{@glxtr@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\ifglxtrinsertinside
\glxtrinsertinsidefalse
\newcommand*{\glxtrlongshortname}{%
\protect\glsabbrvfont{\the\glsshorttok}%
}

```

```

\newabbreviationstyle{long-short}%
{%
  \glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glstrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glstrfirstlongfont{\the\glslongtok}%
      \protect\glstrfullsep{\the\glslabeltok}%
      \glstrparen{\protect\glstrfirstabbrvfont{\the\glsshorttok}}},%
    firstplural={\protect\glstrfirstlongfont{\the\glslongpltok}%
      \protect\glstrfullsep{\the\glslabeltok}%
      \glstrparen{\protect\glstrfirstabbrvfont{\the\glsshortpltok}}},%
    plural={\protect\glstrabbrvfont{\the\glsshortpltok}},%
    text={\protect\glstrabbrvfont{\the\glsshorttok}},%
    description={\the\glslongtok}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glissetattribute{\the\glslabeltok}{regular}{false}%
    }%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
  \renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvdefaultfont{##1}}%
  \renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvdefaultfont{##1}}%
  \renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glstrfullformat}[2]{%
    \glstrfirstlongfont{\glstraccesslong{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
    \glstrfullsep{##1}%
    \glstrparen{\glstrfirstabbrvfont{\glstraccessshort{##1}}}%
  }%
  \renewcommand*{\glstrfullplformat}[2]{%
    \glstrfirstlongfont{\glstraccesslongpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
    \glstrparen{\glstrfirstabbrvfont{\glstraccessshortpl{##1}}}%
  }%
  \renewcommand*{\GlsXtrfullformat}[2]{%
    \glstrfirstlongfont{\glstraccesslong{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
    \glstrparen{\glstrfirstabbrvfont{\glstraccessshort{##1}}}%
  }%
  \renewcommand*{\GlsXtrfullplformat}[2]{%
    \glstrfirstlongfont{\glstraccesslongpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
    \glstrparen{\glstrfirstabbrvfont{\glstraccessshortpl{##1}}}%
  }%
}

```

```

}
\setabbreviationstyle{long-short}
\newcommand*{\glxtrlongshortdescsort}{%
  \expandonce\glxtrorglong\space (\expandonce\glxtrorgshort)%
}
\newcommand*{\glxtrlongshortdescname}{%
  \protect\glslongfont{\the\glslongtok}
  \glxtrparen{\protect\glsabbrvfont{\the\glsshorttok}}%
}
\newabbreviationstyle{long-short-desc}%
{%
  \glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrlongshortdescname},
    sort={\glxtrlongshortdescsort},%
    first={\protect\glsfirstlongfont{\the\glslongtok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstabbrvfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongfont{\the\glslongpltok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstabbrvfont{\the\glsshortpltok}}},%
    text={\protect\glsabbrvfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
\GlsXtrUseAbbrStyleFmts{long-short}%
}
\newcommand*{\glxtrshortlongname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
}
\newabbreviationstyle{short-long}%
{%
  \glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrshortlongname},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstlongfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
      \protect\glxtrfullsep{\the\glslabeltok}%

```

```

\glxtrparen{\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}{\glxtrabbrvpluralsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
}%
}%
\newcommand*{\glxtrshortlongdescsort}{\the\glsshorttok}
\newcommand*{\glxtrshortlongdescname}{%
\protect\glsabbrvfont{\the\glsshorttok}
\glxtrparen{\protect\glslongfont{\the\glslongtok}}}%
}
\newabbreviationstyle{short-long-desc}%
{%
\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortlongdescname},

```

```

sort={\glxtrshortlongdescsort},
first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
\protect\glxtrfullsep{\the\glslabeltok}%
\glxtrparen{\protect\glsfirstlongfont{\the\glslongtok}}},%
firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
\protect\glxtrfullsep{\the\glslabeltok}%
\glxtrparen{\protect\glsfirstlongfont{\the\glslongpltok}}},%
text={\protect\glsabbrvfont{\the\glsshorttok}},%
plural={\protect\glsabbrvfont{\the\glsshortpltok}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{\%
\GlsXtrUseAbbrStyleFmts{short-long}%
}
\newcommand*{\glsfirstlongfootnotefont}[1]{\glslongfootnotefont{#1}}%
\newcommand*{\glslongfootnotefont}[1]{\glslongdefaultfont{#1}}%
\newcommand*{\glxtrabbrvfootnote}[2]{\footnote{#2}}
\newcommand*{\glxtrfootnotename}{%
\protect\glsabbrvfont{\the\glsshorttok}%
}
\newabbreviationstyle{footnote}%
{%
\glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrfootnotename},
sort={\the\glsshorttok},
description={\the\glslongtok},%
first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
\protect\glxtrabbrvfootnote{\the\glslabeltok}%
{\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
\protect\glxtrabbrvfootnote{\the\glslabeltok}%
{\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
text={\protect\glsabbrvfont{\the\glsshorttok}},%
plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glissetattribute{\the\glslabeltok}{nohyperfirst}{true}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}%

```

```

}%
{%
\renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
\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}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \protect\glstrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \protect\glstrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \protect\glstrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \protect\glstrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glxtrininlinefullformat}[2]{%
  \glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glxtrininlinefullplformat}[2]{%
  \glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrininlinefullformat}[2]{%
  \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrininlinefullplformat}[2]{%
  \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}

```

```

}%
}
\letabbreviationstyle{short-footnote}{footnote}
\newcommand*{\glxtrfootnotedesname}{%
  \protect\glsabbrvfont{\the\glsshorttok}%
  \protect\glxtrfullsep{\the\glslabeltok}%
  \protect\glxtrparen{\protect\glslongfont{\the\glslongtok}}%
}
\newcommand*{\glxtrfootnotedesort}{\the\glsshorttok}
\newabbreviationstyle{short-footnote-desc}{%
  {%
    \glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glxtrfootnotedesname},
      sort={\glxtrfootnotedesort},
      first={\protect\glsfirstabbrvfont{\the\glsshorttok}%
        \protect\glxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnote{\the\glslongtok}}},%
      firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}%
        \protect\glxtrabbrvfootnote{\the\glslabeltok}%
        {\protect\glsfirstlongfootnote{\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}%
      }%
      {}%
    }%
  }%
}%
\letabbreviationstyle{footnote-desc}{short-footnote-desc}
\newabbreviationstyle{postfootnote}{%
  {%
    \glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%
      name={\glxtrfootnotename},
      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{glxtrpostlink\glscategorylabel}{%
        \glxtrifwasfirstuse

```

```

    {%
      \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}%
        {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}}%
    }%
  }%
  {%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glissetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
  \renewcommand*{\glxtrsetupfulldefs}{%
    \let\glxtrifwasfirstuse\@secondoftwo
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
  \renewcommand*{\glssabbrvfont}[1]{\glssabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glxtrfullformat}[2]{%
    \glsfirstabbrvfont{\glssaccessshort{##1}\ifglxtrininsertinside##2\fi}%
    \ifglxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\glxtrfullplformat}[2]{%
    \glsfirstabbrvfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
    \ifglxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
    \ifglxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
    \ifglxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\glxtrinlinefullformat}[2]{%
    \glsfirstabbrvfont{\glssaccessshort{##1}\ifglxtrininsertinside##2\fi}%
    \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
  }%
  \renewcommand*{\glxtrinlinefullplformat}[2]{%
    \glsfirstabbrvfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
    \ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%

```



```

        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\glsfirstlongfootnotefont{\glaccesslong{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
        \glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\glsfirstlongfootnotefont{\glaccesslongpl{##1}}}%
    }%
}
\letabbreviationstyle{short-postfootnote}{postfootnote}
\newabbreviationstyle{short-postfootnote-desc}%
{%
    \glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%
        name={\glxtrfootnotedesname},
        sort={\glxtrfootnotedesort},
        first={\protect\glsfirstabbrvfont{\the\glsshorttok}},%
        firstplural={\protect\glsfirstabbrvfont{\the\glsshortpltok}},%
        text={\protect\glsabbrvfont{\the\glsshorttok}},%
        plural={\protect\glsabbrvfont{\the\glsshortpltok}}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \csdef{glxtrpostlink\glscategorylabel}{%
            \glxtrifwasfirstuse
            {%
                \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}%
                    {\glsfirstlongfootnotefont{\glentrylong{\glslabel}}}}}%
            }%
        }%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
            \glissetattribute{\the\glslabeltok}{regular}{false}%
        }%
    }%
    \renewcommand*{\glxtrsetupfulldefs}{%
        \let\glxtrifwasfirstuse\@secondoftwo
    }%
}%
{%
    \GlsXtrUseAbbrStyleFmts{postfootnote}%
}
\letabbreviationstyle{postfootnote-desc}{short-postfootnote-desc}
\newcommand*{\glxtrshortnolongname}{%
    \protect\glsabbrvfont{\the\glsshorttok}%
}
\newabbreviationstyle{short}%
{%
    \glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%

```

```

name={\glxtrshortnolongname},
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}{\glxtrabbrvpluralsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\protect\glsfirstabbrvfont{\glsaccessshort{##1}}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\protect\glsfirstabbrvfont{\glsaccessshort{##1}}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\Glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\Glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstabbrvfont{\glsaccessshort{##1}}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstabbrvfont{\glsaccessshortpl{##1}}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvfont{\glsaccessshort{##1}}\ifglxtrinsertinside##2\fi}%

```

```

\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
}
\setabbreviationstyle[acronym]{short}
\letabbreviationstyle{short-nolong}{short}
\newabbreviationstyle{short-nolong-noreg}{%
{%
\GlsXtrUseAbbrStyleSetup{short-nolong}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glsattribute{\the\glslabeltok}{regular}%
{%
\glssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}%
}%
{%
\GlsXtrUseAbbrStyleFmts{short-nolong}%
}
\newcommand*{\glxtrshortdescname}{%
\protect\glsabbrvfont{\the\glsshorttok}%
\protect\glxtrfullsep{\the\glslabeltok}%
\protect\glxtrparen{\protect\glslongfont{\the\glslongtok}}%
}
\newabbreviationstyle{short-desc}{%
{%
\glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortdescname},
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}{\glxtrabbrvpluralsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
}
}

```

```

\glxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstabbrvfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstabbrvfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
}
\letabbreviationstyle{short-nolong-desc}{short-desc}
\newabbreviationstyle{short-nolong-desc-noreg}%
{%
\GlsXtrUseAbbrStyleSetup{short-nolong-desc}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}%
\GlsXtrUseAbbrStyleFmts{short-nolong-desc}%
}
\newabbreviationstyle{nolong-short}%

```

```

{%
  \GlsXtrUseAbbrStyleSetup{short-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-nolong}%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongfont{\glssaccesslong{##1}}%
    \ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glssaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongfont{\glssaccesslongpl{##1}}%
    \ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\glssaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongfont{\glssaccesslong{##1}}%
    \ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\Glsaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongfont{\glssaccesslongpl{##1}}%
    \ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvfont{\Glsaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{nolong-short-noreg}%
{%
  \GlsXtrUseAbbrStyleSetup{nolong-short}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {%
  }%
}
}%
{%
  \GlsXtrUseAbbrStyleFmts{nolong-short}%
}
\newcommand*{\glsxtrlongnoshortdescname}{%
  \protect\glslongfont{\the\glslongtok}%
}
\newabbreviationstyle{long-desc}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%

```

```

name={\glxtrlongnoshortdescname},
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}{\glxtrabbrvpluralsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvdefaultfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glxtrsubsequentfmt}[2]{%
\glslongfont{\glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\glxtrsubsequentplfmt}[2]{%
\glslongfont{\glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
\glslongfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
\glslongfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\glsfirstlongfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glsfirstlongfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstlongfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstlongfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%

```

```

\glstrparen{\protect\glsfirstabbrvfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glstrfullformat}[2]{%
\glsfirstlongfont{\glsaccesslong{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi
}%
\renewcommand*{\glstrfullplformat}[2]{%
\glsfirstlongfont{\glsaccesslongpl{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstlongfont{\glsaccesslong{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstlongfont{\glsaccesslongpl{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi
}%
}
\letabbreviationstyle{long-noshort-desc}{long-desc}
\newabbreviationstyle{long-noshort-desc-noreg}%
{%
\GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glsssetAttribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}%
{%
\GlsXtrUseAbbrStyleFmts{long-noshort-desc}%
}
\newcommand*{\glxtrlongnoshortname}{%
\protect\glsabbrvfont{\the\glssshorttok}%
}
\newabbreviationstyle{long}%
{%
\glxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrlongnoshortname},
sort={\the\glssshorttok},
first={\protect\glsfirstlongfont{\the\glslongtok}},
firstplural={\protect\glsfirstlongfont{\the\glslongpltok}},
text={\glslongfont{\the\glslongtok}},
plural={\glslongfont{\the\glslongpltok}},%
description={\the\glslongtok}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%

```

```

\glsssetattribute{\the\glslabeltok}{regular}{true}}%
}%
{%
\GlsXtrUseAbbrStyleFmts{long-desc}%
}
\letabbreviationstyle{long-noshort}{long}
\newabbreviationstyle{long-noshort-noreg}%
{%
\GlsXtrUseAbbrStyleSetup{long-noshort}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\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}%
{%
\glsssetattribute{\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}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi
  \glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
}%
\renewcommand*\glsxtrfullplformat[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*\Glsxtrfullformat[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
}%
\renewcommand*\Glsxtrfullplformat[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\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}%
{%
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glstrshortlongname},
sort={\the\glsshorttok},
description={\the\glslongtok},%
first={\protect\glstrfirstabbrvscfont{\the\glsshorttok}%
\protect\glstrfullsep{\the\glslabeltok}%
\glstrparen{\protect\glstrfirstlongdefaultfont{\the\glslongtok}}},%
firstplural={\protect\glstrfirstabbrvscfont{\the\glsshortpltok}%
\protect\glstrfullsep{\the\glslabeltok}%
\glstrparen{\protect\glstrfirstlongdefaultfont{\the\glslongpltok}}},%
text={\protect\glstrabbrvscfont{\the\glsshorttok}},%
plural={\protect\glstrabbrvscfont{\the\glsshortpltok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{%
\renewcommand*{\abbrvpluralsuffix}{\glstrscsuffix}%
\renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvscfont{##1}}%
\renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvscfont{##1}}%
\renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlongdefaultfont{##1}}%
\renewcommand*{\glstrlongfont}[1]{\glstrlongdefaultfont{##1}}%
\renewcommand*{\glstrfullformat}[2]{%
\glstrfirstabbrvscfont{\glstraccessshort{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi
\glstrfullsep{##1}%
\glstrparen{\glstrfirstlongdefaultfont{\glstraccesslong{##1}}}%
}%
\renewcommand*{\glstrfullplformat}[2]{%
\glstrfirstabbrvscfont{\glstraccessshortpl{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi
\glstrfullsep{##1}%
\glstrparen{\glstrfirstlongdefaultfont{\glstraccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glstrfirstabbrvscfont{\Glsstraccessshort{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
\glstrparen{\glstrfirstlongdefaultfont{\glstraccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glstrfirstabbrvscfont{\Glsstraccessshortpl{##1}\ifglstrinsertinside##2\fi}%

```

```

        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\glsfirstlongdefaultfont{\glssaccesslongpl{##1}}}%
    }%
}
\newabbreviationstyle{short-sc-long-desc}%
{%
    \glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%
        name={\glxtrshortlongdescname},
        sort={\glxtrshortlongdescsort},
        first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
            \protect\glxtrfullsep{\the\glslabeltok}%
            \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
        firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
            \protect\glxtrfullsep{\the\glslabeltok}%
            \glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
        text={\protect\glssabbrvscfont{\the\glsshorttok}},%
        plural={\protect\glssabbrvscfont{\the\glsshortpltok}}%
    }%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glshasattribute{\the\glslabeltok}{regular}%
        {%
            \glsssetAttribute{\the\glslabeltok}{regular}{false}%
        }%
        {}%
    }%
}%
{%
    \GlsXtrUseAbbrStyleFmts{short-sc-long}%
}
\newabbreviationstyle{short-sc}%
{%
    \glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%
        name={\glxtrshortnolongname},
        sort={\the\glsshorttok},
        first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},
        firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}},
        text={\protect\glssabbrvscfont{\the\glsshorttok}},
        plural={\protect\glssabbrvscfont{\the\glsshortpltok}},
        description={\the\glslongtok}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%
    \renewcommand*{\abbrvpluralsuffix}{\glxtrscsuffix}%
    \renewcommand*{\glssabbrvfont[1]}{\glssabbrvscfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont[1]}{\glsfirstabbrvscfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
}

```

```

\renewcommand*{\glxtrinlinefullformat}[2]{%
  \protect\glsfirstabbrvscfont{\glsaccessshort{##1}%
    \ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}%
    \ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \protect\glsfirstabbrvscfont{\Glsaccessshort{##1}%
    \ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \protect\glsfirstabbrvscfont{\Glsaccessshortpl{##1}%
    \ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
  \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstabbrvscfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstabbrvscfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
}
\letabbreviationstyle{short-sc-nolong}{short-sc}
\newabbreviationstyle{short-sc-desc}%
{%
  \glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrshortdescname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}},
    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}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
\glsfirstabbrvscfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glsxtrfullformat}[2]{%
\glsfirstabbrvscfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
\glsfirstabbrvscfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvscfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvscfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
}%
\letabbreviationstyle{short-sc-nolong-desc}{short-sc-desc}
\newabbreviationstyle{nolong-short-sc}%

```

```

{%
  \GlsXtrUseAbbrStyleSetup{short-sc-nolong}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sc-nolong}%
  \renewcommand*{\glxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glssaccesslong{##1}%
      \ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\glsfirstabbrvscfont{\glssaccessshort{##1}}}%
  }%
  \renewcommand*{\glxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\glssaccesslongpl{##1}%
      \ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\glsfirstabbrvscfont{\glssaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslong{##1}%
      \ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\glsfirstabbrvscfont{\glssaccessshort{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \protect\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}%
      \ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\glsfirstabbrvscfont{\glssaccessshortpl{##1}}}%
  }%
}
\newabbreviationstyle{long-noshort-sc}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrlongnoshortname},
    sort={\the\glssshorttok},
    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}{%
    \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glxtrscsuffix}%
  \renewcommand*{\glabbrvfont}[1]{\glabbrvscfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
}

```

```

\renewcommand*{\glxtrsubsequentfmt}[2]{%
  \glslongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
  \ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\glxtrsubsequentplfmt}[2]{%
  \glslongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
  \ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
  \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
  \ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
  \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
  \ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvscfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%

```

```

        \ifglxtrinsertinside\else##2\fi
    }%
}
\@glxtr@deprecated@abbrstyle{long-sc}{long-noshort-sc}
\newabbreviationstyle{long-noshort-sc-desc}%
{%
    \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
}%
{%
    \renewcommand*{\abbrvpluralsuffix}{\glxtrscsuffix}%
    \renewcommand*{\glabbrvfont}[1]{\glabbrvscfont{##1}}%
    \renewcommand*{\glfirstabbrvfont}[1]{\glfirstabbrvscfont{##1}}%
    \renewcommand*{\glfirstlongfont}[1]{\glfirstlongdefaultfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
    \renewcommand*{\glxtrsubsequentfmt}[2]{%
        \glslongdefaultfont{\glaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
        \ifglxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\glxtrsubsequentplfmt}[2]{%
        \glslongdefaultfont{\glaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
        \ifglxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
        \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
        \ifglxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
        \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
        \ifglxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\glxtrinlinefullformat}[2]{%
        \glfirstlongdefaultfont{\glaccesslong{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\protect\glfirstabbrvscfont{\glaccessshort{##1}}}%
    }%
    \renewcommand*{\glxtrinlinefullplformat}[2]{%
        \glfirstlongdefaultfont{\glaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\protect\glfirstabbrvscfont{\glaccessshortpl{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullformat}[2]{%
        \glfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\protect\glfirstabbrvscfont{\glaccessshort{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
        \glfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\protect\glfirstabbrvscfont{\glaccessshortpl{##1}}}%
    }%
}

```



```

\renewcommand*{\glxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\glaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\glaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\glaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\glaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
}
\@glxtr@deprecated@abbrstyle{long-desc-sc}{long-noshort-sc-desc}
\newabbreviationstyle{short-sc-footnote}%
{%
  \glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrfootnotename},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvscfont{\the\glsshorttok}%
      \protect\glxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}%
      \protect\glxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvscfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsssetAttribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glxtrscsuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glxtrfullformat}[2]{%
    \glsfirstabbrvscfont{\glaccessshort{##1}\ifglxtrinsertinside##2\fi}%

```

```

\ifglxtrinsertinside\else##2\fi
\protect\glxtrabbrvfootnote{##1}%
{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstabbrvscfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\protect\glxtrabbrvfootnote{##1}%
{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\protect\glxtrabbrvfootnote{##1}%
{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\protect\glxtrabbrvfootnote{##1}%
{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\glxtrininlinefullformat}[2]{%
\glsfirstabbrvscfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\glxtrininlinefullplformat}[2]{%
\glsfirstabbrvscfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrininlinefullformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrininlinefullplformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
}
\@glxtr@deprecated@abbrstyle{footnote-sc}{short-sc-footnote}
\newabbreviationstyle{short-sc-footnote-desc}%
{%
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrfootnotedesname},
sort={\glxtrfootnotedesort},

```

```

first={\protect\glsfirstabbrvscfont{\the\glsshorttok}}%
\protect\glstrabbrvfootnote{\the\glslabeltok}%
  {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
firstplural={\protect\glsfirstabbrvscfont{\the\glsshortpltok}}%
\protect\glstrabbrvfootnote{\the\glslabeltok}%
  {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
text={\protect\glsabbrvscfont{\the\glsshorttok}},%
plural={\protect\glsabbrvscfont{\the\glsshortpltok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
\glsattribute{\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\glstrabbrvfootnote{\glslabel}}%
{\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}}%
}%
}%
\glsattribute{\the\glslabeltok}{regular}%
{%
\glssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
\renewcommand*{\glsxtrsetupfulldefs}{%
\let\glsxtrifwasfirstuse\@secondoftwo
}%
}%

```

```

{%
\renewcommand*{\abbrvpluralsuffix}{\glxtrscsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvscfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvscfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstabbrvscfont{\glssaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstabbrvscfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
}%
\renewcommand*{\glxtrininlinefullformat}[2]{%
\glsfirstabbrvscfont{\glssaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\glxtrininlinefullplformat}[2]{%
\glsfirstabbrvscfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrininlinefullformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrininlinefullplformat}[2]{%
\glsfirstabbrvscfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
}
\@glxtr@deprecated@abbrstyle{postfootnote-sc}{short-sc-postfootnote}
\newabbreviationstyle{short-sc-postfootnote-desc}%
{%
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrfootnotedesname},
sort={\glxtrfootnotedesort},

```

```

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{\glstrylong{\glslabel}}}}}%
}%
}%
\glsattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\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}{%
\glsattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\the\glslabeltok}{regular}{false}%
}
}

```

```

    }%
    {}%
  }%
}%
{%
  \renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
  \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
  \renewcommand*\{abbrvpluralsuffix}{\glsxtrmsuffix}%
  \renewcommand*\{glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*\{glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*\{glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
    \glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*\{glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*\{Glsxtrfullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*\{Glsxtrfullplformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\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}{%
    \glsattribute{\the\glslabeltok}{regular}%
    {%

```

```

        \glsssetattribute{\the\glslabelltok}{regular}{false}%
    }%
    {}%
}%
}%
{%
    \GlsXtrUseAbbrStyleFmts{long-short-sm}%
}
\newabbreviationstyle{short-sm-long}%
{%
    \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
    \renewcommand*{\CustomAbbreviationFields}{%
        name={\glsxtrshortlongname},
        sort={\the\glssshorttok},
        description={\the\glslongtok},%
        first={\protect\glsfirstabbrvsmfont{\the\glssshorttok}%
            \protect\glsxtrfullsep{\the\glslabelltok}%
            \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
        firstplural={\protect\glsfirstabbrvsmfont{\the\glssshortpltok}%
            \protect\glsxtrfullsep{\the\glslabelltok}%
            \glsxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
        text={\protect\glssabbrvsmfont{\the\glssshorttok}},%
        plural={\protect\glssabbrvsmfont{\the\glssshortpltok}}}%
    \renewcommand*{\GlsXtrPostNewAbbreviation}{%
        \glshasattribute{\the\glslabelltok}{regular}%
        {%
            \glsssetattribute{\the\glslabelltok}{regular}{false}%
        }%
        {}%
    }%
}%
}%
{%
    \renewcommand*\glssabbrvfont[1]{\glssabbrvsmfont{##1}}%
    \renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
    \renewcommand*\abbrvpluralsuffix{\glsxtrsmsuffix}%
    \renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
    \renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%
    \renewcommand*\glsxtrfullformat[2]{%
        \glsfirstabbrvsmfont{\glssaccessshort{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
        \glsxtrfullsep{##1}%
        \glsxtrparen{\glsfirstlongdefaultfont{\glssaccesslong{##1}}}%
    }%
    \renewcommand*\glsxtrfullplformat[2]{%
        \glsfirstabbrvsmfont{\glssaccessshortpl{##1}\ifglsxtrinsertinside##2\fi}%
        \ifglsxtrinsertinside\else##2\fi
        \glsxtrfullsep{##1}%
        \glsxtrparen{\glsfirstlongdefaultfont{\glssaccesslongpl{##1}}}%
    }%
    \renewcommand*\glsxtrfullformat[2]{%

```

```

\glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
}
\newabbreviationstyle{short-sm-long-desc}%
{%
\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortlongdescname},
sort={\glxtrshortlongdescsort},
first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}}%
\protect\glxtrfullsep{\the\glslabeltok}}%
\glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}}%
\protect\glxtrfullsep{\the\glslabeltok}}%
\glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
\GlsXtrUseAbbrStyleFmts{short-sm-long}%
}
\newabbreviationstyle{short-sm}%
{%
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortnolongname},
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}{%
\glissetattribute{\the\glslabeltok}{regular}{true}}%
}%

```



```

{%
\renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
\renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
\renewcommand*\abbrvpluralsuffix{\glsxtrmsuffix}%
\renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%
\renewcommand*\glsxtrinlinefullformat[2]{%
\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}%
\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*\glsxtrinlinefullplformat[2]{%
\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}%
\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*\Glsxtrinlinefullformat[2]{%
\protect\glsfirstabbrvsmfont{\Glsaccessshort{##1}}%
\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*\Glsxtrinlinefullplformat[2]{%
\protect\glsfirstabbrvsmfont{\Glsaccessshortpl{##1}}%
\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*\glsxtrfullformat[2]{%
\glsfirstabbrvsmfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*\glsxtrfullplformat[2]{%
\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*\Glsxtrfullformat[2]{%
\glsfirstabbrvsmfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*\Glsxtrfullplformat[2]{%
\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\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}{\glsxtrmsuffix}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glsxtrinlinefullformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstabbrvsmfont{\Glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstabbrvsmfont{\Glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%

```

```

\glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
}
\letabbreviationstyle{short-sm-nolong-desc}{short-sm-desc}
\newabbreviationstyle{nolong-short-sm}%
{%
\GlsXtrUseAbbrStyleSetup{short-sm-nolong}%
}%
{%
\GlsXtrUseAbbrStyleFmts{short-sm-nolong}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\protect\glsfirstlongdefaultfont{\glsaccesslong{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\protect\glsfirstlongdefaultfont{\glsaccesslongpl{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\protect\glsfirstlongdefaultfont{\Glsaccesslong{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\protect\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
}%
}
\newabbreviationstyle{long-noshort-sm}%
{%
\glsxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrlongnoshortname},
sort={\the\glsshorttok},
first={\protect\glsfirstlongdefaultfont{\the\glslongtok}},
firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},
text={\protect\glsfirstlongdefaultfont{\the\glslongtok}},
plural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}},%
description={\the\glslongtok}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glssetattribute{\the\glslabeltok}{regular}{true}}%

```

```

}%
{%
\renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
\renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
\renewcommand*\abbrvpluralsuffix{\glsxtrmsuffix}%
\renewcommand*\glsfirstlongfont[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*\glslongfont[1]{\glslongdefaultfont{##1}}%
\renewcommand*\glsxtrsubsequentfmt[2]{%
  \glslongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
  \ifglsxtrininsertinside \else##2\fi
}%
\renewcommand*\glsxtrsubsequentplfmt[2]{%
  \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
  \ifglsxtrininsertinside \else##2\fi
}%
\renewcommand*\Glsxtrsubsequentfmt[2]{%
  \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
  \ifglsxtrininsertinside \else##2\fi
}%
\renewcommand*\Glsxtrsubsequentplfmt[2]{%
  \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
  \ifglsxtrininsertinside \else##2\fi
}%
\renewcommand*\glsxtrinlinefullformat[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
}%
\renewcommand*\glsxtrinlinefullplformat[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*\Glsxtrinlinefullformat[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
}%
\renewcommand*\Glsxtrinlinefullplformat[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*\glsxtrfullformat[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*\glsxtrfullplformat[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi
}

```

```

}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
}
\@glxtr@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}{\glxtrsmsuffix}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
    \ifglxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
    \ifglxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
    \ifglxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
    \ifglxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
    \glxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  }

```

```

\glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\protect\glsfirstabbrvsmfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glsxtrfullformat}[2]{%
\glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
\glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\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}%
\glsattribute{\the\glslabeltok}{regular}%
{%
\glssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}

```

```

\renewcommand*\glsabbrvfont[1]{\glsabbrvsmfont{##1}}%
\renewcommand*\glsfirstabbrvfont[1]{\glsfirstabbrvsmfont{##1}}%
\renewcommand*\{abbrvpluralsuffix}{\glsxtrmsuffix}%
\renewcommand*\{glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
\renewcommand*\{glslongfont}[1]{\glslongfootnotefont{##1}}%
\renewcommand*\{glsxtrfullformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi
  \protect\glsxtrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*\{glsxtrfullplformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi
  \protect\glsxtrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*\{Glsxtrfullformat}[2]{%
  \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi
  \protect\glsxtrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*\{Glsxtrfullplformat}[2]{%
  \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi
  \protect\glsxtrabbrvfootnote{##1}%
  {\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*\{glsxtrinlinefullformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*\{glsxtrinlinefullplformat}[2]{%
  \glsfirstabbrvsmfont{\glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*\{Glsxtrinlinefullformat}[2]{%
  \glsfirstabbrvsmfont{\Glsaccessshort{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
}%
\renewcommand*\{Glsxtrinlinefullplformat}[2]{%
  \glsfirstabbrvsmfont{\Glsaccessshortpl{##1}\ifglsxtrininsertinside##2\fi}%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongfootnotefont{\glsaccesslongpl{##1}}}%
}%
}

```

```

\@glxtr@deprecated@abbrstyle{footnote-sm}{short-sm-footnote}
\newabbreviationstyle{short-sm-footnote-desc}%
{%
  \glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrfootnotedesname},
    sort={\glxtrfootnotedesort},
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}%
      \protect\glxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}%
      \protect\glxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sm-footnote}%
}
\newabbreviationstyle{short-sm-postfootnote}%
{%
  \glxtrAccSuppAbbrSetNoLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrfootnotename},
    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{glxtrpostlink\glscategorylabel}{%
      \glxtrifwasfirstuse
      {%
        \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}%
          {\glsfirstlongfootnotefont{\glsentrylong{\glslabel}}}}}%
      }%
      {}%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetattribute{\the\glslabeltok}{regular}{false}%
    }%
  }%
}

```



```

    }%
    {}%
  }%
  \renewcommand*{\glstrsetupfulldefs}{%
    \let\glstrifwasfirstuse\@secondoftwo
  }%
}%
{%
  \renewcommand*\glsabrvfont[1]{\glsabrvsmfont{##1}}%
  \renewcommand*\glsfirstabrvfont[1]{\glsfirstabrvsmfont{##1}}%
  \renewcommand*\abbrvpluralsuffix{\glstrsmsuffix}%
  \renewcommand*\glsfirstlongfont[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*\glslongfont[1]{\glslongfootnotefont{##1}}%
  \renewcommand*\glstrfullformat[2]{%
    \glsfirstabrvsmfont{\glssaccessshort{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
  }%
  \renewcommand*\glstrfullplformat[2]{%
    \glsfirstabrvsmfont{\glssaccessshortpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
  }%
  \renewcommand*\Glsxtrfullformat[2]{%
    \glsfirstabrvsmfont{\Glsaccessshort{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
  }%
  \renewcommand*\Glsxtrfullplformat[2]{%
    \glsfirstabrvsmfont{\Glsaccessshortpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
  }%
  \renewcommand*\glstrinlinefullformat[2]{%
    \glsfirstabrvsmfont{\glssaccessshort{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
    \glstrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
  }%
  \renewcommand*\glstrinlinefullplformat[2]{%
    \glsfirstabrvsmfont{\glssaccessshortpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
    \glstrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
  }%
  \renewcommand*\Glsxtrinlinefullformat[2]{%
    \glsfirstabrvsmfont{\Glsaccessshort{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
    \glstrparen{\glsfirstlongfootnotefont{\Glsaccesslong{##1}}}%
  }%
  \renewcommand*\Glsxtrinlinefullplformat[2]{%
    \glsfirstabrvsmfont{\Glsaccessshortpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi\glstrfullsep{##1}%
    \glstrparen{\glsfirstlongfootnotefont{\Glsaccesslongpl{##1}}}%
  }%
}
}

```

```

\@glxtr@deprecated@abbrstyle{postfootnote-sm}{short-sm-postfootnote}
\newabbreviationstyle{short-sm-postfootnote-desc}%
{%
  \glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrfootnotedesname},
    sort={\glxtrfootnotedesort},
    first={\protect\glsfirstabbrvsmfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvsmfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvsmfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvsmfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glxtrpostlink\glscategorylabel}{%
      \glxtrifwasfirstuse
      {%
        \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}%
          {\glsfirstlongfootnotefont{\glstrylong{\glslabel}}}}}%
        }%
      }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    }%
  }%
  \renewcommand*{\glxtrsetupfulldefs}{%
    \let\glxtrifwasfirstuse\@secondoftwo
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-sm-postfootnote}%
}
\newcommand*{\glsabbrvemfont}[1]{\emph{#1}}%
\newcommand*{\glsfirstabbrvemfont}[1]{\glsabbrvemfont{#1}}%
\newcommand*{\glxtremsuffix}{\glxtrabbrvpluralsuffix}
\newcommand*{\glsfirstlongemfont}[1]{\glslongemfont{#1}}%
\newcommand*{\glslongemfont}[1]{\emph{#1}}%
\newabbreviationstyle{long-short-em}%
{%
  \glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstlongdefaultfont{\the\glslongtok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongdefaultfont{\the\glslongpltok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%

```

```

text={\protect\glsabbrvemfont{\the\glsshorttok}},%
plural={\protect\glsabbrvemfont{\the\glsshortpltok}},%
description={\the\glslongtok}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glsattribute{\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}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*\glsxtrfullplformat[2]{%
\glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*\Glsxtrfullformat[2]{%
\glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*\Glsxtrfullplformat[2]{%
\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\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}%

```

```

\glstrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
text={\protect\glsabbrvemfont{\the\glsshorttok}},%
plural={\protect\glsabbrvemfont{\the\glsshortpltok}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glsasattribute{\the\glslabeltok}{regular}%
{%
\glssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{%
\GlsXtrUseAbbrStyleFmts{long-short-em}%
}
\newabbreviationstyle{long-em-short-em}%
{%
\glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glstrlongshortname},
sort={\the\glsshorttok},
first={\protect\glsfirstlongemfont{\the\glslongtok}%
\protect\glstrfullsep{\the\glslabeltok}%
\glstrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}%
\protect\glstrfullsep{\the\glslabeltok}%
\glstrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
text={\protect\glsabbrvemfont{\the\glsshorttok}},%
plural={\protect\glsabbrvemfont{\the\glsshortpltok}},%
description={\protect\glslongemfont{\the\glslongtok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glsasattribute{\the\glslabeltok}{regular}%
{%
\glssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
}%
{%
\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
\renewcommand*{\glstrfullformat}[2]{%
\glsfirstlongemfont{\glsaccesslong{##1}\ifglstrinsertinside##2\fi}%
\ifglstrinsertinside\else##2\fi
\glstrfullsep{##1}%
\glstrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
}

```

```

\renewcommand*{\glxstrfullplformat}[2]{%
  \glsfirstlongemfont{\glssaccesslongpl{##1}\ifglxstrinsertinside##2\fi}%
  \ifglxstrinsertinside\else##2\fi\glxstrfullsep{##1}%
  \glxstrparen{\glsfirstabbrvemfont{\glssaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongemfont{\Glsaccesslong{##1}\ifglxstrinsertinside##2\fi}%
  \ifglxstrinsertinside\else##2\fi\glxstrfullsep{##1}%
  \glxstrparen{\glsfirstabbrvemfont{\glssaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglxstrinsertinside##2\fi}%
  \ifglxstrinsertinside\else##2\fi\glxstrfullsep{##1}%
  \glxstrparen{\glsfirstabbrvemfont{\glssaccessshortpl{##1}}}%
}%
}
\newabbreviationstyle{long-em-short-em-desc}%
{%
  \glxstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxstrlongshortdescname},
    sort={\glxstrlongshortdescsort},%
    first={\protect\glsfirstlongemfont{\the\glslongtok}%
      \protect\glxstrfullsep{\the\glslabeltok}%
      \glxstrparen{\protect\glsfirstabbrvemfont{\the\glsshorttok}}},%
    firstplural={\protect\glsfirstlongemfont{\the\glslongpltok}%
      \protect\glxstrfullsep{\the\glslabeltok}%
      \glxstrparen{\protect\glsfirstabbrvemfont{\the\glsshortpltok}}},%
    text={\protect\glssabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glssabbrvemfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-em-short-em}%
}
\newabbreviationstyle{short-em-long}%
{%
  \glxstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxstrshortlongname},
    sort={\the\glsshorttok},
    description={\the\glslongtok},%
    first={\protect\glsfirstabbrvemfont{\the\glsshorttok}}%
  }%
}

```

```

\protect\glxtrfullsep{\the\glslabeltok}%
\glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongtok}}},%
firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
\protect\glxtrfullsep{\the\glslabeltok}%
\glxtrparen{\protect\glsfirstlongdefaultfont{\the\glslongpltok}}},%
text={\protect\glsabbrvemfont{\the\glsshorttok}}},%
plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{%
\renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongdefaultfont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongdefaultfont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongdefaultfont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrininsertinside##2\fi}%
\ifglxtrininsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongdefaultfont{\glssaccesslongpl{##1}}}%
}%
}%
\newabbreviationstyle{short-em-long-desc}%
{%
\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrshortlongdescname},
sort={\glxtrshortlongdescsort},

```

```

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}{%
\glsasattribute{\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}{%
\glsasattribute{\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}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongemfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
\glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongemfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
\glsxtrparen{\glsfirstlongemfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\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}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
\newabbreviationstyle{short-em}%
{%
\GlsXtrUseAbbrStyleFmts{short-em-long-em}%
}
\newabbreviationstyle{short-em}%
{%

```



```

\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}{\glxtremsuffix}%
\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}}%
  \ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\glsxtrinlinefullplformat}[2]{%
  \protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}%
  \ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \protect\glsfirstabbrvemfont{\Glsaccessshort{##1}}%
  \ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \protect\glsfirstabbrvemfont{\Glsaccessshortpl{##1}}%
  \ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
  \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
}%
\renewcommand*{\glsxtrfullformat}[2]{%
  \glsfirstabbrvemfont{\glsaccessshort{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
  \glsfirstabbrvemfont{\glsaccessshortpl{##1}}\ifglsxtrininsertinside##2\fi%
  \ifglsxtrininsertinside\else##2\fi
}%

```

```

\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\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}{\glxtremsuffix}%
  \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}\ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
    \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
    \ifglxtrinsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\glsfirstlongdefaultfont{\glsaccesslongpl{##1}}}%
  }%
  \renewcommand*{\glsxtrfullformat}[2]{%

```

```

\glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvemfont{\glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvemfont{\glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
}
\letabbreviationstyle{short-em-nolong-desc}{short-em-desc}
\newabbreviationstyle{nolong-short-em}%
{%
\GlsXtrUseAbbrStyleSetup{short-em-nolong}%
}%
{%
\GlsXtrUseAbbrStyleFmts{short-em-nolong}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\protect\glsfirstlongdefaultfont{\glsaccesslong{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\protect\glsfirstlongdefaultfont{\glsaccesslongpl{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\protect\glsfirstlongdefaultfont{\Glsaccesslong{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\protect\glsfirstlongdefaultfont{\Glsaccesslongpl{##1}%
\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\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}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
  }%
}

```

```

\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvemfont{\Glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
}
\@glxtr@deprecated@abbrstyle{long-em}{long-noshort-em}
\newabbreviationstyle{long-em-noshort-em}%
{%
  \glxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrlongnoshortname},
    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}{%
    \glsssetAttribute{\the\glslabeltok}{regular}{true}}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glxtremsuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
  \renewcommand*{\glxtrsubsequentfmt}[2]{%
    \glslongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
    \ifglxtrinsertinside \else##2\fi
  }%
  \renewcommand*{\glxtrsubsequentplfmt}[2]{%
    \glslongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%

```

```

\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
\glslongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
\glslongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\glsfirstlongemfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstlongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstlongemfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstlongemfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
}
\newabbreviationstyle{long-em-noshort-em-noreg}%
{%
\glxtrAccSuppAbbrSetNameShortAttrs\glscategorylabel

```

```

\GlsXtrUseAbbrStyleSetup{long-em-noshort-em}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-em-noshort-em}%
}
\newabbreviationstyle{long-noshort-em-desc}%
{%
  \GlsXtrUseAbbrStyleSetup{long-noshort-desc}%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrmsuffix}%
  \renewcommand*{\glssabrvfont}[1]{\glssabrvemfont{##1}}%
  \renewcommand*{\glsfirstabrvfont}[1]{\glsfirstabrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongdefaultfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongdefaultfont{##1}}%
  \renewcommand*{\glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\glssaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\glssaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
    \glslongdefaultfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside ##2\fi}%
    \ifglsxtrininsertinside \else##2\fi
  }%
  \renewcommand*{\glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\glssaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabrvemfont{\glssaccessshort{##1}}}%
  }%
  \renewcommand*{\glsxtrinlinefullplformat}[2]{%
    \glsfirstlongdefaultfont{\glssaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi\glsxtrfullsep{##1}%
    \glsxtrparen{\protect\glsfirstabrvemfont{\glssaccessshortpl{##1}}}%
  }%
  \renewcommand*{\Glsxtrinlinefullformat}[2]{%
    \glsfirstlongdefaultfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%

```

```

        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
    }%
    \renewcommand*{\Glsxtrinlinefullplformat}[2]{%
        \glsfirstlongdefaultfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
        \glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
    }%
    \renewcommand*{\glxtrfullformat}[2]{%
        \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\glxtrfullplformat}[2]{%
        \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullformat}[2]{%
        \glsfirstlongdefaultfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi
    }%
    \renewcommand*{\Glsxtrfullplformat}[2]{%
        \glsfirstlongdefaultfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
        \ifglxtrinsertinside\else##2\fi
    }%
}
\@glxtr@deprecated@abbrstyle{long-desc-em}{long-noshort-em-desc}
\newabbreviationstyle{long-em-noshort-em-desc}%
{%
    \renewcommand*{\CustomAbbreviationFields}{%
        name={\glxtrlongnoshortdescname},
        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}{\glxtremsuffix}%
    \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
    \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
    \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongemfont{##1}}%
    \renewcommand*{\glslongfont}[1]{\glslongemfont{##1}}%
    \renewcommand*{\glxtrsubsequentfmt}[2]{%
        \glslongemfont{\glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
        \ifglxtrinsertinside \else##2\fi
    }%
    \renewcommand*{\glxtrsubsequentplfmt}[2]{%

```



```

\glslongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentfmt}[2]{%
\glslongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\Glsxtrsubsequentplfmt}[2]{%
\glslongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside ##2\fi}%
\ifglxtrinsertinside \else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\glsfirstlongemfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstlongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshort{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvemfont{\glsaccessshortpl{##1}}}%
}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstlongemfont{\glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstlongemfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstlongemfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstlongemfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
}
\newabbreviationstyle{long-em-noshort-em-desc-noreg}%
{%

```

```

\GlsXtrUseAbbrStyleSetup{long-em-noshort-em-desc}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glissetattribute{\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\glxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvemfont{\the\glsshortpltok}%
      \protect\glxtrabbrvfootnote{\the\glslabeltok}%
      {\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%
    text={\protect\glsabbrvemfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glissetattribute{\the\glslabeltok}{nohyperfirst}{true}%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glissetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glxtrremsuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvemfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvemfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongfootnotefont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongfootnotefont{##1}}%
  \renewcommand*{\glxtrfullformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshort{##1}}\ifglxtrininsertinside##2\fi}%
    \ifglxtrininsertinside\else##2\fi
  \protect\glxtrabbrvfootnote{##1}%
    {\glsfirstlongfootnotefont{\glsaccesslong{##1}}}%
  }%
  \renewcommand*{\glxtrfullplformat}[2]{%
    \glsfirstabbrvemfont{\glsaccessshortpl{##1}}\ifglxtrininsertinside##2\fi}%

```

```

\ifglxtrinsertinside\else##2\fi
\protect\glxtrabbrvfootnote{##1}%
{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\protect\glxtrabbrvfootnote{##1}%
{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\protect\glxtrabbrvfootnote{##1}%
{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstabbrvemfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
\glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
}
\@glxtr@deprecated@abbrstyle{footnote-em}{short-em-footnote}
\newabbreviationstyle{short-em-footnote-desc}{%
{%
\glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtrfootnotedesname},
sort={\glxtrfootnotedesort},
first={\protect\glsfirstabbrvemfont{\the\glssshorttok}%
\protect\glxtrabbrvfootnote{\the\glslabeltok}%
{\protect\glsfirstlongfootnotefont{\the\glslongtok}}},%
firstplural={\protect\glsfirstabbrvemfont{\the\glssshortpltok}%
\protect\glxtrabbrvfootnote{\the\glslabeltok}%
{\protect\glsfirstlongfootnotefont{\the\glslongpltok}}},%

```

```

text={\protect\glsabbrvemfont{\the\glsshorttok}},%
plural={\protect\glsabbrvemfont{\the\glsshortpltok}}}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\glssetattribute{\the\glslabeltok}{nohyperfirst}{true}%
\glsattribute{\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}}}}%
}%
}%
}%
\glsattribute{\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}}%
\renewcommand*{\glslongfont}[1]{\glsfirstlongfootnotefont{##1}}%

```

```

\renewcommand*{\glxtrfullformat}[2]{%
  \glsfirstabbrvemfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glsfirstabbrvemfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
  \glsfirstabbrvemfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glsfirstabbrvemfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstabbrvemfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslong{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstabbrvemfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi\glxtrfullsep{##1}%
  \glxtrparen{\glsfirstlongfootnotefont{\glssaccesslongpl{##1}}}%
}%
}
\@glxtr@deprecated@abbrstyle{postfootnote-em}{short-em-postfootnote}
\newabbreviationstyle{short-em-postfootnote-desc}%
{%
  \glxtrAccSuppAbbrSetNameLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrfootnotedesname},
    sort={\glxtrfootnotedesort},
    first={\protect\glsfirstabbrvemfont{\the\glssshorttok}},%
    firstplural={\protect\glsfirstabbrvemfont{\the\glssshortpltok}},%
    text={\protect\glsabbrvemfont{\the\glssshorttok}},%
    plural={\protect\glsabbrvemfont{\the\glssshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glxtrpostlink\glscategorylabel}{%

```

```

\glxtrifwasfirstuse
{%
  \glxtrdopostpunc{\protect\glxtrabbrvfootnote{\glslabel}%
    {\glslfirstlongfootnotefont{\glsentrylong{\glslabel}}}}%
 }%
}%
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
  \glsssetAttribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
\renewcommand*{\glxtrsetupfulldefs}{%
  \let\glxtrifwasfirstuse\@secondoftwo
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-em-postfootnote}%
}
\newcommand*{\glxtruserfield}{useri}
\ifdef\glscurrentfieldvalue
{
  \newcommand*{\glxtruserparen}[2]{%
    \glxtrfullsep{#2}%
    \glxtrparen
    {#1\ifglshasfield{\glxtruserfield}{#2}{, \glscurrentfieldvalue}{}}%
  }
}
{
  \newcommand*{\glxtruserparen}[2]{%
    \glxtrfullsep{#2}%
    \glxtrparen
    {#1\ifglshasfield{\glxtruserfield}{#2}{, \@glo@thisvalue}{}}%
  }
}
\newcommand*{\glsabbrvuserfont}[1]{\glsabbrvdefaultfont{#1}}
\newcommand*{\glslfirstabbrvuserfont}[1]{\glsabbrvuserfont{#1}}
\newcommand*{\glslonguserfont}[1]{\glslongdefaultfont{#1}}
\newcommand*{\glslfirstlonguserfont}[1]{\glslonguserfont{#1}}
\newcommand*{\glxtrusersuffix}{\glxtrabbrvpluralsuffix}
\newcommand*{\gluserdescription}[2]{\glslonguserfont{#1}}
\newabbreviationstyle{long-short-user}%
{%
  \glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrlongshortname},
    sort={\the\glsshorttok},
    first={\protect\glslfirstlonguserfont{\the\glslongtok}%
      \protect\glxtruserparen{\protect\glslfirstabbrvuserfont{\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}{%
\glsasattribute{\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}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
\glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
}%
\renewcommand*{\glsxtrfullplformat}[2]{%
\glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
\glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\else##2\fi
\glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
\ifglsxtrininsertinside\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}}%
    }%
  }%
}%
\glsattribute{\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}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrinelinefullformat}[2]{%
    \glsfirstlonguserfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
    \glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
  }%
  \renewcommand*{\glsxtrinelinefullplformat}[2]{%

```



```

\glsfirstlonguserfont{\glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshortpl{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glsxtruserparen{\glsfirstabbrvuserfont{\glsaccessshort{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\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}%
}%
}%
}%
}%
}

```

```

{%
\renewcommand*{\abbrvpluralsuffix}{\glxtrscusersuffix}%
\renewcommand*{\glsabbrvfont}[1]{\glsabbrvcuserfont{##1}}%
\renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvcuserfont{##1}}%
\renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonguserfont{##1}}%
\renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
\renewcommand*{\glxtrfullformat}[2]{%
\glsfirstlonguserfont{\glssaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
\glsfirstlonguserfont{\glssaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrininlinefullformat}[2]{%
\glsfirstlonguserfont{\glssaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtruserparen{\glsfirstabbrvcuserfont{\glssaccessshort{##1}}}{##1}%
}%
\renewcommand*{\glxtrininlinefullplformat}[2]{%
\glsfirstlonguserfont{\glssaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtruserparen{\glsfirstabbrvcuserfont{\glssaccessshortpl{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrininlinefullformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtruserparen{\glsfirstabbrvcuserfont{\glssaccessshort{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrininlinefullplformat}[2]{%
\glsfirstlonguserfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtruserparen{\glsfirstabbrvcuserfont{\glssaccessshortpl{##1}}}{##1}%
}%
}
\newcommand*{\glxtrlongshortuserdescname}{%
\protect\glslonguserfont{\the\glslongtok}%
\protect\glxtruserparen
{\protect\glsabbrvuserfont{\the\glssshorttok}}{\the\glslabeltok}%
}
\newabbreviationstyle{long-postshort-user-desc}%
{%

```

```

\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
  name={\glxtrlongshortuserdescname},
  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{glxtrpostlink\glscategorylabel}{%
    \glxtrifwasfirstuse
    {%
      \glxtruserparen
      {\glsfirstabbrvuserfont{\glstryshort{\glslabel}}}%
      {\glslabel}%
    }%
  }%
  {}%
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
  \glsssetattribute{\the\glslabeltok}{regular}{false}%
}%
{}%
}%
{}%
\GlsXtrUseAbbrStyleFmts{long-postshort-user}%
}
\newcommand*{\glxtrlongshortscuserdescname}{%
  \protect\glslonguserfont{\the\glslongtok}%
  \protect\glxtruserparen
  {\protect\glsabbrvscuserfont{\the\glsshorttok}}{\the\glslabeltok}%
}
\newabbreviationstyle{long-postshort-sc-user-desc}%
{%
  \glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrlongshortscuserdescname},
    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{glxtrpostlink\glscategorylabel}{%
      \glxtrifwasfirstuse
      {%
        \glxtruserparen

```

```

        {\glsfirstabbrvscuserfont{\glsentryshort{\glslabel}}}%
        {\glslabel}%
    }%
    {}%
}%
\glsasattribute{\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}%
            }%
            {}%
        }%
        \glsasattribute{\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*{\glxtrfullformat}[2]{%
  \glsfirstabbrvuserfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrfullplformat}[2]{%
  \glsfirstabbrvuserfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
  \glsfirstabbrvuserfont{\glssaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtruserparen{\glsfirstlonguserfont{\glssaccesslong{##1}}}{##1}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glsfirstabbrvuserfont{\glssaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtruserparen{\glsfirstlonguserfont{\glssaccesslongpl{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstabbrvuserfont{\Glsaccessshort{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtruserparen{\glsfirstlonguserfont{\glssaccesslong{##1}}}{##1}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstabbrvuserfont{\Glsaccessshortpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtruserparen{\glsfirstlonguserfont{\glssaccesslongpl{##1}}}{##1}%
}%
}
\newcommand*{\glxtrshortlonguserdescname}{%
  \protect\glssabbrvuserfont{\the\glssshorttok}%
  \protect\glxtruserparen
    {\protect\glsslonguserfont{\the\glsslongpltok}}%
    {\the\glsslabeltok}%
}
\newabbreviationstyle{short-postlong-user-desc}%
{%
  \glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrshortlonguserdescname},
    sort={\the\glssshorttok},
    first={\protect\glsfirstlonguserfont{\the\glsslongtok}},%

```

```

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}}%
}%
}%
\glsasattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\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}{%
\glsasattribute{\the\glslabeltok}{regular}%
{%
\glsssetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{\%
\GlsXtrUseAbbrStyleFmts{long-short-user}%
}

```

```

\newabbreviationstyle{short-long-user}%
{%
  \glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glstrshortlongname},
    sort={\the\glsshorttok},
    description={\protect\glstruserdescription{\the\glslongtok}%
      {\the\glslabeltok}},%
    first={\protect\glstrfirstabbrvuserfont{\the\glsshorttok}%
      \protect\glstrxtruserparen{\protect\glstrfirstlonguserfont{\the\glslongtok}}%
      {\the\glslabeltok}},%
    firstplural={\protect\glstrfirstabbrvuserfont{\the\glsshortpltok}%
      \protect\glstrxtruserparen{\protect\glstrfirstlonguserfont{\the\glslongpltok}}%
      {\the\glslabeltok}},%
    text={\protect\glstrabbrvuserfont{\the\glsshorttok}},%
    plural={\protect\glstrabbrvuserfont{\the\glsshortpltok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glstrusersuffix}%
  \renewcommand*{\glstrabbrvfont}[1]{\glstrabbrvuserfont{##1}}%
  \renewcommand*{\glstrfirstabbrvfont}[1]{\glstrfirstabbrvuserfont{##1}}%
  \renewcommand*{\glstrfirstlongfont}[1]{\glstrfirstlonguserfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonguserfont{##1}}%
  \renewcommand*{\glstrfullformat}[2]{%
    \glstrfirstabbrvuserfont{\glstraccessshort{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
    \glstrxtruserparen{\glstrfirstlonguserfont{\glstraccesslong{##1}}}{##1}%
  }%
  \renewcommand*{\glstrfullplformat}[2]{%
    \glstrfirstabbrvuserfont{\glstraccessshortpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
    \glstrxtruserparen{\glstrfirstlonguserfont{\glstraccesslongpl{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrfullformat}[2]{%
    \glstrfirstabbrvuserfont{\glstraccessshort{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
    \glstrxtruserparen{\glstrfirstlonguserfont{\glstraccesslong{##1}}}{##1}%
  }%
  \renewcommand*{\Glsxtrfullplformat}[2]{%
    \glstrfirstabbrvuserfont{\glstraccessshortpl{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
    \glstrxtruserparen{\glstrfirstlonguserfont{\glstraccesslongpl{##1}}}{##1}%
  }%
}%

```

```

}
\newabbreviationstyle{short-long-user-desc}%
{%
  \glstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glstrshortlonguserdescname},
    sort={\glstrshortlongdescsort},%
    first={\protect\glstrfirstabbrvuserfont{\the\glsshorttok}%
      \protect\glstruserparen{\protect\glstrfirstlonguserfont{\the\glslongtok}}%
      {\the\glslabeltok}},%
    firstplural={\protect\glstrfirstabbrvuserfont{\the\glsshortpltok}%
      \protect\glstruserparen{\protect\glstrfirstlonguserfont{\the\glslongpltok}}%
      {\the\glslabeltok}},%
    text={\protect\glstrabbrvfont{\the\glsshorttok}},%
    plural={\protect\glstrabbrvfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glissetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-long-user}%
}
\newrobustcmd*{\glstrifhyphenstart}[3]{%
  \ifx\glstrinsert#1\relax
    \expandafter\@glstrifhyphenstart#1\relax\relax
    \@end@glstrifhyphenstart{#2}{#3}%
  \else
    \@glstrifhyphenstart#1\relax\relax\@end@glstrifhyphenstart{#2}{#3}%
  \fi
}
\def\@glstrifhyphenstart#1#2\@end@glstrifhyphenstart#3#4{%
  \ifx-#1\relax#3\else #4\fi
}
\newcommand*{\glstrlonghyphenshort}[4]{%
  {%
    \glstrifhyphenstart{#4}{\def\glstrwordsep{-}}{}%
    \glstrfirstlonghyphenfont{#2\ifglstrinsertinside{#4}\fi}%
    \ifglstrinsertinside\else{#4}\fi
    \glstrfullsep{#1}%
    \glstrparen{\glstrfirstabbrvhyphenfont{#3\ifglstrinsertinside{#4}\fi}%
      \ifglstrinsertinside\else{#4}\fi}%
  }%
}
\newcommand*{\glstrabbrvhyphenfont}{\glstrabbrvdefaultfont}%
\newcommand*{\glstrfirstabbrvhyphenfont}{\glstrabbrvhyphenfont}%

```



```

\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={\glxstrlongshortdescname},
  sort={\glxstrlongshortdescsort},
  first={\protect\glsfirstlonghyphenfont{\the\glslongtok}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \glxstrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}}},%
  firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}%
    \protect\glxstrfullsep{\the\glslabeltok}%
    \glxstrparen{\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}}},%
  text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
  plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}}
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%
}
\newcommand*{\glxstrlonghyphennoshort}[3]{%
  {%
    \glxstrifhyphenstart{#3}{\def\glxtrwordsep{-}}{}%
    \glsfirstlonghyphenfont{#2\ifglxtrininsertinside{#3}\fi}%
    \ifglxtrininsertinside\else{#3}\fi
  }%
}
\newabbreviationstyle{long-hyphen-noshort-desc-noreg}%
{%
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxstrlongnoshortdescname},
    sort={\expandonce\glxtrorglong},
    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}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-short-hyphen}%
}

```

```

\renewcommand*{\abbrvpluralsuffix}{\glxtrabbrvpluralsuffix}%
\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}%
{%
  \glstrAccSuppAbbrSetNameShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glstrlongnoshortname},
    sort={\the\glsshorttok},
    first={\protect\glstrfirstlonghyphenfont{\the\glslongtok}},%
    firstplural={\protect\glstrfirstlonghyphenfont{\the\glslongpltok}},%
    text={\protect\glslonghyphenfont{\the\glslongtok}},%
    plural={\protect\glslonghyphenfont{\the\glslongpltok}},%
    description={\the\glslongtok}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-noshort-desc-noreg}%
}
\newcommand*{\glstrlonghyphen}[3]{%
  {%
    \glstrifhyphenstart{#3}{\def\glstrwordsep{-}}{}%
    \glstrfirstlonghyphenfont{#1}%
  }%
}
\newcommand*{\glstrposthyphenshort}[2]{%
  {%
    \glstrifhyphenstart{#2}{\def\glstrwordsep{-}}{}%
    \ifglstrinsertinside{\glstrfirstlonghyphenfont{#2}}\else{#2}\fi
    \glstrfullsep{#1}%
    \glstrparen
    {\glstrfirstabbrvhyphenfont{\glstryshort{#1}}\ifglstrinsertinside{#2}\fi}%
    \ifglstrinsertinside\else{#2}\fi
  }%
}%
}
\newcommand*{\glstrposthyphensubsequent}[2]{%
  \glstrabbrvfont{\ifglstrinsertinside {#2}\fi}%
  \ifglstrinsertinside \else{#2}\fi
}
\newabbreviationstyle{long-hyphen-postshort-hyphen}%
{%
  \glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glstrlongshortname},
    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}%
}%
}%
\glsattribute{\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]{%

```

```

\glxstrlonghyphen{\Glsaccesslongpl{##1}}{##1}{##2}%
}%
\renewcommand*{\glxstrinlinefullformat}[2]{%
\glsfirstlonghyphenfont{\Glsaccesslong{##1}%
\ifglxstrinsertinside{##2}\fi}%
\ifglxstrinsertinside \else{##2}\fi
}%
\renewcommand*{\glxstrinlinefullplformat}[2]{%
\glsfirstlonghyphenfont{\Glsaccesslongpl{##1}%
\ifglxstrinsertinside{##2}\fi}%
\ifglxstrinsertinside \else{##2}\fi
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstlonghyphenfont{\Glsaccesslong{##1}%
\ifglxstrinsertinside{##2}\fi}%
\ifglxstrinsertinside \else{##2}\fi
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstlonghyphenfont{\Glsaccesslongpl{##1}%
\ifglxstrinsertinside{##2}\fi}%
\ifglxstrinsertinside \else{##2}\fi
}%
}
\newabbreviationstyle{long-hyphen-postshort-hyphen-desc}%
{%
\glxstrAccSuppAbbrSetTextShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxstrlongshortdescname},
sort={\glxstrlongshortdescsort},%
first={\protect\glsfirstlonghyphenfont{\the\glslongtok}},%
firstplural={\protect\glsfirstlonghyphenfont{\the\glslongpltok}},%
text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
\csdef{glxstrpostlink\glscategorylabel}{%
\glxstrifwasfirstuse
{%
\glxstrposthyphenshort{\glslabel}{\glsinsert}%
}%
{%
\glxstrposthyphensubsequent{\glslabel}{\glsinsert}%
}%
}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
\glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%

```

```

}%
{%
  \GlsXtrUseAbbrStyleFmts{long-hyphen-postshort-hyphen}%
}
\newcommand*{\glxtrshorthyphenlong}[4]{%
  {%
    \glxtrifhyphenstart{#4}{\def\glxtrwordsep{-}}{}%
    \glsfirstabbrvhyphenfont{#2\ifglxtrininsertinside{#4}\fi}%
    \ifglxtrininsertinside\else{#4}\fi
    \glxtrfullsep{#1}%
    \glxtrparen{\glsfirstlonghyphenfont{#3\ifglxtrininsertinside{#4}\fi}%
      \ifglxtrininsertinside\else{#4}\fi}%
  }%
}
\newabbreviationstyle{short-hyphen-long-hyphen}%
{%
  \glxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrshortlongname},
    sort={\the\glsshorttok},
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%
    text={\protect\glssabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glssabbrvhyphenfont{\the\glsshortpltok}},%
    description={\protect\glslonghyphenfont{\the\glslongtok}}}
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glxtrhyphensuffix}%
  \renewcommand*{\glssabbrvfont}[1]{\glssabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlonghyphenfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
  \renewcommand*{\glxtrfullformat}[2]{%
    \glxtrshorthyphenlong{##1}{\glssaccessshort{##1}}{\glssaccesslong{##1}}{##2}%
  }%
  \renewcommand*{\glxtrfullplformat}[2]{%
    \glxtrshorthyphenlong{##1}%
    {\glssaccessshortpl{##1}}{\glssaccesslongpl{##1}}{##2}%
  }%
}

```

```

\renewcommand*{\Glsxtrfullformat}[2]{%
  \glxtrshorthyphenlong{##1}{\glsaccessshort{##1}}{\Glsaccesslong{##1}}{##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glxtrshorthyphenlong{##1}%
  {\glsaccessshortpl{##1}}{\Glsaccesslongpl{##1}}{##2}%
}%
}
\newabbreviationstyle{short-hyphen-long-hyphen-desc}%
{%
  \glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrshortlongdescname},
    sort={\glxtrshortlongdescsort},
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongtok}}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}%
      \protect\glxtrfullsep{\the\glslabeltok}%
      \glxtrparen{\protect\glsfirstlonghyphenfont{\the\glslongpltok}}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glsattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
\GlsXtrUseAbbrStyleFmts{short-hyphen-long-hyphen}%
}
\newcommand*{\glxtrshorthyphen}[3]{%
  {%
    \glxtrifhyphenstart{#3}{\def\glxtrwordsep{-}}{}%
    \glsfirstabbrvhyphenfont{#1}%
  }%
}
\newcommand*{\glxtrposthyphenlong}[2]{%
  {%
    \glxtrifhyphenstart{#2}{\def\glxtrwordsep{-}}{}%
    \ifglxtrininsertinside{\glsfirstabbrvhyphenfont{#2}}\else{#2}\fi
    \glxtrfullsep{#1}%
    \glxtrparen
    {\glsfirstlonghyphenfont{\glsentrylong{#1}\ifglxtrininsertinside{#2}\fi}%
    \ifglxtrininsertinside\else{#2}\fi
  }%
}%

```



```

}
\newabbreviationstyle{short-hyphen-postlong-hyphen}%
{%
  \glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glstrshortlongname},
    sort={\the\glsshorttok},
    first={\protect\glfirstabbrvhyphenfont{\the\glsshorttok}},%
    firstplural={\protect\glfirstabbrvhyphenfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}},%
    description={\protect\glslonghyphenfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glstrpostlink\glscategorylabel}{%
      \glstrifwasfirstuse
      {%
        \glstrposthyphenlong{\glslabel}{\glsinsert}%
      }%
      {%
        \glstrposthyphensubsequent{\glslabel}{\glsinsert}%
      }%
    }%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glissetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glstrabbrvpluralsuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvhyphenfont{##1}}%
  \renewcommand*{\glfirstabbrvfont}[1]{\glfirstabbrvhyphenfont{##1}}%
  \renewcommand*{\glfirstlongfont}[1]{\glfirstlonghyphenfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslonghyphenfont{##1}}%
  \renewcommand*{\glstrsubsequentfmt}[2]{%
    \glsabbrvfont{\glsaccessshort{##1}}%
  }%
  \renewcommand*{\glstrsubsequentplfmt}[2]{%
    \glsabbrvfont{\glsaccessshortpl{##1}}%
  }%
  \renewcommand*{\Glsstrsubsequentfmt}[2]{%
    \glsabbrvfont{\Glsaccessshort{##1}}%
  }%
  \renewcommand*{\Glsstrsubsequentplfmt}[2]{%
    \glsabbrvfont{\Glsaccessshortpl{##1}}%
  }%
  \renewcommand*{\glstrfullformat}[2]{%
    \glstrshorthyphen{\glsaccessshort{##1}}{##1}{##2}%
  }%
}

```

```

\renewcommand*{\glxtrfullplformat}[2]{%
  \glxtrshorthyphen{\glsaccessshortpl{##1}}{##1}{##2}%
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
  \glxtrshorthyphen{\Glsaccessshort{##1}}{##1}{##2}%
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glxtrshorthyphen{\Glsaccessshortpl{##1}}{##1}{##2}%
}%
\renewcommand*{\glxtrininlinefullformat}[2]{%
  \glsfirstabbrvhyphenfont{\glsaccessshort{##1}}%
  \ifglxtrininsertinside{##2}\fi}%
  \ifglxtrininsertinside \else{##2}\fi
}%
\renewcommand*{\glxtrininlinefullplformat}[2]{%
  \glsfirstabbrvhyphenfont{\glsaccessshortpl{##1}}%
  \ifglxtrininsertinside{##2}\fi}%
  \ifglxtrininsertinside \else{##2}\fi
}%
\renewcommand*{\Glsxtrininlinefullformat}[2]{%
  \glsfirstabbrvhyphenfont{\Glsaccessshort{##1}}%
  \ifglxtrininsertinside{##2}\fi}%
  \ifglxtrininsertinside \else{##2}\fi
}%
\renewcommand*{\Glsxtrininlinefullplformat}[2]{%
  \glsfirstabbrvhyphenfont{\Glsaccessshortpl{##1}}%
  \ifglxtrininsertinside{##2}\fi}%
  \ifglxtrininsertinside \else{##2}\fi
}%
}
\newabbreviationstyle{short-hyphen-postlong-hyphen-desc}%
{%
  \glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrshortlongdescname},%
    sort={\glxtrshortlongdescsort},%
    first={\protect\glsfirstabbrvhyphenfont{\the\glsshorttok}},%
    firstplural={\protect\glsfirstabbrvhyphenfont{\the\glsshortpltok}},%
    text={\protect\glsabbrvhyphenfont{\the\glsshorttok}},%
    plural={\protect\glsabbrvhyphenfont{\the\glsshortpltok}}%
  }%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \csdef{glxtrpostlink\glscategorylabel}{%
      \glxtrifwasfirstuse
      {%
        \glxtrposthyphenlong{\glslabel}{\glsinsert}%
      }%
      {%
        \glxtrposthyphensubsequent{\glslabel}{\glsinsert}%
      }%
    }%
  }%
}

```

```

}%
\glshasattribute{\the\glslabeltok}{regular}%
{%
  \glissetattribute{\the\glslabeltok}{regular}{false}%
}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{short-hyphen-postlong-hyphen}%
}
\newcommand*{\glsabbrvonlyfont}{\glsabbrvdefaultfont}%
\newcommand*{\glsfirstabbrvonlyfont}{\glsabbrvonlyfont}%
\newcommand*{\glslongonlyfont}{\glslongdefaultfont}%
\newcommand*{\glsfirstlongonlyfont}{\glslongonlyfont}%
\newcommand*{\glstronlysuffix}{\glstrabbrvpluralsuffix}%
\newcommand*{\glstronlyname}{%
  \protect\glsabbrvonlyfont{\the\glsshorttok}%
}
\newabbreviationstyle{long-only-short-only}%
{%
  \glstrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glstronlyname},
    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}%
    {%
      \glissetattribute{\the\glslabeltok}{regular}{false}%
    }%
    {%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glstronlysuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvonlyfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvonlyfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glslongonlyfont{##1}}%
  \renewcommand*{\glstrfullformat}[2]{%
    \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglstrinsertinside##2\fi}%
    \ifglstrinsertinside\else##2\fi
  }%
  \renewcommand*{\glstrfullplformat}[2]{%
    \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglstrinsertinside##2\fi}%

```

```

\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullformat}[2]{%
\glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
\glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
\glsfirstlongonlyfont{\glssaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvonlyfont{\glssaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
\glsfirstlongonlyfont{\glssaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvonlyfont{\glssaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
\glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvonlyfont{\glssaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
\glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
\ifglxtrinsertinside\else##2\fi
\glxtrfullsep{##1}%
\glxtrparen{\protect\glsfirstabbrvonlyfont{\glssaccessshortpl{##1}}}%
}%
}
\newcommand*{\glxtronlydescsort}{\the\glslongtok}
\newcommand*{\glxtronlydescname}{%
\protect\glslongfont{\the\glslongtok}%
}
\newabbreviationstyle{long-only-short-only-desc}%
{%
\glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
\renewcommand*{\CustomAbbreviationFields}{%
name={\glxtronlydescname},
sort={\glxtronlydescsort},%
first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
text={\protect\glssabbrvonlyfont{\the\glssshorttok}},%
plural={\protect\glssabbrvonlyfont{\the\glssshortpltok}}}%
}%

```

```

\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }%
  {}%
}%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-only-short-only}%
}
\newcommand*{\glsabbrvsconlyfont}{\glsabbrvscfont}%
\newcommand*{\glsfirstabbrvsconlyfont}{\glsabbrvsconlyfont}%
\newcommand*{\glsxtrscnlysuffix}{\glsxtrscsuffix}
\newcommand*{\glsxtrscnlyname}{%
  \protect\glsabbrvsconlyfont{\the\glsshorttok}%
}
\newabbreviationstyle{long-only-short-sc-only}%
{%
  \glsxtrAccSuppAbbrSetFirstLongAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glsxtrscnlyname},
    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\glsfirstlongonlyfont{\the\glslongtok}}}%
  \renewcommand*{\GlsXtrPostNewAbbreviation}{%
    \glshasattribute{\the\glslabeltok}{regular}%
    {%
      \glsssetAttribute{\the\glslabeltok}{regular}{false}%
    }%
    {}%
  }%
}%
{%
  \renewcommand*{\abbrvpluralsuffix}{\glsxtrscnlysuffix}%
  \renewcommand*{\glsabbrvfont}[1]{\glsabbrvsconlyfont{##1}}%
  \renewcommand*{\glsfirstabbrvfont}[1]{\glsfirstabbrvsconlyfont{##1}}%
  \renewcommand*{\glsfirstlongfont}[1]{\glsfirstlongonlyfont{##1}}%
  \renewcommand*{\glslongfont}[1]{\glsfirstlongonlyfont{##1}}%
  \renewcommand*{\glsxtrfullformat}[2]{%
    \glsfirstlongonlyfont{\glsaccesslong{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
  }%
  \renewcommand*{\glsxtrfullplformat}[2]{%
    \glsfirstlongonlyfont{\glsaccesslongpl{##1}\ifglsxtrininsertinside##2\fi}%
    \ifglsxtrininsertinside\else##2\fi
  }%
}

```

```

\renewcommand*{\Glsxtrfullformat}[2]{%
  \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\Glsxtrfullplformat}[2]{%
  \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
}%
\renewcommand*{\glxtrinlinefullformat}[2]{%
  \glsfirstlongonlyfont{\glssaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvsconlyfont{\glssaccessshort{##1}}}%
}%
\renewcommand*{\glxtrinlinefullplformat}[2]{%
  \glsfirstlongonlyfont{\glssaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvsconlyfont{\glssaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullformat}[2]{%
  \glsfirstlongonlyfont{\Glsaccesslong{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvsconlyfont{\glssaccessshortpl{##1}}}%
}%
\renewcommand*{\Glsxtrinlinefullplformat}[2]{%
  \glsfirstlongonlyfont{\Glsaccesslongpl{##1}\ifglxtrinsertinside##2\fi}%
  \ifglxtrinsertinside\else##2\fi
  \glxtrfullsep{##1}%
  \glxtrparen{\protect\glsfirstabbrvsconlyfont{\Glsaccessshortpl{##1}}}%
}%
}
\newcommand*{\glxtrsconlydescsort}{\glxtronlydescsort}
\newcommand*{\glxtrsconlydescname}{\glxtronlydescname}
\newabbreviationstyle{long-only-short-sc-only-desc}%
{%
  \glxtrAccSuppAbbrSetTextShortAttrs\glscategorylabel
  \renewcommand*{\CustomAbbreviationFields}{%
    name={\glxtrsconlydescname},
    sort={\glxtrsconlydescsort},%
    first={\protect\glsfirstlongonlyfont{\the\glslongtok}},%
    firstplural={\protect\glsfirstlongonlyfont{\the\glslongpltok}},%
    text={\protect\glssabbrvsconlyfont{\the\glssshorttok}},%
    plural={\protect\glssabbrvsconlyfont{\the\glssshortpltok}}}%
}%
\renewcommand*{\GlsXtrPostNewAbbreviation}{%
  \glshasattribute{\the\glslabeltok}{regular}%
  {%
    \glsssetAttribute{\the\glslabeltok}{regular}{false}%
  }
}

```

```

    }%
    {}%
  }%
}%
{%
  \GlsXtrUseAbbrStyleFmts{long-only-short-sc-only}%
}
\let\@glxstr@org@markright\markright
\renewcommand*{\markright}[1]{%
  \glxstrmarkhook
  \@glxstr@org@markright{\@glxstrinmark#1\@glxstrnotinmark}%
  \glxstrrestoremarkhook
}
\let\@glxstr@org@markboth\markboth
\renewcommand*{\markboth}[2]{%
  \glxstrmarkhook
  \@glxstr@org@markboth
    {\@glxstrinmark#1\@glxstrnotinmark}%
    {\@glxstrinmark#2\@glxstrnotinmark}%
  \glxstrrestoremarkhook
}
\let\@glxstr@org@@starttoc\@starttoc
\renewcommand*{\@starttoc}[1]{%
  \glxstrmarkhook
  \@glxstrinmark
  \@glxstr@org@@starttoc{#1}%
  \@glxstrnotinmark
  \glxstrrestoremarkhook
}
\newcommand*{\glxstrRevertMarks}{%
  \let\markright\@glxstr@org@markright
  \let\markboth\@glxstr@org@markboth
  \let\@starttoc\@glxstr@org@@starttoc
}
\newcommand*{\glxstrRevertTocMarks}{%
  \let\@starttoc\@glxstr@org@@starttoc
}
\newcommand*{\glxstrifinmark}[2]{#2}
\newrobustcmd*{\@glxstrinmark}{%
  \let\glxstrifinmark\@firstoftwo
}
\newrobustcmd*{\@glxstrnotinmark}{%
  \let\glxstrifinmark\@secondoftwo
}
\ifdef\texorpdfstring
{
  \newcommand*{\glxstrtitleorpdforheading}[3]{\texorpdfstring{#1}{#2}}
}
{
  \newcommand*{\glxstrtitleorpdforheading}[3]{#1}
}

```

```

}
\newcommand*{\glxstrmarkhook}{%
  \let\@glxstr@org@MakeUppercase\MakeUppercase
  \let\@glxstr@org@glxstrtitleorpdforheading\glxstrtitleorpdforheading
  \let\@glxstr@org@glxstrtitleshort\glxstrtitleshort
  \let\@glxstr@org@glxstrtitleshortpl\glxstrtitleshortpl
  \let\@glxstr@org@Glsxstrtitleshort\Glsxstrtitleshort
  \let\@glxstr@org@Glsxstrtitleshortpl\Glsxstrtitleshortpl
  \let\@glxstr@org@glxstrtitlename\glxstrtitlename
  \let\@glxstr@org@Glsxstrtitlename\Glsxstrtitlename
  \let\@glxstr@org@glxstrtitletext\glxstrtitletext
  \let\@glxstr@org@Glsxstrtitletext\Glsxstrtitletext
  \let\@glxstr@org@glxstrtitleplural\glxstrtitleplural
  \let\@glxstr@org@Glsxstrtitleplural\Glsxstrtitleplural
  \let\@glxstr@org@glxstrtitlefirst\glxstrtitlefirst
  \let\@glxstr@org@Glsxstrtitlefirst\Glsxstrtitlefirst
  \let\@glxstr@org@glxstrtitlefirstplural\glxstrtitlefirstplural
  \let\@glxstr@org@Glsxstrtitlefirstplural\Glsxstrtitlefirstplural
  \let\@glxstr@org@glxstrtitlelong\glxstrtitlelong
  \let\@glxstr@org@glxstrtitlelongpl\glxstrtitlelongpl
  \let\@glxstr@org@Glsxstrtitlelong\Glsxstrtitlelong
  \let\@glxstr@org@Glsxstrtitlelongpl\Glsxstrtitlelongpl
  \let\@glxstr@org@glxstrtitlefull\glxstrtitlefull
  \let\@glxstr@org@glxstrtitlefullpl\glxstrtitlefullpl
  \let\@glxstr@org@Glsxstrtitlefull\Glsxstrtitlefull
  \let\@glxstr@org@Glsxstrtitlefullpl\Glsxstrtitlefullpl
  \let\glxstrifinmark\@firstoftwo
  \let\MakeUppercase\MakeTextUppercase
  \let\glxstrtitleorpdforheading\@thirdofthree
  \let\glxstrtitleshort\glxstrheadshort
  \let\glxstrtitleshortpl\glxstrheadshortpl
  \let\Glsxstrtitleshort\Glsxstrheadshort
  \let\Glsxstrtitleshortpl\Glsxstrheadshortpl
  \let\glxstrtitlename\glxstrheadname
  \let\Glsxstrtitlename\Glsxstrheadname
  \let\glxstrtitletext\glxstrheadtext
  \let\Glsxstrtitletext\Glsxstrheadtext
  \let\glxstrtitleplural\glxstrheadplural
  \let\Glsxstrtitleplural\Glsxstrheadplural
  \let\glxstrtitlefirst\glxstrheadfirst
  \let\Glsxstrtitlefirst\Glsxstrheadfirst
  \let\glxstrtitlefirstplural\glxstrheadfirstplural
  \let\Glsxstrtitlefirstplural\Glsxstrheadfirstplural
  \let\glxstrtitlelong\glxstrheadlong
  \let\glxstrtitlelongpl\glxstrheadlongpl
  \let\Glsxstrtitlelong\Glsxstrheadlong
  \let\Glsxstrtitlelongpl\Glsxstrheadlongpl
  \let\glxstrtitlefull\glxstrheadfull
  \let\glxstrtitlefullpl\glxstrheadfullpl
  \let\Glsxstrtitlefull\Glsxstrheadfull

```



```

\let\Glsxtrtitlefullpl\Glsxtrheadfullpl
}
\newcommand*{\glxtrrestoremarkhook}{%
\let\glxtrifinmark\@secondoftwo
\let\MakeUppercase\@glxtr@org@MakeUppercase
\let\glxtrtitleorpdforheading\@glxtr@org@glxtrtitleorpdforheading
\let\glxtrtitleshort\@glxtr@org@glxtrtitleshort
\let\glxtrtitleshortpl\@glxtr@org@glxtrtitleshortpl
\let\Glsxtrtitleshort\@glxtr@org@Glsxtrtitleshort
\let\Glsxtrtitleshortpl\@glxtr@org@Glsxtrtitleshortpl
\let\glxtrtitlename\@glxtr@org@glxtrtitlename
\let\Glsxtrtitlename\@glxtr@org@Glsxtrtitlename
\let\glxtrtitletext\@glxtr@org@glxtrtitletext
\let\Glsxtrtitletext\@glxtr@org@Glsxtrtitletext
\let\glxtrtitleplural\@glxtr@org@glxtrtitleplural
\let\Glsxtrtitleplural\@glxtr@org@Glsxtrtitleplural
\let\glxtrtitlefirst\@glxtr@org@glxtrtitlefirst
\let\Glsxtrtitlefirst\@glxtr@org@Glsxtrtitlefirst
\let\glxtrtitlefirstplural\@glxtr@org@glxtrtitlefirstplural
\let\Glsxtrtitlefirstplural\@glxtr@org@Glsxtrtitlefirstplural
\let\glxtrtitlelong\@glxtr@org@glxtrtitlelong
\let\glxtrtitlelongpl\@glxtr@org@glxtrtitlelongpl
\let\Glsxtrtitlelong\@glxtr@org@Glsxtrtitlelong
\let\Glsxtrtitlelongpl\@glxtr@org@Glsxtrtitlelongpl
\let\glxtrtitlefull\@glxtr@org@glxtrtitlefull
\let\glxtrtitlefullpl\@glxtr@org@glxtrtitlefullpl
\let\Glsxtrtitlefull\@glxtr@org@Glsxtrtitlefull
\let\Glsxtrtitlefullpl\@glxtr@org@Glsxtrtitlefullpl
}
\newcommand*{\glxtrheadshort}[1]{%
\protect\NoCaseChange
{%
\glusifattribute{#1}{headuc}{true}%
{%
\Glsxtrshort[noindex,hyper=false]{#1}[]%
}%
}%
\glxtrshort[noindex,hyper=false]{#1}[]%
}%
}%
}
\newrobustcmd*{\glxtrtitleshort}[1]{%
\glxtrshort[noindex,hyper=false]{#1}[]%
}
\newcommand*{\glxtrheadshortpl}[1]{%
\protect\NoCaseChange
{%
\glusifattribute{#1}{headuc}{true}%
{%
\Glsxtrshortpl[noindex,hyper=false]{#1}[]%
}
}
}

```

```

    }%
    {%
        \glxstrshortpl[noindex,hyper=false]{#1}[]%
    }%
}%
}
\newrobustcmd*{\glxstrtitleshortpl}[1]{%
    \glxstrshortpl[noindex,hyper=false]{#1}[]%
}
\newcommand*{\Glsxtrheadshort}[1]{%
    \protect\NoCaseChange
    {%
        \glxifattribute{#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
    {%
        \glxifattribute{#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*{\glxtrheadname}[1]{%
    \protect\NoCaseChange
    {%
        \glxifattribute{#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}[]%
        }%
    }%
}

```

```

        \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]{%
    \glstrtitleshort{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*\glsfmtshortpl}[1]{%
    \texorpdfstring
      {\glstrtitleshortpl{#1}}%
      {\glstryshortpl{#1}}%
  }
}
{
  \newcommand*\glsfmtshortpl}[1]{%
    \glstrtitleshortpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*\Glsfmtshort}[1]{%
    \texorpdfstring
      {\Glsxtrtitleshort{#1}}%
      {\glstryshort{#1}}%
  }
}
{
  \newcommand*\Glsfmtshort}[1]{%
    \Glsxtrtitleshort{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*\Glsfmtshortpl}[1]{%
    \texorpdfstring
      {\Glsxtrtitleshortpl{#1}}%
      {\glstryshortpl{#1}}%
  }
}
{
  \newcommand*\Glsfmtshortpl}[1]{%
    \Glsxtrtitleshortpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*\glsfmtname}[1]{%
    \texorpdfstring
      {\glxtrtitlename{#1}}%
      {\glstryname{#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}}%
      {\glsppdfmtfull{#1}{}}%
    }
  }
  {
    \newcommand*{\Glsfmtfull}[1]{%
      \Glsxtrtitlefull{#1}}
  }
  \ifdef\texorpdfstring
  {
    \newcommand*{\GLSfmtfull}[1]{%
      \texorpdfstring
      {\GLSxtrtitlefull{#1}}%
      {\glsppdfmtfull{#1}}%
    }
  }
  {
    \newcommand*{\GLSfmtfull}[1]{%
      \GLSxtrtitlefull{#1}}
  }
  \ifdef\texorpdfstring
  {
    \newcommand*{\glsfmtfullpl}[1]{%
      \texorpdfstring
      {\glsxtrtitlefullpl{#1}}%
      {\glsppdfmtfullpl{#1}}%
    }
  }
  {
    \newcommand*{\glsfmtfullpl}[1]{%
      \glsxtrtitlefullpl{#1}}
  }
  \ifdef\texorpdfstring
  {
    \newcommand*{\Glsfmtfullpl}[1]{%
      \texorpdfstring
      {\Glsxtrtitlefullpl{#1}}%
      {\glsppdfmtfullpl{#1}{}}%
    }
  }
  {
    \newcommand*{\Glsfmtfullpl}[1]{%
      \Glsxtrtitlefullpl{#1}}
  }

```

```

{
  \newcommand*{\Glsfmtfullpl}[1]{%
    \Glsxtrtitlefullpl{#1}}
}
\ifdef\texorpdfstring
{
  \newcommand*{\Glsfmtfullpl}[1]{%
    \texorpdfstring
    {\Glsxtrtitlefullpl{#1}}%
    {\glspdfmtfullpl{#1}{}}%
  }
}
{
  \newcommand*{\Glsfmtfullpl}[1]{%
    \Glsxtrtitlefullpl{#1}}
}
\newcommand*{\multiglossaryentrysetup}[1]{\setkeys{glxstrcombined}{#1}}
\newcommand*{@gls@combined@indexmain}{1}
\define@choicekey{glxstrcombined}{indexmain}%
  [\@gls@combined@indexmain@val\@gls@combined@indexmain]
  {false,true,first}[true]{}
\newcommand*{@gls@combined@indexothers}{2}
\define@choicekey{glxstrcombined}{indexothers}%
  [\@gls@combined@indexothers@val\@gls@combined@indexothers]
  {false,true,first}[true]{}
\newcommand*{@gls@combined@hyper}{3}
\define@choicekey{glxstrcombined}{hyper}%
  [\@gls@combined@hyper@val\@gls@combined@hyper]
  {none,allmain,mainonly,individual,otheronly,notmainfirst,nototherfirst,notfirst}{}
\newcommand*{@gls@combined@encapmain}{glsnumberformat}
\define@key{glxstrcombined}{encapmain}{%
  \renewcommand*{@gls@combined@encapmain}{#1}%
}
\newcommand*{@gls@combined@encapothers}{glsnumberformat}
\define@key{glxstrcombined}{encapothers}{%
  \renewcommand*{@gls@combined@encapothers}{#1}%
}
\newcommand*{@gls@combined@textformat}{@firstofone}
\define@key{glxstrcombined}{textformat}{%
  \renewcommand*{@gls@combined@textformat}{#1}%
}
\newcommand*{@gls@combined@category}{}
\define@key{glxstrcombined}{category}{%
  \renewcommand*{@gls@combined@category}{#1}%
}
\define@key{glxstrcombinedpreopts}{category}{%
  \renewcommand*{@gls@combined@category}{#1}%
}
\newcommand*{@gls@combined@mglsopts}{}
\define@key{glxstrcombined}{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*{\mglsoptions@do}{%
\let\@gls@combined@mglsopts@do\@gls@combined@mglsopts@do@not
}
\newcommand*{\mglsoptions@enable}{%
\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}%
}%
}%

```

```

\ifinlist{\@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
    \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
            \let\writemultiglossentry\@gobblefour

```

```

\multiglossaryentryglobaltrue
\@multi@glossary@entry{#3}{#4}%
\egroup
\fi
}
\newcommand*\writemultiglossentry}[4]{%
\protected@write\@auxout{}\string\@glstr@multientry{#1}{#2}{#3}{#4}}%
}
\newcommand*\ifmglsused}[3]{%
\ifbool{@gls@combined@#1@flag}{#2}{#3}%
}
\newcommand*\mglsunset}[1]{%
\gls@ifnotmeasuring
{%
\glstrifmulti{#1}{\@mglsunset{#1}}%
{%
\glstrundefaction{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
{%
\glstrifmulti{#1}{\@mglsreset{#1}}%
{%
\glstrundefaction{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
{%
\glstrifmulti{#1}{\@mglslocalunset{#1}}%
{%
\glstrundefaction{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
  {%
    \glstrifmulti{#1}{\mglslocalreset{#1}}%
    {%
      \glstrundefaction{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)}{}%
    \fi
  }%
}

```

```

\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}{}}%
  {%
    \ifcseempty{@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\@mgl@unsetaction{0}
\define@choicekey{mgl}{multiunset}[\@mgl@unsetaction@val\@mgl@unsetaction]%
{global,local,none}{}
\define@boolkey{mgl}{presetlocal}[true]{}
\KV@mgl@presetlocalfalse
\newcommand*\@mgl@hyper{}{}
\define@choicekey{mgl}{hyper}[\@mgl@hyper@val\@mgl@hyper@nr]{true,false}[true]%
{%
  \renewcommand*\@mgl@hyper{hyper=#1}%
  \ifnum\@mgl@hyper@nr=1\relax
    \let\@mgl@hyperlink\@secondoftwo
  \else
    \let\@mgl@hyperlink\@mgl@hyperlink
  \fi
}
\newcommand*\@mgl@hyperlink[2]{%
  \ifx\@glslink\glsdonohyperlink
    #2%
  \else
    \glsxtr@org@dohyperlink{\glslinkprefix#1}{#2}%
  \fi
}
\let\@mgl@hyperlink\@mgl@hyperlink
\newcommand*\@mgl@forelements[3]{%
  \expandafter\@for\expandafter#2\expandafter:\expandafter
    =\csname @gls@combined@#1@list\endcsname\do{#3}%
}
\newcommand*\@mgl@forotherelements[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*\@mgl@unsetothers[1]{%
  \mgl@forotherelements{#1}{\@gls@tmp}{\glsunset{\@gls@tmp}}%
}
\newcommand*\@mgl@localunsetothers[1]{%
  \mgl@forotherelements{#1}{\@gls@tmp}{\glslocalunset{\@gls@tmp}}%
}
\newcommand*\@mgl@elementreset[1]{%
  \ifKV@mgl@presetlocal
    \glslocalreset{#1}%
  \else
    \glsreset{#1}%
  \fi
}
\newcommand*\@mgl@elementunset[1]{%
  \ifKV@mgl@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{\glxtr@setup@docurrent}{%
\ifx\mglscurrentlabel\mglscurrentmainlabel
  \mglsisfirstuse
  {%
    \ifKV@glxtrcombined@firstskipmain
      \let\@mgls@do@current@element\@gobble
    \else
      \let\@mgls@do@current@element\@firstofone
    \fi
  }%
  {%
    \ifKV@glxtrcombined@usedskipmain
      \let\@mgls@do@current@element\@gobble
    \else
      \let\@mgls@do@current@element\@firstofone

```

```

\fi
}%
\else
\mglsisfirstuse
{%
\ifKV@glstrcombined@firstskipothers
\let\@mgl@do@current@element\@gobble
\else
\let\@mgl@do@current@element\@firstofone
\fi
}%
{%
\ifKV@glstrcombined@usedskipothers
\let\@mgl@do@current@element\@gobble
\else
\let\@mgl@do@current@element\@firstofone
\fi
}%
\fi
}
\newcommand*{\glstr@mgl@checklastelement}[2]{%
\ifbool{KV@glstrcombined@#1skipmain}%
{%
\ifbool{KV@glstrcombined@#1skipothers}%
{%
}%
}%
{%
\ifnum\mglselementindex=\glstrmultilastotherindex{#2}\relax
\let\mgl@iflast\@firstoftwo
\else
\let\mgl@iflast\@secondoftwo
\fi
}%
}%
{%
\ifbool{KV@glstrcombined@#1skipothers}%
{%
\ifnum\mglselementindex=\glstrmultimainindex{#2}\relax
\let\mgl@iflast\@firstoftwo
\else
\let\mgl@iflast\@secondoftwo
\fi
}%
}%
\let\mgl@iflast\@secondoftwo
}%
}
\newcommand{\glstrmglWarnAllSkipped}[3]{%
\GlossariesExtraWarning{#1}%

```

```

#3{#2}%
}
\newcommand*{\glxtr@mgl@applyopts}[1]{%
  \edef\@mgl@doptions{\noexpand\setkeys*{mgl}{\expandonce#1}}%
  \@mgl@doptions
  \ifdefined\XKV@rm{\eappto\@mgl@all{\expandonce\XKV@rm}}%
  \ifdefined\@mgl@setup
  {%
    \edef\@mgl@doptions{%
      \noexpand\setkeys*{glxtrcombinedpreopts}{\expandonce\@mgl@setup}}%
    \mgl@disable@mglsopts
    \@mgl@doptions
    \mgl@enable@mglsopts
    \ifx\@mgl@setuptoptions\@empty
      \let\@mgl@setuptoptions\XKV@rm
    \else
      \eappto\@mgl@setuptoptions{\expandonce\XKV@rm}%
    \fi
  }%
  \@mgl@resetall
  \@mgl@unsetall
  \@mgl@resetmain
  \@mgl@unsetmain
  \@mgl@resetothers
  \@mgl@unsetothers
  \let\@mgl@resetall\@empty
  \let\@mgl@resetmain\@empty
  \let\@mgl@resetothers\@empty
  \let\@mgl@unsetall\@empty
  \let\@mgl@unsetmain\@empty
  \let\@mgl@unsetothers\@empty
  \ifmglused\mglcurrentmultilabel
  {\let\mglisfirstuse\@secondoftwo}%
  {\let\mglisfirstuse\@firstoftwo}%
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\providecommand{\@thirdofthree}[3]{#3}
\newcommand*{\glxtr@mgl@inner}[7]{%
  \let\mglslastmainlabel\@empty
  \let\mgl@iflastmainwasfirstuse\@firstoftwo
  \let\mgl@iflastmainwasplural\@secondoftwo
  \let\mgl@iflastmainincaps\@firstofthree
  \let\mgl@iflastmainskipped\@firstoftwo
  \bgroup
  \ifcsundef{@gls@combined@#2@main}%
  {%
    \glxtrundefaction{Multi entry ‘#2’ hasn’t been defined}%
    {You need to define ‘#2’ with \string\multiglossaryentry}%
  }
}

```

```

\gdef\@mgls@post@hookdefs{%
  \protected@edef\mglslastmultilabel{#2}%
  \let\mglswasfirsttuse\@firstoftwo
  \let\mglslastcategory\@empty
  \let\mglsiflastelementskipped\@firstoftwo
  \let\mglsiflastelementwasfirsttuse\@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\mglsisfirsttuse\@secondoftwo}%
  {\let\mglsisfirsttuse\@firstoftwo}%
  \edef\@mgls@doptions{%
    \noexpand\setkeys*{glxtrcombinedpreopts}{\expandonce\mglscurrentoptions}}%
  \@mgls@doptions
  \let\@mgls@setuptoptions\XKV@rm
  \mgls@disable@setup
  \ifdefvoid\@gls@combined@mgls@sopts
  {}%
  {\glxtr@mgls@applyopts\@gls@combined@mgls@sopts}%
  \mgls@enable@setup
  \ifstrempy{#1}{\def\@mgls@options{#1}\glxtr@mgls@applyopts\@mgls@options}%
  \ifx\@gls@combined@category\empty
  \else
    \glshascategoryattribute{\@gls@combined@category}{\multioptions}%
    {%
      \letcs\@mgls@attroptions{\glxtr@categoryattr@\@gls@combined@category
        @multioptions}%
      \let\@gls@combined@mgls@sopts\@empty
      \edef\@mgls@doptions{%
        \noexpand\setkeys*{glxtrcombinedpreopts}{\expandonce\@mgls@attroptions}}%
      \@mgls@doptions
      \eappto\@mgls@setuptoptions{,\expandonce\XKV@rm}%
      \ifx\@gls@combined@mgls@sopts\@empty
      \else
        \let\@mgls@setup\@empty
        \mgls@disable@setup
        \glxtr@mgls@applyopts\@gls@combined@mgls@sopts
        \mgls@enable@setup
      \fi
    }%
  }%
}%

```

```

\fi
\edef\@mgls@dooptions{%
  \noexpand\setkeys{glsxtrcombined}{\expandonce\@mgls@setuptools}}%
\@mgls@dooptions
\let\mgls@currentcategory\@gls@combined@category
\ifnum\@gls@combined@hyper=1\relax
  \def\@mgls@combinedlink{\@mgls@hyperlink{\mgls@currentmainlabel}}%
\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@glxtrcombined@firstskipmain
    \ifKV@glxtrcombined@firstskipothers
      \let\@gls@org@combined@encapsulator\@gls@combined@encapsulator
      \def\@gls@combined@encapsulator##1{%
        \glxtrmglsWarnAllSkipped{All elements skipped for
          first use of multi-entry '#2'}{#3}%
        {\@gls@org@combined@encapsulator}%
      }%
      \let\@mgls@do@current@element\@gobble
    \fi
  \fi
}%
{%
  \ifKV@glxtrcombined@usedskipmain
    \ifKV@glxtrcombined@usedskipothers
      \let\@gls@org@combined@encapsulator\@gls@combined@encapsulator
      \def\@gls@combined@encapsulator##1{%
        \glxtrmglsWarnAllSkipped{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\mgls@currentprefix\@gls@combined@firstprefix
  \let\mgls@currentsuffix\@gls@combined@firstsuffix
}%
{%
  \let\mgls@currentprefix\@gls@combined@usedprefix
  \let\mgls@currentsuffix\@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\mglsiflastelements\skipped\@firstoftwo
\let\mglslastelementlabel\@empty
\let\mglsiflastelementwasfirstuse\@firstoftwo
\let\mglsiflastelementwasplural\@secondoftwo
\let\mglsiflastelementcaps\case\@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\@mgl@post@hookdefs{\let\mgl@do@postlinkhook\relax}%
\or
\ifcase\@gls@combined@mpostlinkelement@nr\relax
\gappto\@mgl@post@hookdefs{\let\mgl@do@postlinkhook\mglslastelementpostlinkhook}%
\or
\gappto\@mgl@post@hookdefs{\let\mgl@do@postlinkhook\mglslastmainpostlinkhook}%
\or
\gappto\@mgl@post@hookdefs{\let\mgl@do@postlinkhook\mglscustompostlinkhook}%
\fi
\fi
}%
\let\mgl@org@postlinkhook\glspostlinkhook
\mglsprefix
\let\mglslastelementlabel\@empty
\@gls@combined@encapsulator
{%
\def\@mgl@previouslabel{}%
\mglselementindex=0\relax
\@for\mglscurrentlabel:=\mglscurrentlist\do{%
\advance\mglselementindex by 1\relax
\glxtr@setup@docurrent
\ifx\@xfor@nextelement\@nnil
\let\mgl@iflast\@firstoftwo
\else
\let\mgl@iflast\@secondoftwo
\mglsisfirstuse
{%
\glxtr@mgl@checklastelement{first}{#2}%
}%
{%
\glxtr@mgl@checklastelement{used}{#2}%
}%
\fi
\ifcase\@gls@combined@postlinks@nr\relax
\let\glspostlinkhook\relax
\or
\let\glspostlinkhook\mgl@org@postlinkhook
\or
\mgl@iflast
{%
\let\glspostlinkhook\relax
}%
{%
\let\glspostlinkhook\mgl@org@postlinkhook
}%
\or
\ifx\mglscurrentlabel\mglscurrentmainlabel
\mgl@iflast
{%
\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\@mgl@current@options{,noindex=false}%
\or
\@mgl@current@iffirstuse
{\appto\@mgl@current@options{,noindex=false}}%
{\appto\@mgl@current@options{,noindex}}%
\fi
\ifcase\@gl@combined@hyper\relax
\appto\@mgl@current@options{,hyper=false}% none
\or
\appto\@mgl@current@options{,hyper=false}% allmain
\or
\@appto\@mgl@current@options{,\@mgl@hyper}% mainonly
\or
\@appto\@mgl@current@options{,\@mgl@hyper}% individual
\or
\appto\@mgl@current@options{,hyper=false}% otheronly
\or
\mgl@firstuse
{%
\appto\@mgl@current@options{,hyper=false}% notmainfirst
}%
{%
\@appto\@mgl@current@options{,\@mgl@hyper}% notmainfirst
}%
\or
\@appto\@mgl@current@options{,\@mgl@hyper}% nototherfirst
\or
\mgl@firstuse
{%
\appto\@mgl@current@options{,hyper=false}% notfirst
}%
{%
\@appto\@mgl@current@options{,\@mgl@hyper}% notfirst
}%
\fi
\@appto\@mgl@current@options{,\@mgl@all,\@mgl@main}%
\else
\edef\@mgl@current@options{format=\@gl@combined@encapothers}%
\ifcase\@gl@combined@indexothers\relax
\appto\@mgl@current@options{,noindex}%
\or
\appto\@mgl@current@options{,noindex=false}%
\or
\@mgl@current@iffirstuse
{\appto\@mgl@current@options{,noindex=false}}%
{\appto\@mgl@current@options{,noindex}}%
\fi
\ifcase\@gl@combined@hyper\relax
\appto\@mgl@current@options{,hyper=false}% none
\or

```

```

\appto\@mgl@current@options{,hyper=false}% allmain
\or
\appto\@mgl@current@options{,hyper=false}% mainonly
\or
\appto\@mgl@current@options{,\@mgl@hyper}% individual
\or
\appto\@mgl@current@options{,\@mgl@hyper}% otheronly
\or
\appto\@mgl@current@options{,\@mgl@hyper}% notmainfirst
\or
\mgl@sisfirstuse
{%
\appto\@mgl@current@options{,hyper=false}% nototherfirst
}%
{%
\appto\@mgl@current@options{,\@mgl@hyper}% nototherfirst
}%
\or
\mgl@sisfirstuse
{%
\appto\@mgl@current@options{,hyper=false}% notfirst
}%
{%
\appto\@mgl@current@options{,\@mgl@hyper}% notfirst
}%
\fi
\appto\@mgl@current@options{,\@mgl@all,\@mgl@others}%
\fi
\ifx\@mgl@previouslabel\empty
\ifx\mgl@currentlabel\mgl@currentmainlabel
\let\@mgl@cs#6\relax
\else
\let\@mgl@cs#4\relax
\fi
\else
\@mgl@previous@iffirstuse
{%
\@mgl@current@iffirstuse
{\glscombinedfirstsepfirst{\@mgl@previouslabel}{\mgl@currentlabel}}%
{\glscombinedfirstsep{\@mgl@previouslabel}{\mgl@currentlabel}}%
}%
{%
\@mgl@current@iffirstuse
{\glscombinedsepfirst{\@mgl@previouslabel}{\mgl@currentlabel}}%
{\glscombinedsep{\@mgl@previouslabel}{\mgl@currentlabel}}%
}%
\ifx\mgl@currentlabel\mgl@currentmainlabel
\let\@mgl@cs#7\relax
\else
\let\@mgl@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}%
}%
\glstrifwasfirstuse
{%
\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
}%
\mgls@element@posthook
}%
\ifx\mglslastmainlabel\@empty
\gappto\@mgls@post@hookdefs{\let\mglsiflastmainskipped\@firstoftwo}%
\else
\gappto\@mgls@post@hookdefs{\let\mglsiflastmainskipped\@secondoftwo}%
\fi

```

```

\fi
\ifx\@mgls@do@current@element\@gobble
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementskipped\@firstoftwo}%
\else
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementskipped\@secondoftwo}%
\fi
\glstrifwasfirstuse
{%
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@firstoftwo}%
}%
{%
\gappto\@mgls@post@hookdefs{\let\mglsiflastelementwasfirstuse\@secondoftwo}%
}%
\glssifplural
{%
\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
\mglsuffix
\ifcase\@mgls@unsetaction\relax
\xappto\@mgls@post@hookdefs{%
\noexpand\mglsunset{\expandonce\mglscurrentmultilabel}}%
\or
\xappto\@mgls@post@hookdefs{%
\noexpand\mglslocalunset{\expandonce\mglscurrentmultilabel}}%
\fi
}%
\glstrmglswrite{#2}%
\egroup
\@mgls@post@hookdefs

```

```

\mgl@do@postlinkhook
}
\newcommand*{\mglscustompostlinkhook}{}
\newcommand*{\mglslastelementpostlinkhook}{%
\let\glxtrifwasfirstuse\mgl@iflastelementwasfirstuse
\let\gl@ifplural\mgl@iflastelementwasplural
\let\glscapscase\mgl@iflastelementcapscase
\let\glslabel\mglslastelementlabel
\glspostlinkhook
}
\newcommand*{\mglslastmainpostlinkhook}{%
\let\glxtrifwasfirstuse\mgl@iflastmainwasfirstuse
\let\gl@ifplural\mgl@iflastmainwasplural
\let\glscapscase\mgl@iflastmaincapscase
\let\glslabel\mglslastmainlabel
\glspostlinkhook
}
\newcommand*{\mgldefcategoryprefix}[2]{%
\csdef{mglsprefix@#1}{#2}%
}
\newcommand*{\mglshascategoryprefix}[3]{%
\ifcsdef{mglsprefix@#1}{#2}{#3}%
}
\newcommand*{\mglsecategoryprefix}[1]{%
\csuse{mglsprefix@#1}%
}
\newcommand*{\mglsprefix}{%
\ifdefempty\mglcurrentcategory
{\mglcurrentprefix}%
{%
\mglshascategoryprefix{\mglcurrentcategory}%
{\mglsecategoryprefix{\mglcurrentcategory}}%
{\mglcurrentprefix}%
}%
}
\newcommand*{\mgldefcategorysuffix}[2]{%
\csdef{mglssuffix@#1}{#2}%
}
\newcommand*{\mglshascategorysuffix}[3]{%
\ifcsdef{mglssuffix@#1}{#2}{#3}%
}
\newcommand*{\mglsecategorysuffix}[1]{%
\csuse{mglssuffix@#1}%
}
\newcommand*{\mglssuffix}{%
\ifdefempty\mglcurrentcategory
{\ifdefempty\mglcurrentsuffix}{ }\space(\mglcurrentsuffix)}%
{%
\mglshascategorysuffix{\mglcurrentcategory}%
{\mglsecategorysuffix{\mglcurrentcategory}}%
}

```

```

    {\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*{\glssetcombinedsepabbrvnbs}{%
  \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*{\glscombinedsefirst}[2]{%
    \glshasattribute{##1}{combinedsefirst}%
    {\glsgetattribute{##1}{combinedsefirst}}%
    {\ifhasshort{##1}{ } }%
}%
\renewcommand*{\glscombinedfirstsep}[2]{%
    \glshasattribute{##1}{combinedfirstsep}%
    {\glsgetattribute{##1}{combinedfirstsep}}%
    {\ifhasshort{##2}{ } }%
}%
\renewcommand*{\glscombinedfirstsefirst}[2]{%
    \glshasattribute{##1}{combinedfirstsefirst}%
    {\glsgetattribute{##1}{combinedfirstsefirst}}%
    { }%
}%
}
\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*{\glscombinedsefirst}[2]{%
    \glshasattribute{##1}{combinedsefirst}%
    {\glsgetattribute{##1}{combinedsefirst}}%
    {%
        \ifhasshort{##1}%
        {\settowidth{\dimen@}{\glsentryshort{##1}}}%
        {\settowidth{\dimen@}{\glsentrytext{##1}}}%
        \ifdim\dimen@<#1\relax
            #2%

```

```

\else
  \ifhaslong{##2}%
    {\settowidth{\dimen@}{\glentrylong{##2}}}%
    {\settowidth{\dimen@}{\glentryfirst{##2}}}%
    \ifdim\dimen@<#1\relax
      #2%
    \else
      \space
    \fi
  \fi
}%
}%
\renewcommand*{\glscmbinedfirstsep}[2]{%
\glshasattribute{##1}{combinedfirstsep}%
{\glsggetattribute{##1}{combinedfirstsep}}%
{%
  \ifhaslong{##1}%
    {\settowidth{\dimen@}{\glentrylong{##1}}}%
    {\settowidth{\dimen@}{\glentryfirst{##1}}}%
    \ifdim\dimen@<#1\relax
      #2%
    \else
      \ifhasshort{##2}%
        {\settowidth{\dimen@}{\glentryshort{##2}}}%
        {\settowidth{\dimen@}{\glentrytext{##2}}}%
        \ifdim\dimen@<#1\relax
          #2%
        \else
          \space
        \fi
      \fi
    }%
  }%
}%
\renewcommand*{\glscmbinedfirstsepfir}[2]{%
\glshasattribute{##1}{combinedfirstsepfir}%
{\glsggetattribute{##1}{combinedfirstsepfir}}%
{%
  \ifhaslong{##1}%
    {\settowidth{\dimen@}{\glentrylong{##1}}}%
    {\settowidth{\dimen@}{\glentryfirst{##1}}}%
    \ifdim\dimen@<#1\relax
      #2%
    \else
      \ifhaslong{##2}%
        {\settowidth{\dimen@}{\glentrylong{##2}}}%
        {\settowidth{\dimen@}{\glentryfirst{##2}}}%
        \ifdim\dimen@<#1\relax
          #2%
        \else
          \space
        \fi
      \fi
    }%
  }%
}%

```

```

        \fi
    \fi
}%
}%
}
\newcommand{\glstrmglswrite}[1]{%
\ifx\@glstr@record@setting\@glstr@record@setting@off
\else
\protected@edef\@glstr@mglslabel{#1}%
\ifdef\@glstr@mglssreflist
{%
\expandafter\DTLifinlist\expandafter{\@glstr@mglslabel}%
{\@glstr@mglssreflist}{}%
{%
\xappto\@glstr@mglssreflist{,\expandonce\@glstr@mglslabel}%
\if@mglss@writeseparaterefs
\protected@write\@auxout{}{\string\@glstr@mglssrefs{#1}}%
\fi
}%
}%
}%
}%
{%
\global\let\@glstr@mglssreflist\@glstr@mglslabel
\if@mglss@writeseparaterefs
\protected@write\@auxout{}{\string\@glstr@mglssrefs{#1}}%
\else
\AtEndDocument{\immediate\protected@write\@auxout{}%
{\string\@glstr@mglssrefs{\@glstr@mglssreflist}}}%
\fi
\@mglss@disable@writeseparateref@cond
}%
\fi
}
\newcommand{\@glstr@mglssrefs}[1]{%
\newif\if@mglss@writeseparaterefs \@mglss@writeseparaterefsfalse
\newcommand{\mglssWriteSeparateRefsTrue}{\global\@mglss@writeseparaterefstrue}
\newcommand{\mglssWriteSeparateRefsFalse}{\global\@mglss@writeseparaterefsfalse}
\newcommand*{\@mglss@disable@writeseparateref@cond}{%
\gdef\mglssWriteSeparateRefsTrue{\PackageError{glossaries-extra}%
{Too late to use \string\mglssWriteSeparateRefsTrue}%
{\string\mglssWriteSeparateRefsTrue\space can only be used before
the first instance of any \string\mglss-like command}}%
\gdef\mglssWriteSeparateRefsFalse{\PackageError{glossaries-extra}%
{Too late to use \string\mglssWriteSeparateRefsFalse}%
{\string\mglssWriteSeparateRefsFalse\space can only be used before
the first instance of any \string\mglss-like command}}%
}
\newcommand{\glstr@newmglss}[5]{%
\edef\@glstr@newmglss@do{%
\noexpand\newrobustcmd*{\expandonce{\csname #1\endcsname}}%
{\noexpand\@glss@hyp@opt\expandonce{\csname ns@glstr@#1\endcsname}}%

```



```

\glxstrifmulti{#2}%
{\@glxstr@mgl@mcs+[#1]{#2}}%
{\@glxstr@mgl@gcs+[#1]{#2}}%
}
\newcommand*{\@alt@GlsXtrMglOrGls}[2][{}]{%
\glxstrifmulti{#2}%
{\expandafter\@glxstr@mgl@mcs\@gl@alt@hyp@opt@char[#1]{#2}}%
{\expandafter\@glxstr@mgl@gcs\@gl@alt@hyp@opt@char[#1]{#2}}%
}
\glxtr@newmgl@mgl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@pl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@mainpl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@pl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@mainpl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@pl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@mainpl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@pl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\glxtr@newmgl@mgl@mainpl@{\@gl@}{\@gl@}{\@gl@}{\@gl@}%
\def\@gl@longortext#1#2[#3]{%
\ifglshaslong{#2}{\@glxtr@long{#1}{#2}[#3]}{\@glstext@{#1}{#2}[#3]}%
}
\def\@gl@shortortext#1#2[#3]{%
\ifglshasshort{#2}{\@glxtr@short{#1}{#2}[#3]}{\@glstext@{#1}{#2}[#3]}%
}
\def\@gl@fullorfirst#1#2[#3]{%
\ifglshasshort{#2}{\@glxtr@full{#1}{#2}[#3]}{\@gl@first@{#1}{#2}[#3]}%
}
\def\@Gls@longortext#1#2[#3]{%
\ifglshaslong{#2}{\@Glsxtr@long{#1}{#2}[#3]}{\@Glstext@{#1}{#2}[#3]}%
}
\def\@Gls@shortortext#1#2[#3]{%
\ifglshasshort{#2}{\@Glsxtr@short{#1}{#2}[#3]}{\@Glstext@{#1}{#2}[#3]}%
}
\def\@Gls@fullorfirst#1#2[#3]{%
\ifglshasshort{#2}{\@Glsxtr@full{#1}{#2}[#3]}{\@Gls@first@{#1}{#2}[#3]}%
}
\glxtr@newmgl@mgl@short@%
{\@gl@shortortext}{\@gl@shortortext}{\@gl@shortortext}{\@gl@shortortext}%
\glxtr@newmgl@mgl@long@%
{\@gl@longortext}{\@gl@longortext}{\@gl@longortext}{\@gl@longortext}%
\glxtr@newmgl@mgl@full@%
{\@gl@fullorfirst}{\@gl@fullorfirst}{\@gl@fullorfirst}{\@gl@fullorfirst}%
\glxtr@newmgl@mgl@short@%
{\@gl@shortortext}{\@gl@shortortext}{\@gl@shortortext}{\@gl@shortortext}%
\glxtr@newmgl@mgl@long@%
{\@gl@longortext}{\@gl@longortext}{\@gl@longortext}{\@gl@longortext}%
\glxtr@newmgl@mgl@full@%

```

```

{\@Glsfullorfirst}{\@Glsfullorfirst}{\@Glsfullorfirst}{\@Glsfullorfirst}%
\glxtr@newmgls{mglsname}%
{\@Glsname@}{\@Glsname@}{\@Glsname@}{\@Glsname@}%
\glxtr@newmgls{MGlsname}%
{\@Glsname@}{\@Glsname@}{\@Glsname@}{\@Glsname@}%
\glxtr@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]}%
}
\glxtr@newmgls{mglsymbol}%
{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
\glxtr@newmgls{Mglsymbol}%
{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
\glxtr@newmgls{MGlsymbol}%
{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}{\@glssymbolorgls}%
\newcommand{\mglsfield}{useri}
\def\@glsgfieldorgls#1#2[#3]{%
  \glxtrifhasfield{\mglsfield}{#2}%
  {\@Glsdisp[#1]{#2}{\glscurrentfieldvalue#3}}%
  {\@Gls@{#1}{#2}[#3]}%
}
\def\@Glsfieldorgls#1#2[#3]{%
  \glxtrifhasfield{\mglsfield}{#2}%
  {\@Glsdisp[#1]{#2}{\xmakefirstuc\glscurrentfieldvalue#3}}%
  {\@Gls@{#1}{#2}[#3]}%
}
\glxtr@newmgls{mglsusefield}%
{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}%
\glxtr@newmgls{Mglsusefield}%
{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}{\@Glsfieldorgls}%
\glxtr@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\@glsp@{#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]}%
}
\glxstr@newmgls{mpgls}{\@pglsorgls@}{\@gls@}{\@pglsorgls@}{\@gls@}%
\glxstr@newmgls{mpglspl}{\@pglsorglspl@}{\@glsp@}{\@pglsorglspl@}{\@glsp@}%
\glxstr@newmgls{mpglsmainpl}{\@pglsorgls@}{\@gls@}{\@pglsorglspl@}{\@glsp@}%
\glxstr@newmgls{Mpgls}{\@Pglsorgls@}{\@gls@}{\@Pglsorgls@}{\@gls@}%
\glxstr@newmgls{Mpglspl}{\@Pglsorglspl@}{\@glsp@}{\@Pglsorglspl@}{\@glsp@}%
\glxstr@newmgls{Mpglsmainpl}{\@Pglsorgls@}{\@gls@}{\@Pglsorglspl@}{\@glsp@}%
\glxstr@newmgls{MPGls}{\@Pglsorgls@}{\@GLS@}{\@Pglsorgls@}{\@GLS@}%
\glxstr@newmgls{MPGLspl}{\@Pglsorglspl@}{\@GLSpl@}{\@Pglsorglspl@}{\@GLSpl@}%
\glxstr@newmgls{MPGLSmainpl}{\@PGLSorgls@}{\@GLS@}{\@PGLSorglspl@}{\@GLSpl@}%
\glxstr@newmgls{MPGLS}{\@PGLSorgls@}{\@GLS@}{\@PGLSorgls@}{\@GLS@}%
\glxstr@newmgls{MPGLSpl}{\@PGLSorglspl@}{\@GLSpl@}{\@PGLSorglspl@}{\@GLSpl@}%
\glxstr@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{\glxstr@loaddialect}{%
  \IfTrackedLanguageFileExists{\this@dialect}%
  {glossariesxtr-}% prefix
  {.ldf}%
  {%
    \RequireGlossariesExtraLang{\CurrentTrackedTag}%
  }%
  {}% not found
  \@glxstr@dialecthook
}
\@ifpackageloaded{tracklang}{%
  \AnyTrackedLanguages
  {%
    \ForEachTrackedDialect{\this@dialect}{\glxstr@loaddialect}%
  }%
  {}%
} {}
\@glxstr@redefstyles
\@glxstr@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*\glsapturedgroup{\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*\glstrprovidecommand{\providecommand}
\newcommand*\glstrenewcommand{\@star@or@long\glstr@renewcommand}
\newcommand*\glstr@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
\newcommand#1%
}
\newcommand*\glstr@wrglossarylocation[2]{#1}
\ifdef\hyperref
{%
\newcommand*\GlsXtrIndexCounterLink[2]{%
\glstrifhasfield{indexcounter}{#2}%
{\hyperref[wrglossary.\glscurrentfieldvalue]{#1}}%
{#1}%
}
}
{
\newcommand*\GlsXtrIndexCounterLink[2]{#1}
}
\newcommand*\GlsXtrDualField{dual}
\newcommand*\GlsXtrDualBackLink[2]{%
\glstrifhasfield{\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}{}\glstrbibaddress}%

```

```

\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}{\glsxtrbibschooll}%
\glsaddstoragekey{series}{\glsxtrbibseries}%
\glsaddstoragekey{title}{\glsxtrbibtitle}%
\glsaddstoragekey{bibtexkey}{\glsxtrbibtype}%
\glsaddstoragekey{volume}{\glsxtrbibvolume}%
}
\newcommand*{\glsxtrmultisupplocation}[3]{%
  {%
    \def\glsxtrsupplocationurl{#2}%
    \glsnumber{#1}%
  }%
}
\newcommand*{\glsxtrdisplaysupplc}[5]{%
  \setentrycounter{#1}{#2}%
  \glsxtrmultisupplocation{#5}{#4}{#3}%
}
\ifundef\hyperlink
{
  \newcommand*{\glsxtrdisplaylocnameref}[8]{%
    \glsnoidxdisplayloc{#1}{#2}{#3}{#4}%
  }
}
{
  \newcommand*{\glsxtrdisplaylocnameref}[8]{%
    \ifcsdef{glsxtr#2locfmt}%
    {\glsxtrnameref{#3}{\csuse{glsxtr#2locfmt}{#4}{#5}}{#2.#7}{#8}}%
    {%
      \ifstrempy{#5}%
      {%
        \glsxtrnameref{#3}{#4}{#2.#7}{#8}%
      }%
      {%
        \ifstrequal{#2}{page}%
        {\glsxtrnameref{#3}{#4}{#2.#7}{#8}}%
        {\glsxtrnameref{#3}{#5}{#2.#7}{#8}}%
      }%
    }%
  }%
}

```

```

    }
}
\newcommand*{\glxstrequationlocfmt}[2]{(#1)}
\newcommand*{\glxtrnamerefink}[4]{%
  \begingroup
    \let\glshypernumber\@firstofone
    \ifstrempy{#4}%
      {\glxtrfmtinternalnameref{#3}{#1}{#2}}%
      {\glxtrfmtexternalnameref{#3}{#1}{#2}{#4}}%
    \endgroup
}
\newcommand{\glxtrnameloclink}[6]{%
  \begingroup
    \setentrycounter[#1]{#2}%
    \def\glxtr@locationhypertext{#5}%
    \let\glshypernumber\@firstofone
    \def\@glsnumberformat{#3}%
    \def\glxtrsupplocationurl{#6}%
    \toks@={}%
    \@glxtr@bibgls@removespaces#4 \@nil
  \endgroup
}
\def\@glxtr@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{\glsetentrycounter\@glo@counterprefix\the\toks@}%
        \ifdefvoid\glxtrsupplocationurl
          {%
            \expandafter\glxtrfmtinternalnameref\expandafter{\@glo@tmp}%
            {\@glsnumberformat}{\glxtr@locationhypertext}%
          }%
          {%
            \expandafter\glxtrfmtexternalnameref\expandafter{\@glo@tmp}%
            {\@glsnumberformat}{\glxtr@locationhypertext}{\glxtrsupplocationurl}%
          }%
        \fi
      \else
        \@gls@ReturnAfterFi{%
          \@glxtr@bibgls@removespaces#2\@nil
        }%
      \fi
    }
\newcommand*{\glxtrfmtinternalnameref}[3]{%
  \csuse{#2}{\glsdohyperlink{#1}{#3}}%
}
\newcommand*{\glxtrfmtexternalnameref}[4]{%
  \csuse{#2}{\hyperref{#4}{#1}{#3}}%
}

```

```

}
\newcommand*{\glxtrSetWidest}[3]{%
  \ifdef\glupdatewidest
  {%
    \ifdef\glslongextraUpdateWidest
    {%
      \ifstrepty{#1}
      {%
        \glupdatewidest[#2]{#3}%
        \ifnum#2=0\relax
        \glslongextraUpdateWidest{#3}%
        \else
        \glslongextraUpdateWidestChild{#2}{#3}%
        \fi
      }%
    }%
    \apptoglossarypreamble[#1]{\glupdatewidest[#2]{#3}}%
    \ifnum#2=0\relax
    \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
    \else
    \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
    \fi
  }%
}%
{%
  \ifstrepty{#1}
  {%
    \glupdatewidest[#2]{#3}%
  }%
  {%
    \apptoglossarypreamble[#1]{\glupdatewidest[#2]{#3}}%
  }%
}%
\ifdef\glsetwidest
{%
  \ifdef\glslongextraUpdateWidest
  {%
    \ifstrepty{#1}
    {%
      \glsetwidest[#2]{#3}%
      \ifnum#2=0\relax
      \glslongextraUpdateWidest{#3}%
      \else
      \glslongextraUpdateWidestChild{#2}{#3}%
      \fi
    }%
  }%
  {%
    \apptoglossarypreamble[#1]{\glsetwidest[#2]{#3}}%
  }%
}

```

```

        \ifnum#2=0\relax
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidest{#3}}%
        \else
        \apptoglossarypreamble[#1]{\glslongextraUpdateWidestChild{#2}{#3}}%
        \fi
    }%
}%
{%
    \ifstrepty{#1}
    {%
        \glssetwidest[#2]{#3}%
    }%
    {%
        \apptoglossarypreamble[#1]{\glssetwidest[#2]{#3}}%
    }%
}%
}%
{%
    \ifdef\glslongextraUpdateWidest
    {%
        \ifstrepty{#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}%
    }%
}

```

```

\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]{%
\ifstrempy{#1}%
{\glsxtraddlabelprefix{\empty}}%
{%
\ifdefempty\@glsxtr@labelprefixes
{\def\@glsxtr@labelprefixes{#1}}%
{\appto\@glsxtr@labelprefixes{,#1}}%
}%
}
}
\newcommand*{\glsxtrprependlabelprefix}[1]{%
\ifstrempy{#1}%
{\glsxtrprependlabelprefix{\empty}}%
{%
\ifdefempty\@glsxtr@labelprefixes
{\def\@glsxtr@labelprefixes{#1}}%
{\preto\@glsxtr@labelprefixes{#1,}}%
}%
}
}
\newcommand*{\glsxtrifinlabelprefixlist}[3]{%
\ifstrempy{#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*{\dgl}{\@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\@glxstrmultientryadjustednamesep\glxstrmultientryadjustednamesep
\let\@glxstrmultientryadjustednamepresep\glxstrmultientryadjustednamepresep
\let\@glxstrmultientryadjustednamepostsep\glxstrmultientryadjustednamepostsep
\let\@glxstrmultientryadjustednameother\glxstrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefmt\glxstrmultientryadjustednamefmt
\let\@glxstrmultientryadjustednamefirstother\glxstrmultientryadjustednameother
\let\@glxstrmultientryadjustednamefirstfmt\glxstrmultientryadjustednamefmt
\@glxstrmultientryadjustedname{#1}{#2}{#3}{#4}%
\egroup
}
\newrobustcmd*\Glsxstrmultientryadjustedname}[4]{%
  \bgroup
  \let\@glxstrmultientryadjustednamesep\glxstrmultientryadjustednamesep
  \let\@glxstrmultientryadjustednamepresep\glxstrmultientryadjustednamepresep
  \let\@glxstrmultientryadjustednamepostsep\glxstrmultientryadjustednamepostsep
  \let\@glxstrmultientryadjustednameother\glxstrmultientryadjustednameother
  \let\@glxstrmultientryadjustednamefmt\glxstrmultientryadjustednamefmt
  \let\@glxstrmultientryadjustednamefirstother\Glsxstrmultientryadjustednameother
  \let\@glxstrmultientryadjustednamefirstfmt\Glsxstrmultientryadjustednamefmt
  \@glxstrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
\newrobustcmd*\GlsXstrmultientryadjustedname}[4]{%
  \bgroup
  \let\@glxstrmultientryadjustednamesep\glxstrmultientryadjustednamesep
  \let\@glxstrmultientryadjustednamepresep\glxstrmultientryadjustednamepresep
  \let\@glxstrmultientryadjustednamepostsep\glxstrmultientryadjustednamepostsep
  \let\@glxstrmultientryadjustednameother\GlsXstrmultientryadjustednameother
  \let\@glxstrmultientryadjustednamefmt\GlsXstrmultientryadjustednamefmt
  \let\@glxstrmultientryadjustednamefirstother\GlsXstrmultientryadjustednameother
  \let\@glxstrmultientryadjustednamefirstfmt\GlsXstrmultientryadjustednamefmt
  \@glxstrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
\newrobustcmd*\GLSxstrmultientryadjustedname}[4]{%
  \bgroup
  \let\@glxstrmultientryadjustednamesep\glxstrmultientryadjustednamesep
  \let\@glxstrmultientryadjustednamepresep\glxstrmultientryadjustednamepresep
  \let\@glxstrmultientryadjustednamepostsep\glxstrmultientryadjustednamepostsep
  \let\@glxstrmultientryadjustednameother\GLSxstrmultientryadjustednameother
  \let\@glxstrmultientryadjustednamefmt\GLSxstrmultientryadjustednamefmt
  \let\@glxstrmultientryadjustednamefirstother\GLSxstrmultientryadjustednameother
  \let\@glxstrmultientryadjustednamefirstfmt\GLSxstrmultientryadjustednamefmt
  \@glxstrmultientryadjustedname{#1}{#2}{#3}{#4}%
  \egroup
}
\newcommand*\@glxstrmultientryadjustedname}[4]{%
  \letcs\mglscurrentmainlabel{\@glsc@combined@#4@main}%
  \letcs\mglscurrentmainlist{\@glsc@combined@#4@list}%
  \letcs\mglscurrentmainoptions{\@glsc@combined@#4@options}%

```



```

\ifblank{#1}%
{%
  \@glxstrmultientryadjustednamefirstfmt{#2}%
}%
{%
  \def\@mgls@previouslabel{}%
  \let\@gls@xtradjustedother\@glxstrmultientryadjustednamefirstother
  \@for\mglscurrentlabel:=#1\do{%
    \ifx\@mgls@previouslabel\empty
    \else
      \@glxstrmultientryadjustednamesep{\@mgls@previouslabel}{\mglscurrentlabel}%
    \fi
    \@gls@xtradjustedother{\mglscurrentlabel}%
    \let\@mgls@previouslabel\mglscurrentlabel
    \let\@gls@xtradjustedother\@glxstrmultientryadjustednameother
  }%
  \@glxstrmultientryadjustednamepresep{\@mgls@previouslabel}{\mglscurrentmainlabel}%
  \@glxstrmultientryadjustednamefmt{#2}%
}%
\ifblank{#3}%
{%
  \let\@mgls@previouslabel\mglscurrentmainlabel
  \let\@gls@xtrmultientryadjustednamesep\@glxstrmultientryadjustednamepostsep
  \@for\mglscurrentlabel:=#3\do{%
    \@glxstrmultientryadjustednamesep{\@mgls@previouslabel}{\mglscurrentlabel}%
    \@glxstrmultientryadjustednameother{\mglscurrentlabel}%
    \let\@mgls@previouslabel\mglscurrentlabel
    \let\@gls@xtrmultientryadjustednamesep\@glxstrmultientryadjustednamesep
  }%
}%
}
\newcommand*{\glxstrmultientryadjustednamesep}{\glscombinedfirstsepfirst}
\newcommand*{\glxstrmultientryadjustednamepresep}{\glxstrmultientryadjustednamesep}
\newcommand*{\glxstrmultientryadjustednamepostsep}{\glxstrmultientryadjustednamesep}
\newcommand*{\glxstrmultientryadjustednamefmt}[1]{#1}
\newcommand*{\glxstrmultientryadjustednameother}[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}}}

```

```

\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*{\glxtrcontrolrules}{%
  \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*{\glxtrspacerules}{%
\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*{\glxtrnonprintablerules}{%
\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*{\glxtrcombiningdiacriticrules}{%
\glxtrcombiningdiacriticIrules\string;
\glxtrcombiningdiacriticIIrules\string;
\glxtrcombiningdiacriticIIIrules\string;
\glxtrcombiningdiacriticIVrules
}
\newcommand*{\glxtrcombiningdiacriticIrules}{%
\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*{\glxtrcombiningdiacriticIIrules}{%

```

```

\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*{\glxtrcombiningdiacriticIIIrules}{%
\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 diaeresis 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\string;% combining Cyrillic psili pneumata
}
\newcommand*{\glxtrcombiningdiacriticIVrules}{%
\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\string;% combining left right arrow above
}
\newcommand*{\glxtrhyphenrules}{%
\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*{\glxtrgeneralpuncrules}{%
\glxtrgeneralpuncIrules
\string<\glxtrcurrencyrules
\string<\glxtrgeneralpuncIIrules
}

```

```

\newcommand*{\glxtrgeneralpuncIrules}{%
\string'\glshex 005F\string'% underscore
\string<\glshex 00AF\string'% 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\string'% inverted exclamation mark
\string<\string'\glshex 003F\string'% question mark
\string<\glshex 00BF\string'% inverted question mark
\string<\string'\glshex 002F\string'% solidus
\string<\string'\glshex 002E\string'% full stop
\string<\glshex 00B4\string'% acute accent
\string<\string'\glshex 0060\string'% grave accent
\string<\string'\glshex 005E\string'% circumflex accent
\string<\glshex 00A8\string'% diaeresis
\string<\string'\glshex 007E\string'% tilde
\string<\glshex 00B7\string'% middle dot
\string<\glshex 00B8\string'% cedilla
\string<\string'\glshex 0027\string'% straight apostrophe
\string<\string'\glshex 0022\string'% straight double quote
\string<\glshex 00AB\string'% left guillemet
\string<\glshex 00BB\string'% right guillemet
\string<\string'\glshex 0028\string'% left parenthesis
\string=>\glshex 207D\string=>\glshex 208D\string'% super/subscript left parenthesis
\string<\string'\glshex 0029\string'% right parenthesis
\string=>\glshex 207E\string=>\glshex 208E\string'% 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\string'% section sign
\string<\glshex 00B6\string'% pilcrow sign
\string<\glshex 00A9\string'% copyright sign
\string<\glshex 00AE\string'% registered sign
\string<\string'\glshex 0040\string'% at sign
}
\newcommand*{\glxtrcurrencyrules}{%
\glshex 00A4\string'% currency sign
\string<\glshex 0E3F\string'% Thai currency symbol baht
\string<\glshex 00A2\string'% cent sign
\string<\glshex 20A1\string'% colon sign
\string<\glshex 20A2\string'% cruzeiro sign
\string<\string'\glshex 0024\string'% dollar sign
\string<\glshex 20AB\string'% dong sign
\string<\glshex 20AC\string'% euro sign
\string<\glshex 20A3\string'% French franc sign
\string<\glshex 20A4\string'% lira sign
\string<\glshex 20A5\string'% mill sign
\string<\glshex 20A6\string'% 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*{\glxtrgeneralpuncIrules}{%
\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*{\glxtrGeneralLatinIrules}{%
\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string<\glxtrLatinT
\string<u,U%
\string<v,V%

```

```

\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z
}
\newcommand*{\glxtrGeneralLatinIIrules}{%
\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SS \string, \glxtrLatinEszettSs
\string<\glxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glxtrGeneralLatinIIIrules}{%
\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM

```



```

\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SZ, \glxtrLatinEszettSz
\string<\glxtrLatinT
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glxtrGeneralLatinIVrules}{%
\glxtrLatinA
\string& AE , \glxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string& OE , \glxtrLatinOELigature
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SS , \glxtrLatinEszettSs
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glxtrGeneralLatinVrules}{%

```

```

\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SS , \glxtrLatinEszettSs
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glxtrGeneralLatinVirules}{%
\glxtrLatinA
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinP
\string<q,Q%

```

```

\string<r,R%
\string<\glxtrLatinS
\string& SZ , \glxtrLatinEszettSz
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glxtrGeneralLatinVIIrules}{%
\glxtrLatinA
\string<\glxtrLatinAELigature
\string<b,B%
\string<c,C%
\string<d,D%
\string<\glxtrLatinEth
\string<\glxtrLatinE
\string<f,F%
\string<\glxtrLatinInsularG
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glxtrLatinL
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glxtrLatinO
\string<\glxtrLatinOELigature
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glshex 017F=\glxtrLatinS % s and long s
\string<\glxtrLatinT
\string<\glxtrLatinThorn
\string<u,U%
\string<v,V%
\string< w\string=\glshex 01BF, W\string=\glshex 01F7
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glxtrGeneralLatinVIIIrules}{%
\glxtrLatinA
\string& AE , \glxtrLatinAELigature
\string<b,B%
\string<c,C%

```

```

\string<\glshex 00F0\string;d,\glshex 00D0\string;D% D and eth
\string<\glxtrLatinE
\string<f,F%
\string<g,G%
\string<\glxtrLatinH
\string<\glxtrLatinI
\string<j,J%
\string<\glxtrLatinK
\string<\glshex 0142\string=\glxtrLatinL\string=\glshex 0141% L and \L
\string<\glxtrLatinM
\string<\glxtrLatinN
\string<\glshex 00F8\string=\glxtrLatinO\string=\glshex 00D8% O and \O
\string& OE , \glxtrLatinOELigature
\string<\glxtrLatinP
\string<q,Q%
\string<r,R%
\string<\glxtrLatinS
\string& SS , \glxtrLatinEszettSs
\string<\glxtrLatinT
\string& th =\glshex 00DE
\string& TH =\glshex 00FE
\string<u,U%
\string<v,V%
\string<w,W%
\string<\glxtrLatinX
\string<y,Y%
\string<z,Z%
}
\newcommand*{\glxtrLatinA}{%
  a\string=\glshex 00AA\string=\glshex 2090,A
}
\newcommand*{\glxtrLatinE}{%
  e\string=\glshex 2091,E
}
\newcommand*{\glxtrLatinH}{%
  h\string=\glshex 2095,H
}
\newcommand*{\glxtrLatinI}{%
  i\string=\glshex 2071,I
}
\newcommand*{\glxtrLatinK}{%
  k\string=\glshex 2096,K
}
\newcommand*{\glxtrLatinL}{%
  l\string=\glshex 2097,L
}
\newcommand*{\glxtrLatinM}{%
  m\string=\glshex 2098,M
}
\newcommand*{\glxtrLatinN}{%

```

```

    n\string=\glshex 207F\string=\glshex 2099,N
}
\newcommand*{\glxtrLatinO}{%
    o\string=\glshex 00BA\string=\glshex 2092,O
}
\newcommand*{\glxtrLatinP}{%
    p\string=\glshex 209A,P
}
\newcommand*{\glxtrLatinS}{%
    s\string=\glshex 209B,S
}
\newcommand*{\glxtrLatinT}{%
    t\string=\glshex 209C,T
}
\newcommand*{\glxtrLatinX}{%
    x\string=\glshex 2093,X
}
\newcommand*{\glxtrLatinSchwa}{%
    \glshex 0259\string=\glshex 2094,\glshex 018F
}
\newcommand*{\glxtrLatinEszettSs}{%
    \glshex 00DF% eszett
    \string=\glshex 017Fs % long S s
}
\newcommand*{\glxtrLatinEszettSz}{%
    \glshex 00DF% eszett
    \string= \glshex 017Fz % long S z
}
\newcommand*{\glxtrLatinEth}{%
    \glshex 00F0,\glshex 00D0% eth
}
\newcommand*{\glxtrLatinThorn}{%
    \glshex 00FE,\glshex 00DE% thorn
}
\newcommand*{\glxtrLatinAELigature}{%
    \glshex 00E6,\glshex 00C6% AE-ligature
}
\newcommand*{\glxtrLatinOELigature}{%
    \glshex 0153,\glshex 0152% OE-ligature
}
\newcommand*{\glxtrLatinAA}{%
    \glshex 00E5=a\glshex 030A,% \aa
    \glshex 00C5=A\glshex 030A% \AA
}
\newcommand*{\glxtrLatinWynn}{%
    \glshex 01BF,\glshex 01F7% wynn
}
\newcommand*{\glxtrLatinInsularG}{%
    \glshex 1D79,\glshex A77D% insular G
    \string; g, G
}

```

```

}
\newcommand*{\glxtrLatinOslash}{%
  \glshex 00F8,\glshex 00D8% \o, \O
}
\newcommand*{\glxtrLatinLslash}{%
  \glshex 0142,\glshex 0141% \l, \L
}
\newcommand*{\glxtrMathUpGreekIrules}{%
  \glxtrUpAlpha
  \string<\glxtrUpBeta
  \string<\glxtrUpGamma
  \string<\glxtrUpDelta
  \string<\glxtrUpEpsilon
  \string<\glxtrUpDigamma
  \string<\glxtrUpZeta
  \string<\glxtrUpEta
  \string<\glxtrUpTheta
  \string<\glxtrUpIota
  \string<\glxtrUpKappa
  \string<\glxtrUpLambda
  \string<\glxtrUpMu
  \string<\glxtrUpNu
  \string<\glxtrUpXi
  \string<\glxtrUpOmicron
  \string<\glxtrUpPi
  \string<\glxtrUpRho
  \string<\glxtrUpSigma
  \string<\glxtrUpTau
  \string<\glxtrUpUpsilon
  \string<\glxtrUpPhi
  \string<\glxtrUpChi
  \string<\glxtrUpPsi
  \string<\glxtrUpOmega
}
\newcommand*{\glxtrMathUpGreekIIrules}{%
  \glxtrUpAlpha
  \string<\glxtrUpBeta
  \string<\glxtrUpGamma
  \string<\glxtrUpDelta
  \string<\glxtrUpEpsilon
  \string<\glxtrUpZeta
  \string<\glxtrUpEta
  \string<\glxtrUpTheta
  \string<\glxtrUpIota
  \string<\glxtrUpKappa
  \string<\glxtrUpLambda
  \string<\glxtrUpMu
  \string<\glxtrUpNu
  \string<\glxtrUpXi
  \string<\glxtrUpOmicron

```

```

\string<\glxtrUpPi
\string<\glxtrUpRho
\string<\glxtrUpSigma
\string<\glxtrUpTau
\string<\glxtrUpUpsilon
\string<\glxtrUpPhi
\string<\glxtrUpChi
\string<\glxtrUpPsi
\string<\glxtrUpOmega
}
\newcommand*{\glxtrMathItalicGreekIrules}{%
\glxtrMathItalicAlpha
\string<\glxtrMathItalicBeta
\string<\glxtrMathItalicGamma
\string<\glxtrMathItalicDelta
\string<\glxtrMathItalicEpsilon
\string<\glxtrUpDigamma
\string<\glxtrMathItalicZeta
\string<\glxtrMathItalicEta
\string<\glxtrMathItalicTheta
\string<\glxtrMathItalicIota
\string<\glxtrMathItalicKappa
\string<\glxtrMathItalicLambda
\string<\glxtrMathItalicMu
\string<\glxtrMathItalicNu
\string<\glxtrMathItalicXi
\string<\glxtrMathItalicOmicron
\string<\glxtrMathItalicPi
\string<\glxtrMathItalicRho
\string<\glxtrMathItalicSigma
\string<\glxtrMathItalicTau
\string<\glxtrMathItalicUpsilon
\string<\glxtrMathItalicPhi
\string<\glxtrMathItalicChi
\string<\glxtrMathItalicPsi
\string<\glxtrMathItalicOmega
}
\newcommand*{\glxtrMathItalicGreekIIrules}{%
\glxtrMathItalicAlpha
\string<\glxtrMathItalicBeta
\string<\glxtrMathItalicGamma
\string<\glxtrMathItalicDelta
\string<\glxtrMathItalicEpsilon
\string<\glxtrMathItalicZeta
\string<\glxtrMathItalicEta
\string<\glxtrMathItalicTheta
\string<\glxtrMathItalicIota
\string<\glxtrMathItalicKappa
\string<\glxtrMathItalicLambda
\string<\glxtrMathItalicMu

```

```

\string<\glxstrMathItalicNu
\string<\glxstrMathItalicXi
\string<\glxstrMathItalicOmicron
\string<\glxstrMathItalicPi
\string<\glxstrMathItalicRho
\string<\glxstrMathItalicSigma
\string<\glxstrMathItalicTau
\string<\glxstrMathItalicUpsilon
\string<\glxstrMathItalicPhi
\string<\glxstrMathItalicChi
\string<\glxstrMathItalicPsi
\string<\glxstrMathItalicOmega
}
\newcommand*{\glxstrMathItalicUpperGreekIrules}{%
\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*{\glxstrMathItalicUpperGreekIIrules}{%
\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*{\glxtrMathItalicLowerGreekIrules}{%
\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*{\glxtrMathItalicLowerGreekIrules}{%
\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*{\glxtrMathGreekIrules}{%
\glxtrMathItalicAlpha
\string;\glxtrUpAlpha
\string<\glxtrMathItalicBeta
\string;\glxtrUpBeta
\string<\glxtrMathItalicGamma
\string;\glxtrUpGamma
\string<\glxtrMathItalicDelta
\string;\glxtrUpDelta
\string<\glxtrMathItalicEpsilon
\string;\glxtrUpEpsilon
\string<\glxtrUpDigamma
\string<\glxtrMathItalicZeta
\string;\glxtrUpZeta
\string<\glxtrMathItalicEta

```

```

\string;\glxtrUpEta
\string<\glxtrMathItalicTheta
\string;\glxtrUpTheta
\string<\glxtrMathItalicIota
\string;\glxtrUpIota
\string<\glxtrMathItalicKappa
\string;\glxtrUpKappa
\string<\glxtrMathItalicLambda
\string;\glxtrUpLambda
\string<\glxtrMathItalicMu
\string;\glxtrUpMu
\string<\glxtrMathItalicNu
\string;\glxtrUpNu
\string<\glxtrMathItalicXi
\string;\glxtrUpXi
\string<\glxtrMathItalicOmicron
\string;\glxtrUpOmicron
\string<\glxtrMathItalicPi
\string;\glxtrUpPi
\string<\glxtrMathItalicRho
\string;\glxtrUpRho
\string<\glxtrMathItalicSigma
\string;\glxtrUpSigma
\string<\glxtrMathItalicTau
\string;\glxtrUpTau
\string<\glxtrMathItalicUpsilon
\string;\glxtrUpUpsilon
\string<\glxtrMathItalicPhi
\string;\glxtrUpPhi
\string<\glxtrMathItalicChi
\string;\glxtrUpChi
\string<\glxtrMathItalicPsi
\string;\glxtrUpPsi
\string<\glxtrMathItalicOmega
\string;\glxtrUpOmega
}
\newcommand*{\glxtrMathGreekIIrules}{%
\glxtrMathItalicAlpha
\string;\glxtrUpAlpha
\string<\glxtrMathItalicBeta
\string;\glxtrUpBeta
\string<\glxtrMathItalicGamma
\string;\glxtrUpGamma
\string<\glxtrMathItalicDelta
\string;\glxtrUpDelta
\string<\glxtrMathItalicEpsilon
\string;\glxtrUpEpsilon
\string<\glxtrMathItalicZeta
\string;\glxtrUpZeta
\string<\glxtrMathItalicEta

```

```

\string;\glxtrUpEta
\string<\glxtrMathItalicTheta
\string;\glxtrUpTheta
\string<\glxtrMathItalicIota
\string;\glxtrUpIota
\string<\glxtrMathItalicKappa
\string;\glxtrUpKappa
\string<\glxtrMathItalicLambda
\string;\glxtrUpLambda
\string<\glxtrMathItalicMu
\string;\glxtrUpMu
\string<\glxtrMathItalicNu
\string;\glxtrUpNu
\string<\glxtrMathItalicXi
\string;\glxtrUpXi
\string<\glxtrMathItalicOmicron
\string;\glxtrUpOmicron
\string<\glxtrMathItalicPi
\string;\glxtrUpPi
\string<\glxtrMathItalicRho
\string;\glxtrUpRho
\string<\glxtrMathItalicSigma
\string;\glxtrUpSigma
\string<\glxtrMathItalicTau
\string;\glxtrUpTau
\string<\glxtrMathItalicUpsilon
\string;\glxtrUpUpsilon
\string<\glxtrMathItalicPhi
\string;\glxtrUpPhi
\string<\glxtrMathItalicChi
\string;\glxtrUpChi
\string<\glxtrMathItalicPsi
\string;\glxtrUpPsi
\string<\glxtrMathItalicOmega
\string;\glxtrUpOmega
}
\newcommand*{\glxtrUpAlpha}{%
\glshex 03B1,% lower case alpha
\glshex 0391% upper case alpha
}
\newcommand*{\glxtrUpBeta}{%
\glshex 03B2,% lower case beta
\glshex 0392% upper case beta
}
\newcommand*{\glxtrUpGamma}{%
\glshex 03B3,% lower case gamma
\glshex 0393% upper case gamma
}
\newcommand*{\glxtrUpDelta}{%
\glshex 03B4,% lower case delta

```

```

\glshex 0394% upper case delta
}
\newcommand*{\glstrUpEpsilon}{%
\glshex 03B5% lower case epsilon
\string=\glshex 03F5,% lower case epsilon variant
\glshex 0395% upper case epsilon
}
\newcommand*{\glstrUpDigamma}{%
\glshex 03DD,% lower case digamma
\glshex 03DC% upper case digamma
}
\newcommand*{\glstrUpZeta}{%
\glshex 03B6,% lower case zeta
\glshex 0396% upper case zeta
}
\newcommand*{\glstrUpEta}{%
\glshex 03B7,% lower case eta
\glshex 0397% upper case eta
}
\newcommand*{\glstrUpTheta}{%
\glshex 03B8% lower case theta
\string=\glshex 03D1,% lower case theta variant
\glshex 0398% upper case theta
}
\newcommand*{\glstrUpIota}{%
\glshex 03B9,% lower case iota
\glshex 0399% upper case iota
}
\newcommand*{\glstrUpKappa}{%
\glshex 03BA% lower case kappa
\string=\glshex 03F0,% lower case kappa variant
\glshex 039A% upper case kappa
}
\newcommand*{\glstrUpLambda}{%
\glshex 03BB,% lower lambda
\glshex 039B% upper case lambda
}
\newcommand*{\glstrUpMu}{%
\glshex 03BC,% lower case mu
\glshex 039C% upper case mu
}
\newcommand*{\glstrUpNu}{%
\glshex 03BD,% lower case nu
\glshex 039D% upper case nu
}
\newcommand*{\glstrUpXi}{%
\glshex 03BE,% lower case xi
\glshex 039E% upper case xi
}
\newcommand*{\glstrUpOmicron}{%

```

```

\glshex 03BF,% lower case omicron
\glshex 039F% upper case omicron
}
\newcommand*{\glxtrUpPi}{%
\glshex 03C0% lower case pi
\string=\glshex 03D6,% lower case pi variant
\glshex 03A0% upper case pi
}
\newcommand*{\glxtrUpRho}{%
\glshex 03C1% lower case rho
\string=\glshex 03F1,% lower case rho variant
\glshex 03A1% upper case rho
}
\newcommand*{\glxtrUpSigma}{%
\glshex 03C2% lower case sigma
\string=\glshex 03C3,% lower case sigma
\glshex 03A3% upper case sigma
}
\newcommand*{\glxtrUpTau}{%
\glshex 03C4,% lower case tau
\glshex 03A4% upper case tau
}
\newcommand*{\glxtrUpUpsilon}{%
\glshex 03C5,% lower case upsilon
\glshex 03A5% upper case upsilon
}
\newcommand*{\glxtrUpPhi}{%
\glshex 03C6% lower case phi
\string=\glshex 03D5,% lower case phi variant
\glshex 03A6% upper case phi
}
\newcommand*{\glxtrUpChi}{%
\glshex 03C7,% lower case chi
\glshex 03A7% upper case chi
}
\newcommand*{\glxtrUpPsi}{%
\glshex 03C8,% lower case psi
\glshex 03A8% upper case psi
}
\newcommand*{\glxtrUpOmega}{%
\glshex 03C9,% lower case omega
\glshex 03A9% upper case omega
}
\newcommand*{\glxtrMathItalicAlpha}{%
\glshex 1D6FC,% lower case alpha (maths italic)
\glshex 1D6E2% upper case alpha (maths italic)
}
\newcommand*{\glxtrMathItalicBeta}{%
\glshex 1D6FD,% lower case beta (maths italic)
\glshex 1D6E3% upper case beta (maths italic)
}

```

```

}
\newcommand*{\glstrMathItalicGamma}{%
\glshex 1D6FE,% lower case gamma (maths italic)
\glshex 1D6E4% upper case gamma (maths italic)
}
\newcommand*{\glstrMathItalicDelta}{%
\glshex 1D6FF,% lower case delta (maths italic)
\glshex 1D6E5% upper case delta (maths italic)
}
\newcommand*{\glstrMathItalicEpsilon}{%
\glshex 1D700% lower case epsilon (maths italic)
\string=\glshex 1D716,% lower case epsilon variant (maths italic)
\glshex 1D6E6% upper case epsilon (maths italic)
}
\newcommand*{\glstrMathItalicZeta}{%
\glshex 1D701,% lower case zeta (maths italic)
\glshex 1D6E7% upper case zeta (maths italic)
}
\newcommand*{\glstrMathItalicEta}{%
\glshex 1D702,% lower case eta (maths italic)
\glshex 1D6E8% upper case eta (maths italic)
}
\newcommand*{\glstrMathItalicTheta}{%
\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*{\glstrMathItalicIota}{%
\glshex 1D704,% lower case iota (maths italic)
\glshex 1D6EA% upper case iota (maths italic)
}
\newcommand*{\glstrMathItalicKappa}{%
\glshex 1D705% lower case kappa (maths italic)
\string=\glshex 1D718,% lower case kappa variant (maths italic)
\glshex 1D6EB% upper case kappa (maths italic)
}
\newcommand*{\glstrMathItalicLambda}{%
\glshex 1D706,% lower case lambda (maths italic)
\glshex 1D6EC% upper case lambda (maths italic)
}
\newcommand*{\glstrMathItalicMu}{%
\glshex 1D707,% lower case mu (maths italic)
\glshex 1D6ED% upper case mu (maths italic)
}
\newcommand*{\glstrMathItalicNu}{%
\glshex 1D708,% lower case nu (maths italic)
\glshex 1D6EE% upper case nu (maths italic)
}
\newcommand*{\glstrMathItalicXi}{%

```

```

\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*{\glxtrMathItalicNabla}{%
  \glshex 1D6FB% nabla (maths italic)
}
\newcommand*{\glxtrDigitrules}{%
  0\string=\glshex 2080\string=\glshex 2070
  \string<\glshex 2081\string=\glshex 00B9
  \string<\glshex 2082\string=\glshex 00B2
  \string<\glshex 2083\string=\glshex 00B3
  \string<\glshex 2084\string=\glshex 2074
  \string<\glshex 2085\string=\glshex 2075
  \string<\glshex 2086\string=\glshex 2076
  \string<\glshex 2087\string=\glshex 2077
  \string<\glshex 2088\string=\glshex 2078
  \string<\glshex 2089\string=\glshex 2079
}
\newcommand*{\glxtrBasicDigitrules}{%
  0\string<\glshex 2080\string<\glshex 2070
  \string<\glshex 2081\string<\glshex 00B9
  \string<\glshex 2082\string<\glshex 00B2
  \string<\glshex 2083\string<\glshex 00B3
  \string<\glshex 2084\string<\glshex 2074
  \string<\glshex 2085\string<\glshex 2075
  \string<\glshex 2086\string<\glshex 2076
  \string<\glshex 2087\string<\glshex 2077
  \string<\glshex 2088\string<\glshex 2078
  \string<\glshex 2089\string<\glshex 2079
}
\newcommand*{\glxtrSubScriptDigitrules}{%
  \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*{\glxtrSuperScriptDigitrules}{%
  \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*{\glxtrFractionrules}{%
  \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 ( $1/6 \sim 0.167$ )
\string<\glshex 2155% one fifth ( $1/5 = 0.2$ )
\string<\glshex 00BC% one quarter ( $1/4 = 0.25$ )
\string<\glshex 2153% one third ( $1/3 \sim 0.333$ )
\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 ( $2/3 \sim 0.667$ )
\string<\glshex 00BE% three quarters ( $3/4 = 0.75$ )
\string<\glshex 2158% four fifths ( $4/5 = 0.8$ )
\string<\glshex 215A% five sixths ( $5/6 \sim 0.833$ )
\string<\glshex 215E% seven eighths ( $7/8 = 0.875$ )
}
\renewcommand{\@glxtrdialecthook}{%
  \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\glxtr@loaddialect
{%
  \@ifpackageloaded{tracklang}
  {%
    \AnyTrackedLanguages
    {%
      \ForEachTrackedDialect{\this@dialect}{\glxtr@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*{\@glxtr@loadstyles}{}
\DeclareOption{all}{%
  \appto\@glxtr@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\@glxtr@loadstyles{%
    \noexpand\RequirePackage{glossary-\CurrentOption}}}%
  {%
    \PackageError{glossaries-extra-styles}%
    {Unknown option ‘\CurrentOption’}{}%
  }%
}
\ProcessOptions
\@glxtr@loadstyles
\providecommand*{\glxtrprelocation}{\space}
\providecommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glstyle@#1}%
  {%
    \PackageError{glossaries-extra}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glstyle@#1}{#2}%
  }%
}
\ifdef{\@glstyle@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}%
}
}
{%
}
\ifdef{\@glsstyle@list}
{%
  \newcommand{\glslistprelocation}{\glstxtrprelocation}
  \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}{%

```

```

\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{\glssetgrouptitle{##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
    }
  }
}

```

```

        \glstrprelocation ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
        \glstrprelocation ##3\tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}
}
{}
\ifcsdef{@glstyle@long3col}
{%
    \renewglossarystyle{long3col}{%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glsdscwidth}p{\glspagelistwidth}}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{}%
        \renewcommand*{\glsgroupheading}[1]{}%
        \renewcommand{\glossentry}[2]{%
            \glssubentryitem{##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{@glstyle@long4col}
{%
    \renewglossarystyle{long4col}{%
        \renewenvironment{theglossary}%
            {\begin{longtable}{llll}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{}%
        \renewcommand*{\glsgroupheading}[1]{}%
    }
}

```

```

\renewcommand{\glossentry}[2]{%
  \glentryitem{##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]{%
      \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription\glstrprelocation ##2%
      \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}%
      \glspostdescription\glstrprelocation ##3%
      \tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
  }
}
{}
\ifcsdef{@glsstyle@longragged3col}
{%

```



```

\renewglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& &\tabularnewline}%
  \fi
}
{}
\ifcsdef{@glstyle@altlongragged4col}
{%
  \renewglossarystyle{altlongragged4col}{%
    \renewenvironment{theglossary}%
      {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}l%
        >{\raggedright}p{\glspagelistwidth}}}%
      {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glentryitem{##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
    }%
    \ifglsgroupskip
      \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
    }%
    \ifglsgroupskip
      \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
    }%
  }
}

```

```

\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{ & &\tabularnewline}%
\fi
}
}
{}
\ifcsdef{@glstyle@super4col}
{%
\renewglossarystyle{super4col}{%
\renewenvironment{theglossary}%
{\tablehead{}\tabletail{}}%
\begin{supertabular}{1111}{}%
\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glentryitem{##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
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& &\tabularnewline}%
\fi
}
}
{}
\ifcsdef{@glstyle@superragged}
{%
\renewglossarystyle{superragged}{%
\renewenvironment{theglossary}%
{\tablehead{}\tabletail{}}%
\begin{supertabular}{1>\raggedright}p{\glstylewidth}}%
\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1}\glspostdescription\glstrprelocation ##2%
\tabularnewline
}%

```

```

\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription
  \glstrprelocation ##3%
  \tabularnewline
}%
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
}
{}
\ifcsdef{@glstyle@superragged3col}
{%
  \renewglossarystyle{superragged3col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}}%
      \begin{supertabular}{l>{\raggedright}p{\glswidth}%
        >{\raggedright}p{\glspagelistwidth}}}%
      {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1}\glspostdescription &
      ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription &
      ##3\tabularnewline
    }%
    \ifglsgroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
  }
}
{}
\ifcsdef{@glstyle@altsuperragged4col}
{%
  \renewglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
      {\tablehead{}\tabletail{}}%
      \begin{supertabular}{l>{\raggedright}p{\glswidth}l%

```

```

>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand{\glossentry}[2]{%
  \glssentryitem{##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{\@glstyle@inline}
{%
  \renewcommand*{\glspostinline}{.\spacefactor\sfcode'\.}
  \renewcommand*{\glsinlinedescformat}[3]{%
    \space#1\glstrpostdescription}
  \renewcommand*{\glsinlinesubdescformat}[3]{%
    #1\glstrpostdescription}
}
{}
\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{\@glstyle@index}
{
  \newcommand*{\glstreeprelocation}{\glstrprelocation}
  \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\glssentryitem{##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}{\ifglsgnogroupskip\else\glstreegroupskip\fi}%
}
{}
\ifdef{\@glsstyle@indexgroup}
{%
    \renewglossarystyle{indexgroup}{%
        \setglossarystyle{index}%
        \renewcommand*{\glsgroupheading}[1]{%
            \glstrgetgrouptitle{##1}{\glstr@grptitle}%
            \glstreePreHeader{##1}{\glstr@grptitle}%
            \item\glstreegroupheaderfmt{\glstr@grptitle}%
            \glstreegroupheaderskip\@afterheading
        }%
    }
}
{}
\ifdef{\@glsstyle@indexhypergroup}
{%
    \renewglossarystyle{indexhypergroup}{%
        \setglossarystyle{index}%
        \renewcommand*{\glossaryheader}{%
            \item\glstreenavigationfmt{\glstr@navigation}%
            \glstreegroupheaderskip\@afterheading}%
    }
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \glstrgetgrouptitle{##1}{\glstr@grptitle}%
  \glstreePreHeader{##1}{\glstr@grptitle}%
  \item\glstreegroupheaderfmt
    {\glshnavhypertarget{##1}{\glstr@grptitle}}%
  \glstreegroupheaderskip\@afterheading}%
}%
}
{}
\ifdef{\@glstyle@tree}
{%
\newcommand{\glstxtrtreepredesc}{\glstreepredesc}
\newcommand{\glstxtrtreechildpredesc}{\glstreechildpredesc}
\newcommand{\glstreedesc}[1]{%
  \glstxtrtreepredesc\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]{%
  \glstxtrtreechildpredesc\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

```

```

\glstentryitem{##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}{\ifglsgnognogroupskip\else\glstreegroupskip\fi}%
}%
}
{}
\ifdef{\@glsstyle@treegroup}
{%
\renewglossarystyle{treegroup}{%
\setglossarystyle{tree}%
\renewcommand{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent\glstreegroupheaderfmt{\glstr@grptitle}%
\glstreegroupheaderskip\@afterheading}%
}
}
{}
\ifdef{\@glsstyle@treehypergroup}
{%
\renewglossarystyle{treehypergroup}{%
\setglossarystyle{tree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsgnavigation}%
\glstreegroupheaderskip\@afterheading}%
\renewcommand*{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent
\glstreegroupheaderfmt
{\glsgnavhypertarget{##1}{\glstr@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
\glstentryitem{##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{\@glstyle@treenonamegroup}
{%
\renewglossarystyle{treenonamegroup}{%
\setglossarystyle{treenoname}%
\renewcommand{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glxtr@grptitle}%
\glstreePreHeader{##1}{\glxtr@grptitle}%
\par\noindent\glstreegroupheaderfmt{\glxtr@grptitle}%
\glstreegroupheaderskip\@afterheading
}
}

```

```

    }%
  }
}
{}
\ifdef{\@glsstyle@treenonamehypergroup}
{%
  \renewglossarystyle{treenonamehypergroup}{%
    \setglossarystyle{treenoname}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip\@afterheading}%
    \renewcommand*{\glsgroupheading}[1]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@grptitle}}%
      \glstreegroupheaderskip\@afterheading}%
    }
  }
}
{}
\ifdef{\@glsstyle@alttree}
{%
  \newcommand{\glsalttreepredesc}{}
  \newcommand{\glsalttreechildpredesc}{\glsalttreepredesc}
  \newcommand{\glstralttreeSymbolDescLocation}[2]{%
    {%
      \let\par\glstrAltTreePar
      \let\glstrtreepredesc\glsalttreepredesc
      \let\glstrtreechildpredesc\glsalttreechildpredesc
      \ifglshassymbol{#1}{(\glossentrysymbol{#1})\space}{}%
      \glstreeDescLoc{#1}{#2}\par
    }%
  }
}
\newlength\glstrAltTreeIndent
\newcommand{\glstrAltTreePar}{%
  \@par
  \glstrAltTreeSetHangIndent
  \setlength{\parindent}{\dimexpr\hangindent+\glstrAltTreeIndent}%
}
\newcommand{\glstralttreeSubSymbolDescLocation}[3]{%
  \glstralttreeSymbolDescLocation{#2}{#3}%
}
\newlength\glstrtreetopindent
\newcommand*{\glstralttreeInit}{%
  \settowidth{\glstrtreetopindent}{\glstreenamefmt{\glsgetwidestname\space}}%
  \glstrAltTreeIndent=\parindent
}
\newcommand*{\glssetwidest}[2][0]{%
  \csgdef{@glswidestname\romannumeral#1}{#2}%
}

```

```

\newcommand*{\eglssetwidest}[2][0]{%
  \protected@csedef{@glswidestname\romannumeral#1}{#2}%
}
\newcommand*{\xglsetwidest}[2][0]{%
  \protected@csxdef{@glswidestname\romannumeral#1}{#2}%
}
\newcommand*{\glupdatewidest}[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*{\gglupdatewidest}[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*{\xglupdatewidest}[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@}%
            {\glstreenamfmt{\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@}%
          {\glstreenamfmt{\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@}%
        {\glstreenamfmt{\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}%
          {\glstreenamfmt{\glsentryname{\@glo@label}}}%
        \ifdim\gls@tmplen>\dimen@ii
          \dimen@ii=\gls@tmplen
          \eglssetwidest[2]{\glsentryname{\@glo@label}}%
        \fi
      }%
    }%
    {%
      \settowidth{\gls@tmplen}%
        {\glstreenamfmt{\glsentryname{\@glo@label}}}%
      \ifdim\gls@tmplen>\dimen@i
        \dimen@i=\gls@tmplen
        \eglssetwidest[1]{\glsentryname{\@glo@label}}%
      \fi
    }%
  }%
}

```

```

        \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
  \foralllglossaries[#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}%
  {%
    \forglentries[\@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}%
  {%
    \forglentries[\@gls@type]{\@glo@label}%
    {%
      \settowidth{\dimen@}%
        {\glstreenamefmt{\glsentryname{\@glo@label}}}%
      \ifdim\dimen@>\gls@tmplen
        \gls@tmplen=\dimen@
        \eglssetwidest{\glsentryname{\@glo@label}}%
      \fi
    }%
  }%
}

```

```

\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}%
{%
\for glsentries[\@gls@type]{\@glo@label}%
{%
\if glsused{\@glo@label}%
{%
\settowidth{\dimen@}%
{\glstreenamfmt{\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}%
{%
\for glsentries[\@gls@type]{\@glo@label}%
{%

```



```

\settowidth{\dimen@}%
{\glstreenamefmt{\glstryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\eglssetwidest{\glstryname{\@glo@label}}%
\fi
\settowidth{\dimen@}%
{\glstrysymbol{\@glo@label}}%
\ifdim\dimen@>#2\relax
#2=\dimen@
\fi
\settowidth{\dimen@}%
{\GlsXtrFormatLocationList{\glstrynumberlist{\@glo@label}}}%
\ifdim\dimen@>#3\relax
#3=\dimen@
\fi
}%
}%
}
\newrobustcmd*{\glsFindWidestUsedAnyNameLocation}[2][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
#2=0pt\relax
\forall glossaries[#1]{\@gls@type}%
{%
\forall glsentries[\@gls@type]{\@glo@label}%
{%
\ifglssused{\@glo@label}%
{%
\settowidth{\dimen@}%
{\glstreenamefmt{\glstryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\eglssetwidest{\glstryname{\@glo@label}}%
\fi
\settowidth{\dimen@}%
{\GlsXtrFormatLocationList{\glstrynumberlist{\@glo@label}}}%
\ifdim\dimen@>#2\relax
#2=\dimen@
\fi
}%
}%
}%
}%
}
\newrobustcmd*{\glsFindWidestAnyNameLocation}[2][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
#2=0pt\relax
\forall glossaries[#1]{\@gls@type}%

```

```

{%
\for glsentries [\@gls@type]{\@glo@label}%
{%
\settowidth{\dimen@}%
{\glstreenamefmt{\glstryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\eglssetwidest{\glstryname{\@glo@label}}%
\fi
\settowidth{\dimen@}%
{\GlsXtrFormatLocationList{\glstrynumberlist{\@glo@label}}}%
\ifdim\dimen@>#2\relax
#2=\dimen@
\fi
}%
}%
}
\newcommand*{\glxtrComputeTreeIndent}[1]{%
\glstreeindent=\glxtrtreetopindent\relax
}
\newcommand*{\glxtrComputeTreeSubIndent}[3]{%
\ifcsundef{\glswidestname\romannumeral#1}%
{%
\settowidth{#3}{\glstreenamefmt{\@glswidestname\space}}%
}%
{%
\settowidth{#3}{\glstreenamefmt{%
\csname @glswidestname\romannumeral#1\endcsname\space}}%
}%
}
}
\newcommand*{\glxtrAltTreeSetHangIndent}{\hangindent\glstreeindent}
\newcommand*{\glxtrAltTreeSetSubHangIndent}[1]{\hangindent\glstreeindent}
\renewglossarystyle{alttree}{%
\renewenvironment{theglossary}%
{%
\glxtralttreeInit
\def\@gls@prevlevel{-1}%
\mbox{}\par}%
{\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glosstry}[2]{%
\ifnum\@gls@prevlevel=0\relax
\else
\glxtrComputeTreeIndent{##1}%
\fi
\parindent\glstreeindent
\glxtrAltTreeSetHangIndent
\makebox[0pt][r]%
}%

```

```

\glstreenamebox{\glstreeindent}%
{%
  \glstryitem{##1}%
  \glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
}%
}%
\glxtralttreeSymbolDescLocation{##1}{##2}%
\def\@gls@prevlevel{0}%
}
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \glxtrComputeTreeSubIndent{##1}{##2}{\gls@tmplen}%
    \ifnum\@gls@prevlevel<##1\relax
      \setlength\glstreeindent\gls@tmplen
      \addtolength\glstreeindent\parindent
      \parindent\glstreeindent
    \else
      \ifnum\@gls@prevlevel=0\relax
        \glxtrComputeTreeIndent{##2}%
      \else
        \glxtrComputeTreeSubIndent{\@gls@prevlevel}{##2}{\glstreeindent}%
      \fi
      \addtolength\parindent{-\glstreeindent}%
      \setlength\glstreeindent\parindent
    \fi
  \fi
  \glxtrAltTreeSetSubHangIndent{##1}%
  \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
  \glxtralttreeSubSymbolDescLocation{##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}%
      \hangindentOpt\relax
      \parindentOpt\relax
      \glxtrgetgrouptitle{##1}{\glxtr@grptitle}%

```

```

        \glstreePreHeader{##1}{\glstr@grptitle}%
        \glstreegroupheaderfmt{\glstr@grptitle}%
        \glstreegroupheaderskip
    }%
}%
}%
{%
}
\ifdef{\@glsstyle@almtreehypergroup}
{%
    \renewglossarystyle{almtreehypergroup}{%
        \setglossarystyle{almtree}%
        \renewcommand*{\glossaryheader}{%
            \par
            \def\@gls@prevlevel{-1}%
            \hangindentOpt\relax
            \parindentOpt\relax
            \glstreenavigationfmt{\glsnavigation}%
            \glstreegroupheaderskip
        }%
        \renewcommand*{\glsgroupheading}[1]{%
            \glstrgetgrouptitle{##1}{\glstr@grptitle}%
            \glstreePreHeader{##1}{\glstr@grptitle}%
            \par
            \def\@gls@prevlevel{-1}%
            \hangindentOpt\relax
            \parindentOpt\relax
            \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@grptitle}}%
            \glstreegroupheaderskip
        }%
    }
}%
{%
}
\ifdef{\@glsstyle@mcolindexgroup}
{%
    \renewglossarystyle{mcolindexgroup}{%
        \setglossarystyle{mcolindex}%
        \renewcommand*{\glsgroupheading}[1]{%
            \glstrgetgrouptitle{##1}{\glstr@grptitle}%
            \glstreePreHeader{##1}{\glstr@grptitle}%
            \item\glstreegroupheaderfmt{\glstr@grptitle}%
            \glstreegroupheaderskip\@afterheading
        }%
    }
}%
{%
}
\ifdef{\@glsstyle@mcolindexhypergroup}
{%

```

```

\renewglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}%
    \glstreegroupheaderskip\@afterheading
  }%
  \renewcommand*{\glsgroupheading}[1]{%
    \glstrgetgrouptitle{##1}{\glstr@grptitle}%
    \glstreePreHeader{##1}{\glstr@grptitle}%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glstr@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]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
      \item\glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glstr@grptitle}}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcoltreegroup}
{%
  \renewglossarystyle{mcoltreegroup}{%
    \setglossarystyle{mcoltree}%
    \renewcommand{\glsgroupheading}[1]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
      \par\noindent\glstreegroupheaderfmt{\glstr@grptitle}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}

```

```

}%
{%
}
\ifdef{\@glsstyle@mcoltreehypergroup}
{%
  \renewglossarystyle{mcoltreehypergroup}{%
    \setglossarystyle{mcoltree}%
    \renewcommand*{\glossaryheader}{%
      \par\noindent\glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }%
    \renewcommand*{\glsgroupheading}[1]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@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]{%
      \glstrgetgrouptitle{##1}{\glstr@grptitle}%
      \glstreePreHeader{##1}{\glstr@grptitle}%
      \par\noindent
      \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@grptitle}}%
      \glstreegroupheaderskip\@afterheading
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcoltreenamegroup}
{%
  \renewglossarystyle{mcoltreenamegroup}{%
    \setglossarystyle{mcoltreename}%
    \renewcommand{\glsgroupheading}[1]{%

```

```

\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent\glstreegroupheaderfmt{\glstr@grptitle}%
\glstreegroupheaderskip\@afterheading
}%
}
}%
{%
}
\ifdef{\@glsstyle@mcoltreenamehypergroup}
{%
\renewglossarystyle{mcoltreenamehypergroup}{%
\setglossarystyle{mcoltreename}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}%
\glstreegroupheaderskip
}%
\renewcommand*{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@grptitle}}%
\glstreegroupheaderskip\@afterheading}%
}
}%
{%
}
\ifdef{\@glsstyle@mcoltreenamepannav}
{%
\renewglossarystyle{mcoltreenamepannav}{%
\setglossarystyle{treename}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}%
[\noindent\glstreenavigationfmt{\glsnavigation}]]%
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\glstrgetgrouptitle{##1}{\glstr@grptitle}%
\glstreePreHeader{##1}{\glstr@grptitle}%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glstr@grptitle}}%
\glstreegroupheaderskip\@afterheading}%
}
}%
{%
}
\ifdef{\@glsstyle@mcolalttree}

```

```

{%
  \renewglossarystyle{mcolalmtree}{%
    \setglossarystyle{almtree}%
    \renewenvironment{theglossary}%
    {%
      \glxtralmtreeInit
      \def\@gls@prevlevel{-1}%
      \begin{multicols}{\glsmcols}%
    }%
    {\par\end{multicols}}%
  }
}%
{%
}
\ifdef{\@glsstyle@mcolalmtreegroup}
{%
  \renewglossarystyle{mcolalmtreegroup}{%
    \setglossarystyle{mcolalmtree}%
    \renewcommand{\glsgroupheading}[1]{%
      \glxtrgetgrouptitle{##1}{\glxtr@grptitle}%
      \glstreePreHeader{##1}{\glxtr@grptitle}%
      \par
      \def\@gls@prevlevel{-1}%
      \hangindentOpt\relax
      \parindentOpt\relax
      \glstreegroupheaderfmt{\glxtr@grptitle}%
      \glstreegroupheaderskip
    }%
  }
}%
{%
}
\ifdef{\@glsstyle@mcolalmtreehypergroup}
{%
  \renewglossarystyle{mcolalmtreehypergroup}{%
    \setglossarystyle{mcolalmtree}%
    \renewcommand*\{glossaryheader}{%
      \par
      \def\@gls@prevlevel{-1}%
      \hangindentOpt\relax
      \parindentOpt\relax
      \glstreenavigationfmt{\glsnavigation}%
      \glstreegroupheaderskip
    }%
    \renewcommand*\{glsgroupheading}[1]{%
      \glxtrgetgrouptitle{##1}{\glxtr@grptitle}%
      \glstreePreHeader{##1}{\glxtr@grptitle}%
      \par
      \def\@gls@prevlevel{-1}%
      \hangindentOpt\relax
    }
  }
}

```



```

        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glxstr@grptitle}}%
        \glstreegroupheaderskip
    }%
}
}%
{%
}
\ifdef{\@glsstyle@mcolalttreespannav}
{%
    \renewglossarystyle{mcolalttreespannav}{%
        \setglossarystyle{alttree}%
        \renewenvironment{theglossary}%
        {%
            \glxstralttreeInit
            \def\@gls@prevlevel{-1}%
            \begin{multicols}{\glsmcols}%
                [\noindent\glstreenavigationfmt{\glsnavigation}}%
            }%
        }\par\end{multicols}}%
    \renewcommand*{\glsgroupheading}[1]{%
        \glxstrgetgrouptitle{##1}{\glxstr@grptitle}%
        \glstreePreHeader{##1}{\glxstr@grptitle}%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glxstr@grptitle}}%
        \glstreegroupheaderskip
    }%
}
}%
{%
}
\ifx\@glossary@default@style\relax
\else
    \setglossarystyle{\@glxstr@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{\glxstrbookindexcols}{2}
\newcommand*{\glxstrbookindexname}[1]{\glossentryname{#1}}
\newcommand*{\glxstrbookindexsubname}[1]{\glxstrbookindexname{#1}}

```

```

\providecommand*\glstrprelocation{\space}

\newcommand*\glstrbookindexprelocation[1]{%
  \glstrifhasfield{location}{#1}%
  {\glstrprelocation}%
  {\glstrprelocation}%
}
\newcommand*\glstrbookindexsubprelocation[1]{%
  \glstrbookindexprelocation{#1}%
}
\newcommand*\glstrbookindexlocation[2]{#2}
\newcommand*\glstrbookindexsublocation{\glstrbookindexlocation}
\newcommand*\glstrbookindexparentchildsep{\nopagebreak}
\newcommand*\glstrbookindexparentschildsep{\glstrbookindexparentchildsep}
\newcommand*\glstrbookindexbetween[2]{}
\newcommand*\glstrbookindexsubbetween[2]{}
\newcommand*\glstrbookindexsubsubbetween[2]{}
\newcommand*\glstrbookindexatendgroup[1]{}
\newcommand*\glstrbookindexsubatendgroup[1]{}
\newcommand*\glstrbookindexsubsubatendgroup[1]{}
\newcommand*\glstrbookindexgroupskip{\ifglsgroupskip\else\indexspace\fi}
\newcommand*\glstrbookindexformatheader[1]{%
  \par{\centering\glstreegroupheaderfmt{#1}\par}%
}
\ifdef\pdfbookmark
{%
  \newcommand*\glstrbookindexbookmark[2]{%
    \ifdefstring{\@glossarysec}{chapter}%
    {\pdfbookmark[1]{#1}{#2}}%
    {\pdfbookmark[2]{#1}{#2}}%
  }
}
{%
  \newcommand*\glstrbookindexbookmark[2]{}
}
\newcommand*\glstrbookindexbookmarkprefix{\currentglossary.}
\newcommand*\glstrbookindexcolspread{}
\newcommand*\glstrbookindexmulticolenv{\multicols}
\newglossarystyle{bookindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
  {%
    \ifnum\glstrbookindexcols>1\relax
    \ifdefempty\glstrbookindexcolspread
    {%
      \edef\glstr@beginbookindex{%
        \noexpand\begin{\glstrbookindexmulticolenv}
        {\glstrbookindexcols}%
      }%
    }%
  }%
}

```

```

{%
  \edef\glstr@beginbookindex{%
    \noexpand\begin{\glstrbookindexmulticolseenv}%
    {\glstrbookindexcols}{\glstrbookindexcolspread}%
  }%
}%
\else
  \def\glstr@beginbookindex{%
\fi
\glstr@beginbookindex
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\@glstr@bookindex@sep\glstrbookindexparentchildsep
\let\@glstr@bookindex@subsep\glstrbookindexparentsubchildsep
\let\@glstr@bookindex@between\@gobble
\let\@glstr@bookindex@subbetween\@gobble
\let\@glstr@bookindex@subsubbetween\@gobble
\let\@glstr@bookindex@atendgroup\relax
\let\@glstr@bookindex@subatendgroup\relax
\let\@glstr@bookindex@subsubatendgroup\relax
\let\@glstr@bookindex@groupskip\relax
}%
{%
  \@glstr@bookindex@subsubatendgroup
  \@glstr@bookindex@subatendgroup
  \@glstr@bookindex@atendgroup
  \ifnum\glstrbookindexcols>1\relax
    \edef\glstr@endbookindex{%
      \noexpand\end{\glstrbookindexmulticolseenv}%
    }%
  \else
    \def\glstr@endbookindex{%
\fi
\glstr@endbookindex
}%
\renewcommand*{\glossaryheader}{\raggedright}%
\renewcommand*{\glossentry}[2]{%
  \@glstr@bookindex@between{##1}%
  \let\@glstr@bookindex@sep\glstrbookindexparentchildsep
  \let\@glstr@bookindex@subsep\glstrbookindexparentsubchildsep
  \let\@glstr@bookindex@subbetween\@gobble
  \let\@glstr@bookindex@subsubbetween\@gobble
  \edef\@glstr@bookindex@between{%
    \noexpand\glstrbookindexbetween{##1}%
  }%
  \edef\@glstr@bookindex@atendgroup{%
    \noexpand\glstrbookindexatendgroup{##1}%
  }%
  \let\@glstr@bookindex@subatendgroup\relax
  \let\@glstr@bookindex@subsubatendgroup\relax

```

```

\glstreeitem
  \glstryitem{##1}%
  \glstarget{##1}{\glxtrbookindexname{##1}}%
  \glxtrbookindexprelocation{##1}%
  \glxtrbookindexlocation{##1}{##2}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifcase##1\relax
    \glstreeitem
  \or
    \@glxtr@bookindex@sep
    \@glxtr@bookindex@subbetween{##2}%
    \let\@glxtr@bookindex@sep\relax
    \let\@glxtr@bookindex@subsubbetween\@gobble
    \let\@glxtr@bookindex@subsep\glxtrbookindexparentschildsep
    \edef\@glxtr@bookindex@subbetween{%
      \noexpand\glxtrbookindexsubbetween{##2}%
    }%
    \edef\@glxtr@bookindex@atsubendgroup{%
      \noexpand\glxtrbookindexatsubendgroup{##1}%
    }%
    \glstreesubitem
    \glssubentryitem{##2}%
  \else
    \@glxtr@bookindex@subsep
    \@glxtr@bookindex@subsubbetween{##2}%
    \let\@glxtr@bookindex@subsep\relax
    \edef\@glxtr@bookindex@subsubbetween{%
      \noexpand\glxtrbookindexsubsubbetween{##2}%
    }%
    \edef\@glxtr@bookindex@atsubsubendgroup{%
      \noexpand\glxtrbookindexatsubsubendgroup{##1}%
    }%
    \glstreesubsubitem
  \fi
  \glstarget{##2}{\glxtrbookindexsubname{##2}}%
  \glxtrbookindexsubprelocation{##2}%
  \glxtrbookindexsublocation{##2}{##3}%
}%
\renewcommand*{\glsgroupskip}{}%
\renewcommand*{\glsgroupheading}[1]{%
  \@glxtr@bookindex@subsubatendgroup
  \@glxtr@bookindex@subatendgroup
  \@glxtr@bookindex@atendgroup
  \@glxtr@bookindex@groupskip
  \let\@glxtr@bookindex@groupskip\glxtrbookindexgroupskip
  \let\@glxtr@bookindex@between\@gobble
  \let\@glxtr@bookindex@atendgroup\relax
  \let\@glxtr@bookindex@subatendgroup\relax
  \let\@glxtr@bookindex@subsubatendgroup\relax

```

```

\glxstrgetgrouptitle{##1}{\glxstrcurrentgrptitle}%
\glxstrbookindexbookmark{\glxstrcurrentgrptitle}{\glxstrbookindexbookmarkprefix##1}%
\glxstrbookindexformatheader{\glxstrcurrentgrptitle}%
\nopagebreak\indexspace\nopagebreak\@afterheading
}%
}
\newcommand{\glxstrbookindexthepage}{%
\ifdef\currentglossary{\currentglossary.\arabic{page}}{\arabic{page}}%
}
\newcommand*{\glxstrbookindexmarkentry}[1]{%
\protected@write\@auxout
{\let\glxstrbookindexthepage\relax}%
{\string\glxstrsetbookindexmark{\glxstrbookindexthepage}{#1}}%
}
\newcommand*{\glxstrsetbookindexmark}[2]{%
\ifcsundef{glxstr@idxfirstmark@#1}%
{\csgdef{glxstr@idxfirstmark@#1}{#2}}%
{}%
\csgdef{glxstr@idxlastmark@#1}{#2}%
}
\newcommand*{\glxstrbookindexfirstmarkfmt}[1]{%
\glentryname{#1}%
}
\newcommand*{\glxstrbookindexfirstmark}{%
\letcs{glxstr@label}{glxstr@idxfirstmark@\glxstrbookindexthepage}%
\ifdef\glxstr@label
{\glxstrbookindexfirstmarkfmt{\glxstr@label}}%
{}%
}
\newcommand*{\glxstrbookindexlastmarkfmt}[1]{%
\glentryname{#1}%
}
\newcommand*{\glxstrbookindexlastmark}{%
\letcs{glxstr@label}{glxstr@idxlastmark@\glxstrbookindexthepage}%
\ifdef\glxstr@label
{\glxstrbookindexlastmarkfmt{\glxstr@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]{%
\glentryitem{#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}[1]{}
\newcommand{\glslongextraDescAlign}{>{\raggedright}p{\glsdescwidth}}
\newcommand{\glslongextraSymbolAlign}{c}
\newcommand{\glslongextraLocationAlign}{>{\raggedright}p{\glslistwidth}}
\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
        }%
      }%
    \fi
  }%
}

```

```

}%
{\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
}%
\ifglsgroupskip
\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
}%
\ifglsgroupskip
\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
}%
\ifglsgroupskip
\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]{%

```

```

        \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{\glosentry}[2]{%
    \glslongextraNameFmt{##1} &
    \glslongextraDescFmt{##1} &
    \glslongextraSymbolFmt{##1}&
    \glslongextraLocationFmt{##1}{##2}\tabularnewline
}%
\renewcommand{\subglosentry}[3]{%
    \glslongextraSubNameFmt{##1}{##2} &
    \glslongextraSubDescFmt{##1}{##2} &
    \glslongextraSubSymbolFmt{##1}{##2}&
    \glslongextraSubLocationFmt{##1}{##2}{##3}%
    \tabularnewline
}%
\ifglsgnognoskip
    \renewcommand*{\glsgrpskip}{}%
\else
    \renewcommand*{\glsgrpskip}{\glspenaltygrpskip}%
\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
}%
\ifglsgroupskip
  \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
}%
\ifglsgnogroupskip
  \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
    }
  \fi
}

```

```

}%
{%
  \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
}%
\ifglsgroupskip
  \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
}%
\ifglsgnogroupskip
\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}%
\beginngroup
\glstopicAssignSubIndent{##1}%
\glstopicSubItem{##1}{##2}{##3}%
\par

```

```

\endgroup
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glstopicGroupHeading}[1]{}
\newcommand*{\glstopicItem}[2]{%
\glspare\glstopicPreSkip\glspare\noindent
\glstopicMarker{#1}%
\glstopicTitleFont
{%
\glstryitem{#1}\glstarget{#1}{\glstopicTitle{#1}}%
}%
\ifglshasdesc{#1}%
{\glspare\nobreak\glstopicMidSkip\glspare\nobreak
\@afterheading\glstopicDesc{#1}\glspare\glstopicPostSkip}%
{\glspare\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{\glstryitem{#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

0.1 – 2015-11-22		\@Acrlong: added redefinition .	130
General: Initial experimental		\@Acrlongpl: added redefinition	131
release	2	\@Acrshort: added redefinition .	127
0.2 – 2015-11-30		\@Acrshortpl: added	
\Glsfmtshort: new	363	redefinition	128
\glsfmtshort: new	362	\@GLSdesc@: added redefinition .	115
\Glsfmtshortpl: new	363	\@GLSdescplural@: added	
\glsfmtshortpl: new	362	redefinition	116
short: switched inline full form		\@GLSfirst@: added redefinition	109
to short (long)	455	\@GLSfirstplural@: added	
0.3 – 2015-12-02		redefinition	113
\@ACRlong: added redefinition .	130	\@GLSname@: added redefinition .	114
\@ACRlongpl: added redefinition	131	\@GLSplural@: added	
\@ACRshort: added redefinition .	128	redefinition	111
\@ACRshortpl: added		\@GLSsymbol@: added	
redefinition	129	redefinition	118

\@GLSsymbolplural@: added		\@glsplural@: added	
redefinition	119	redefinition	110
\@GLStext@: added redefinition .	107	\@glssymbolplural@: added	
\@GLSuseri@: added redefinition	120	redefinition	118
\@GLSuserii@: added		\@glsextr@defaultnoglossarywarning:	
redefinition	121	new	195
\@GLSuseriii@: added		\@glsextr@field@linkdefs: new	106
redefinition	122	\@glsextr@insertdots: new . . .	312
\@GLSuseriv@: added		\@print@glossary: added	
redefinition	124	redefinition	191
\@GLSuserv@: added redefinition	125	\@glabbrvdefaultfont: renamed	
\@GLSuservi@: added		from \@abbrvdefaultfont . .	321
redefinition	126	\@glaccessdesc: new	237
\@GLSdesc@: added redefinition .	115	\@glaccessdescplural: new . .	238
\@GLSdescplural@: added		\@glaccessfirst: new	231
redefinition	116	\@glaccessfirstplural: new .	233
\@GLSfirst@: added redefinition	109	\@glaccesslong: new	242
\@GLSfirstplural@: added		\@glaccesslong: new	242
redefinition	112	\@glaccessname: new	227
\@GLSname@: added redefinition .	114	\@glaccessplural: new	230
\@GLsplural@: added		\@glaccessshort: new	240
redefinition	110	\@glaccessshort: new	239
\@GLsymbol@: added		\@glaccessshortpl: new	241
redefinition	117	\@glaccessshortpl: new	241
\@GLsymbolplural@: added		\@glaccesssymbol: new	234
redefinition	118	\@glaccesssymbolplural: new .	235
\@GLStext@: added redefinition .	108	\@glaccesstext: new	229
\@GLSuseri@: added redefinition	120	\@glentryfmt: added check for	
\@GLSuserii@: added		short	87
redefinition	121	\@glslongpltok: new	312
\@GLSuseriii@: added		\@glsshortpltok: new	312
redefinition	122	\@glsextr@newabbreviation: fixed	
\@GLSuseriv@: added		family name in \@setkeys . .	313
redefinition	123	\@glsextr@discardperiod: added	
\@GLSuserv@: added redefinition	124	check for plural	309
\@GLSuservi@: added		\@GLSxtrlongpl: new	333
redefinition	126	\@GLSxtrlongpl: new	333
\@acrlong: added redefinition .	129	\@glsextrlongpl: new	332
\@acrlongpl: added redefinition	130	\@glsextrNoGlossaryWarning: new	24
\@acrshort: added redefinition .	127	\@glsextrpostlinkAddDescOnFirstUse:	
\@acrshortpl: added		new	308
redefinition	128	\@glsextrpostlinkAddSymbolOnFirstUse:	
\@gl@field@link: added		new	308
optional argument	95	\@glsextrpostlinkendsentence:	
\@glSdescplural@: added		new	308
redefinition	116	\@GLSxtrshortpl: new	331
\@glSfirst@: added redefinition	108	\@GLSxtrshortpl: new	331
\@glSfirstplural@: added		\@glsextrshortpl: new	330
redefinition	112	long-short-desc: fixed name to	
		use \@glSlabeltok	444

short-long-desc: fixed name to use \glslabeltok	446	renamed from \Glsentryfmtshort	363
0.4 – 2015-12-03		\glsfmtshort: changed to use \glsxtrtitleshort	362
\@glsxtr@doabbreviationsdef: added redefinition of \acronymtype	19	renamed from \Glsentryfmtshort	362
\Glsfmtshort: changed to use \Glsxtrshort	363	\Glsfmtshortpl: changed to use \Glsxtrtitleshortpl	363
\glsfmtshort: changed to use \glsxtrshort	362	renamed from \Glsentryfmtshortpl	363
\Glsfmtshortpl: changed to use \glsxtrshortpl	363	\glsfmtshortpl: changed to use \glsxtrtitleshortpl	362
\glsfmtshortpl: changed to use \glsxtrshortpl	362	renamed from \Glsentryfmtshortpl	362
\glsxtrifemptyglossary: new .	38	\Glsfmttext: new	364
\glsxtrnewnumber: added extra argument	288	\glsfmttext: new	364
\glsxtrnewsymbol: added extra argument	288	\glshasattribute: new	284
\MakeAcronymsAbbreviations: set the default type to \acronymtype	172	\glshascategoryattribute: new	283
\newterm: fixed name argument	287	\glsxtremsuffix: new	499
0.5 – 2015-12-07		\GlsXtrEnableEntryCounting: new	157
\@cGLS: new	162	\glsxtrifcounttrigger: new .	160
\@cGLS@: new	162	\glsxtrscfont: new	463
\@cGLSpl: new	163	\glsxtrscsuffix: new	463
\@cGLSpl@: new	163	\glsxtrsmfont: new	481
\@glsxtr@setentrycountunsetattr: new	157	\glsxtrsmsuffix: new	481
\cGLS: new	162	long-noshort-em: new	511
\cGLSformat: new	162	long-noshort-em-desc: new . .	515
\cGLSpl: new	162	long-noshort-sm: new	489
\cGLSplformat: new	163	long-noshort-sm-desc: new . .	491
\GlossariesExtraWarningNoLine: new	18	long-short-em: new	499
\glsenableentrycount: new . .	158	long-short-em-desc: new	500
\glsfirstabbrvdefaultfont: new	320	long-short-sm: new	481
\glsfirstlongdefaultfont: new	322	long-short-sm-desc: new	483
\Glsfmtfirst: new	365	short-em: new	507
\glsfmtfirst: new	365	short-em-desc: new	508
\Glsfmtfirstpl: new	365	short-em-footnote: new	519
\glsfmtfirstpl: new	365	short-em-long: new	503
\Glsfmtplural: new	364	short-em-long-desc: new	504
\glsfmtplural: new	364	short-em-postfootnote: new .	522
\Glsfmtshort: changed to use \Glsxtrtitleshort	363	short-sc-footnote: new	475
		short-sc-postfootnote: new .	478
		short-sm: new	485
		short-sm-desc: new	487
		short-sm-footnote: new	493
		short-sm-long: new	483
		short-sm-long-desc: new	485
		short-sm-postfootnote: new .	496

0.5.1 – 2015-12-02		
\Glsaccessstext: new	229	
0.5.1 – 2015-12-07		
General: removed		
\ifglstruseuchead	350	
\@glstr@doaccsupp: new	24	
footnote: switch off regular		
attribute if set	448	
\Glsaccessdesc: new	237	
\Glsaccessdescplural: new	239	
\Glsaccessfirst: new	232	
\Glsaccessfirstplural: new	233	
\Glsaccessname: new	228	
\Glsaccessplural: new	231	
\Glsaccesssymbol: new	235	
\Glsaccesssymbolplural: new	236	
\Glsxtrheadfirst: now uses		
headuc attribute	356	
\glstrheadfirst: now uses		
headuc attribute	356	
\Glsxtrheadfirstplural: now		
uses headuc attribute	357	
\glstrheadfirstplural: now		
uses headuc attribute	357	
\Glsxtrheadplural: now uses		
headuc attribute	355	
\glstrheadplural: now uses		
headuc attribute	355	
\Glsxtrheadshort: now uses		
headuc attribute	351	
\glstrheadshort: now uses		
headuc attribute	350	
\Glsxtrheadshortpl: now uses		
headuc attribute	352	
\glstrheadshortpl: now uses		
headuc attribute	351	
\Glsxtrheadtext: now uses		
headuc attribute	354	
\glstrheadtext: now uses		
headuc attribute	353	
long-short: switch off regular		
attribute if set	443	
long-short-desc: switch off		
regular attribute if set	445	
long-short-sc-desc: switch off		
regular attribute if set	465	
postfootnote: switch off regular		
attribute if set	451	
short-em-footnote: switch off		
regular attribute if set	520	
short-em-footnote-desc: switch		
off regular attribute if set	522	
short-long: switch off regular		
attribute if set	445	
short-long-desc: switch off		
regular attribute if set	447	
short-postfootnote-desc:		
switch off regular attribute if		
set	454	
short-sc-footnote: switch off		
regular attribute if set	476	
short-sc-footnote-desc: switch		
off regular attribute if set	478	
short-sm-footnote: switch off		
regular attribute if set	493	
short-sm-footnote-desc: switch		
off regular attribute if set	495	
0.5.2 – 2015-12-08		
General: fixed typo in		
glossaries-accsupp and tidied		
up code to use just one		
\@ifpackageloaded	227	
removed \glstrabbrvfmt	334	
\@GLSdesc@: added accessibility		
support	115	
\@GLSdescplural@: added		
accessibility support	116	
\@GLSfirst@: added accessibility		
support	109	
\@GLSfirstplural@: added		
accessibility support	113	
\@GLSname@: added accessibility		
support	114	
\@GLSplural@: added accessibility		
support	111	
\@GLSsymbol@: added accessibility		
support	118	
\@GLSsymbolplural@: added		
accessibility support	119	
\@GLStext@: added accessibility		
support	107	
\@GLSdesc@: added accessibility		
support	115	
\@GLSdescplural@: added		
accessibility support	116	
\@GLSfirst@: added accessibility		
support	109	
\@GLSfirstplural@: added		
accessibility support	112	

\@Glsname@: add accessibility support	114	\GLSaccessshort: new	240, 272
\@Glsplural@: added accessibility support	110	\GLSaccessshortpl: new	242, 273
\@Glsymbol@: added accessibility support	117	\GLSaccesssymbol: new	235, 269
\@Glsymbolplural@: added accessibility support	118	\GLSaccesssymbolplural: new	236, 270
\@Glstext@: added accessibility support	108	\GLSaccessstext: new	230, 266
\@glsdesc@: added accessibility support	114	\glsentryfmt: moved	
\@glsdescplural@: added accessibility support	116	\glssetabbrvfmt from	
\@glsfirst@: added accessibility support	108	\glsxtrabbrvfmt to here	87
\@glsfirstplural@: added accessibility support	112	\GLSXtrEnableInitialTagging: new	303
\@glsname@: added accessibility support	113	\glsxtrfieldtitlecase: new	289
\@glsplural@: added accessibility support	110	\GLSXtrFormatLocationList: new	85
\@glsymbol@: added accessibility support	117	\glsxtrnewabbrevpresetkeyhook: new	317
\@glsymbolplural@: added accessibility support	118	\glsxtrtagfont: new	305
\@glstext@: added accessibility support	107	\KV@printgloss@nonumberlist: added	87
\@glsxtr@activate@initialtagging: new	305	\mfu@checkword@do: added	304
\@glsxtr@do@titlecaps@warn: new	305	\setabbreviationstyle: added	
\@glsxtr@tag: new	305	check for post-definition style switch	340
\glossaryentrynumbers: added	84	0.5.3 – 2015-12-09	
\Glossentrydesc: added	302	General: removed	
\Glossentryname: added	294	\GLSXtrNoGlsWarningNoAutoMakeMain	
\Glossentrysymbol: added	303		194
\GLSaccessdesc: new	238, 271	\@glsxtr@autoindex@at: new	300
\GLSaccessdescplural: new	239, 271	\@glsxtr@autoindex@encap: new	300
\GLSaccessfirst: new	232, 268	\@glsxtr@autoindex@esc: new	301
\GLSaccessfirstplural: new	234, 268	\@glsxtr@autoindex@level: new	301
\GLSaccesslong: new	243, 274	\@glsxtr@autoindex@setname: new	299
\GLSaccesslongpl: new	244	\@glsxtr@doabbreviationsdef: new	19
\Glsaccesslongpl: new	243	\glsdescwidth: added	84
\GLSaccessname: new	228, 265	\glspagelistwidth: added	84
\GLSaccessplural: new	231, 267	\glsxtrdoautoindexname: new	298
		\glsxtrpostnamehook: new	295
		\if@glsxtr@format@override: new	297
		\ProvidesGlossariesExtraLang: new	422
		\RequireGlossariesExtraLang: new	422
		0.5.4 – 2015-12-15	
		\@@newglossaryentry@defunitcounters: new	164

\@GLSxtr@p@acrlong@: new ...	148	\@glxstr@unitcountlist: new .	164
\@GLSxtr@p@acrlongpl@: new .	149	\@glxstrpl: new	81
\@GLSxtr@p@acrshort@: new ..	148	\@newglossaryentryposthook:	
\@GLSxtr@p@acrshortpl@: new .	148	added empty see value if not	
\@GLSxtr@p@long@: new	148	set and added ‘see’ to field	
\@GLSxtr@p@longpl@: new	148	key map	64
\@GLSxtr@p@plural@: new	146	\@sGlsXtrEnableOnTheFly: new	80
\@GLSxtr@p@short@: new	147	\cGlsformat: added	163
\@GLSxtr@p@shortpl@: new ...	147	\cglformat: added	163
\@GLSxtr@p@text@: new	146	\cGlsplformat: added	163
\@GlsXtrEnableOnTheFly: new .	80	\cglspformat: added	163
\@Glsxtr: new	81	\gl:disablehyper: added	145
\@Glsxtr@p@acrlong@: new ...	148	\gl:donohyperlink: added ...	145
\@Glsxtr@p@acrlongpl@: new .	149	\gl:enableentryunitcount:	
\@Glsxtr@p@acrshort@: new ..	148	new	166
\@Glsxtr@p@acrshortpl@: new .	148	\gl:hasattribute: added check	
\@Glsxtr@p@long@: new	148	for entry’s existence	284
\@Glsxtr@p@longpl@: new	148	\gl:sifattribute: added check	
\@Glsxtr@p@plural@: new	146	for entry’s existence	284
\@Glsxtr@p@short@: new	147	\gl:postlinkhook: added	
\@Glsxtr@p@shortpl@: new ...	147	existence check	306
\@Glsxtr@p@text@: new	146	\Glsxtr: new	81
\@Glsxtrpl: new	82	\glxtr: new	80
\@alt@glshyp@opt: new	141	\glxtrcat: new	80
\@glsh@alt@hyp@opt: new	140	\glxtrdohyperlink: added ...	143
\@glsh@alt@hyp@opt@char: new .	141	\glxtrdowrglossaryhook: new	140
\@glsh@alt@hyp@opt@keys: new .	141	\GlsXtrEnableEntryUnitCounting:	
\@glsh@increment@currunitcount:		new	169
new	164	\GlsXtrEnableOnTheFly: new .	79
\@glsh@local@increment@currunitcount:		\Glsxtrpl: new	82
new	165	\glxtrpl: new	81
\@glsh@setdefault@glshlink@opts:		\glxtrpostlocalreset: new .	156
new	137	\glxtrpostlocalunset: new .	156
\@glxtr: new	80	\glxtrpostreset: new	156
\@glxtr@addunitcounter: new	164	\glxtrpostunset: new	153
\@glxtr@currunitcount: new .	165	\glxtrprotectlinks: new ...	145
\@glxtr@ifunitcounter: new .	164	\GlsXtrSetAltModifier: new .	141
\@glxtr@p@acrlong@: new ...	148	\GlsXtrSetDefaultGlsOpts:	
\@glxtr@p@acrlongpl@: new .	149	new	138
\@glxtr@p@acrshort@: new ..	148	\glxtrstarflywarn: new	80
\@glxtr@p@acrshortpl@: new .	148	\GlsXtrWarning: new	82
\@glxtr@p@long@: new	147	\MakeAcronymsAbbreviations:	
\@glxtr@p@longpl@: new	148	now disables	
\@glxtr@p@plural@: new	146	\setacronymstyle	172
\@glxtr@p@short@: new	146	1.0 – 2016-01-24	
\@glxtr@p@shortpl@: new ...	147	\@glxtr@autoindexcrossrefs:	
\@glxtr@p@text@: new	146	new	17
\@glxtr@prevunitcount: new .	165	\@glxtr@idx@displaynumberlist:	
\@glxtr@setentryunitcountunsetattr:		new	185
new	169		

\@glxtr@idx@entrynumberlist:	1.03 – 2016-04-27
new	186
\@glxtr@noidx@displaynumberlist:	\@GLSfirstplural@: bug fix:
new	misspelt cs name 113
\@glxtr@noidx@entrynumberlist:	\@GLSplural@: fixed bug
new	\@GLSplural@ should be
\@glxtr@noidx@numberlistloop:	redefined not \@GLSplural 111
new	\@GLSfirstplural@: bug fix:
\@glxtr@reg@glosslist: new .	misspelt cs name 112
\makeglossaries: new	\@GLSplural@: fixed bug
1.01 – 2016-02-02	\@GLSplural@ should be
\glxtrdiscardperiod: added	redefined not \@GLSplural 110
check for first use	\@GLSplural@: fixed bug
short-desc: fixed typo in	\@GLSplural@ should be
\glxtrinlinefullformat	redefined not \@GLSplural 110
and added missing second	\glxtrtitlelongpl: bug fix:
argument	changed \glxtrlong to
1.02 – 2016-04-25	\glxtrlongpl 358
\@glxtr@current@style: new .	\glxtrtitleshortpl: bug fix:
\Glsfmtfull: new	changed \glxtrshort to
\Glsfmtfull: new	\glxtrshortpl 351
\Glsfmtfullpl: new	1.04 – 2015-04-30
\Glsfmtfullpl: new	short-em-footnote: renamed
\Glsfmtlong: new	from “footnote-em” 519
\Glsfmtlong: new	1.04 – 2016-05-02
\Glsfmtlongpl: new	\@@glxtrpostloctag: new . . . 86
\Glsfmtlongpl: new	\@GLSdesc@: set abbreviation and
\Glsxrheadfull: new	regular format 115
\Glsxrheadfull: new	\@GLSdescplural@: set
\Glsxrheadfullpl: new	abbreviation and regular
\Glsxrheadfullpl: new	format 116
\Glsxrheadlong: new	\@GLSfirst@: set abbreviation
\Glsxrheadlong: new	format 109
\Glsxrheadlongpl: new	\@GLSfirstplural@: set
\Glsxrheadlongpl: new	abbreviation and regular
\Glsxrtitlefull: new	format 113
\Glsxrtitlefull: new	\@GLSname@: set abbreviation and
\Glsxrtitlefullpl: new	regular format 114
\Glsxrtitlefullpl: new	\@GLSplural@: set abbreviation
\Glsxrtitlelong: new	and regular format 111
\Glsxrtitlelong: new	\@GLSsymbol@: set regular
\Glsxrtitlelongpl: new	format 118
\Glsxrtitlelongpl: new	\@GLSsymbolplural@: set regular
\ifglxtrinsertinside: new .	format 119
postfootnote: added redef of	\@GLStext@: set abbreviation and
\glxtrsetupfulldefs . . .	regular format 107
short-postfootnote-desc:	\@GLSuseri@: set regular format 120
added redef of	\@GLSuserii@: set regular
\glxtrsetupfulldefs . . .	format 121
stylemods: new	\@GLSuseriii@: set regular
	format 122

\@GLSuseriv@: set regular format	124	\@glsplural@: set abbreviation and regular format	110
\@GLSuseriv@: set regular format	125	\@glsymbol@: set regular format	117
\@GLSuservi@: set regular format	126	\@glsymbolplural@: set regular format	118
\@Glsdesc@: set abbreviation and regular format	115	\@glstext@: set abbreviation and regular format	107
\@Glsdescplural@: set abbreviation and regular format	116	\@glsxtr@deprecated@abbrstyle: new	344
\@Glsfirst@: set abbreviation and regular format	109	\@glsxtr@do@style: new	26
\@Glsfirstplural@: set abbreviation and regular format	112	\@glsxtr@doloctag: new	87
\@Glsname@: set abbreviation and regular format	114	\@glsxtr@idx@entrynumberlist: switched from \let to \newcommand	186
\@Glsplural@: set abbreviation and regular format	110	\@glsxtr@pagetag: new	86
\@Glsymbol@: set regular format	117	\@glsxtr@pagetag: new	86
\@Glsymbolplural@: set regular format	118	\@glsxtr@preloctag: new	86
\@Glstext@: set abbreviation and regular format	108	\@glsxtr@postloctag: new	86
\@Glsuseri@: set regular format	120	\@glsxtr@preloctag: new	86
\@Glsuserii@: set regular format	121	\glossentrydesc: added glossdescfont attribute check	290
\@Glsuseriii@: set regular format	122	\Glossentryname: added glossnamefont attribute check	294
\@Glsuseriv@: set regular format	123	\glossentryname: added glossnamefont attribute check	291
\@Glsuserv@: set regular format	124	moved post name hook inside condition	294
\@Glsuservi@: set regular format	126	\glsabbrvmfont: new	498
\@gls@preglossaryhook: added check for entry's existence .	305	\glsabbrvuserfont: new	526
\@glsdesc@: set abbreviation and regular format	114	\glsfirstabbrvmfont: new	499
\@glsdescplural@: set abbreviation and regular format	116	\glsfirstabbrvuserfont: new	526
\@glsfirst@: set abbreviation and regular format	108	\glsfirstlongemfont: new	499
\@glsfirstplural@: set abbreviation and regular format	112	\glsfirstlonguserfont: new	526
\@glsname@: set abbreviation and regular format	113	\glsifnotregularcategory: new	286
		\glslongdefaultfont: new	321
		\glslongemfont: new	499
		\glslongfont: new	321
		\glslonguserfont: new	526
		\glsxtrassignfieldfont: new	106
		\GlsXtrEnablePreLocationTag: new	85
		\glsxtrfirstscfont: new	463
		\glsxtrfirstsmfont: new	481
		\glsxtrlongshortdescsort: new	444

\glstrpostnamehook: added		
category check	295	
\glstrregularfont: new	87	
\glstruserfield: new	525	
\glstruserparen: new	525	
\glstrusersuffix: new	526	
\GlsXtrWarnDeprecatedAbbrStyle:		
new	344	
\letabbreviationstyle: new	344	
long-em-noshort-em: new	513	
long-em-noshort-em-desc: new	517	
long-em-short-em: new	501	
long-em-short-em-desc: new	502	
long-noshort: new	463	
long-noshort-desc: new	462	
long-noshort-em: renamed from		
“long-em”	511	
long-noshort-em-desc: renamed		
from “long-desc-em”	515	
long-noshort-sc: renamed from		
“long-sc”	472	
long-noshort-sc-desc: renamed		
from “long-desc-sc”	474	
long-noshort-sm: renamed from		
“long-sm”	489	
long-noshort-sm-desc: renamed		
from \long-desc-sm	491	
long-short-user: new	527	
long-short-user-desc: new	535	
\newabbreviationstyle: bug fix:		
corrected test for existence	342	
\renewabbreviationstyle: new	343	
short-em-long-em: new	505	
short-em-long-em-desc: new	506	
short-em-nolong: new	508	
short-em-nolong-desc: new	510	
short-em-postfootnote:		
renamed from		
“postfootnote-em”	522	
short-footnote: new	450	
short-long-user: new	536	
short-long-user-desc: new	537	
short-nolong: new	456	
short-nolong-desc: new	458	
short-postfootnote: new	453	
short-sc-footnote: renamed		
from “footnote-sc”	475	
short-sc-nolong: new	469	
short-sc-nolong-desc: new	471	
short-sc-postfootnote:		
renamed from		
“postfootnote-sc”	478	
short-sm-footnote: renamed		
from “footnote-sm”	493	
short-sm-nolong: new	487	
short-sm-nolong-desc: new	488	
short-sm-postfootnote:		
renamed from		
“postfootnote-sm”	496	
style: new	26	
1.05 – 2016-06-10		
\eglssetwidest: new	666	
\glsFindWidestAnyName: new	669	
\glsFindWidestAnyNameLocation:		
new	674	
\glsFindWidestAnyNameSymbol:		
new	672	
\glsFindWidestAnyNameSymbolLocation:		
new	673	
\glsFindWidestLevelTwo: new	670	
\glsFindWidestUsedAnyName:		
new	669	
\glsFindWidestUsedAnyNameLocation:		
new	674	
\glsFindWidestUsedAnyNameSymbol:		
new	671	
\glsFindWidestUsedAnyNameSymbolLocation:		
new	672	
\glsFindWidestUsedLevelTwo:		
new	669	
\glsFindWidestUsedTopLevelName:		
new	668	
\glsfirstlongfootnotefont:		
new	447	
\glsgetwidestname: new	668	
\glsgetwidestsubname: new	668	
\glslongfootnotefont: new	447	
\glstrAltTreeIndent: new	666	
\glstrAltTreeInit: new	666	
\glstrAltTreePar: new	666	
\glstrAltTreeSetHangIndent:		
new	675	
\glstrAltTreeSetSubHangIndent:		
new	676	
\glstrAltTreeSubSymbolDescLocation:		
new	666	
\glstrAltTreeSymbolDescLocation:		
new	665	

\glxstrComputeTreeIndent:		\glxstrabbrvfootnote: new ..	447
new	675	\glxstrchecknohyperfirst:	
\glxstrComputeTreeSubIndent:		new	108
new	675	\glxstrfieldtitlecasecs: new	289
\glxstrtreetopindent: new ..	666	\glxstrifinmark: new	347
short-em-long: fixed incorrect		\GLSxtrp: new	152
font used by long form ...	504	\Glsxtrp: new	151
\xglsssetwidest: new	667	\glxstrp: new	151
1.06 – 2016-06-18		\glxstrsetpopts: new	149
General: disabled docdef key at		long-short-desc: added missing	
the start of the document ..	37	text key	444
docdef option changed to		fixed misspelling of	
choice	16	\glsabbrvfont	445
\@glsdoifexistsorwarn: new ..	17	postfootnote: removed	
\@glxstr@docdefval: new	16	\footnote from first keys ..	451
\@glxstr@usesee: new	66	switched from	
\glxstr@usesee: new	66	\glsfirstlongfont to	
\glxstrusesee: new	65	\glsfirstlongfootnotefont	453
\glxstruseseeformat: new ...	66	\RestoreAcronyms: modified	
\if@glxstrdocdefrestricted:		\@gls@link@checkfirsthyper	
new	17	to set	
1.07 – 2016-08-15		\glxstrifwasfirstuse ...	173
\@@glxtrp: new	149	short-long-desc: added text	
\@GLSfirst@: added check for		key	447
nohyperfirst attribute	109	fixed misspelling of	
\@GLSfirstplural@: added check		\glsabbrvfont in plural key	447
for nohyperfirst attribute ..	113	1.08 – 2016-12-13	
\@GLSxtrp: new	150	\@@glxstr@record: new	6
\@GLSfirst@: added check for		\@GLS@: added \@glxstr@record	96
nohyperfirst attribute	109	\@GLSpl@: added	
\@GLSfirstplural@: added check		\@glxstr@record	96
for nohyperfirst attribute ..	112	\@Gls@: added \@glxstr@record	96
\@Glsxtrp: new	150	\@GLspl@: added	
\@gls@preglossaryhook: added		\@glxstr@record	96
\glossxtrsetpopts	306	\@gls@: added \@glxstr@record	95
\@glsfirst@: added check for		\@gls@@link: added	
nohyperfirst attribute	109	\@glxstr@record	97
\@glsfirstplural@: added check		\@gls@field@link: added	
for nohyperfirst attribute ..	112	\@glxstr@record	95
\@glxstrinmark: new	347	\@gls@saveentrycounter: new ..	36
\@glxstrnotinmark: new	347	\@glsdisp: added	
\@glxtrp: new	149	\@glxstr@record	96
\@glxtrp@opt: new	149	\@GLspl@: added	
footnote: changed first forms to		\@glxstr@record	95
use		\@glxstr@dorecord: new	8
\glsfirstlongfootnotefont	448	\@glxstr@err@undefaction: new	4
\glossxtrsetpopts: new	149	\@glxstr@record: new	5
\glsps: new	151	\@glxstr@warn@onexistsordo:	
\glspt: new	151	new	4
\glxstr@entry@p: new	150		

\@glxtr@warn@undefaction:		\glxtr@s@newignoredglossary:	
new	4	new	59
\@print@unsrt@glossary: new ..	205	\glxtr@shortcutsval: new ..	199
\glsadd: added		\glxtr@texencoding: new ...	198
\@glxtr@record	105	\glxtr@writefields: new ...	199
\glsdoifexists: now defines		\GlsXtrLoadResources: new ..	198
\glslabel	62	\glxtrpageref: new	54
\glxtr@@do@wrglossary: new ..	36	\glxtrresourcefile: changed	
\glxtr@addloclistfield: new ..	11	extension to .glstex	196
\glxtr@indexonly@saveentrycounter:		\newignoredglossary: added	
new	11	starred version	59
\glxtr@record: new	201		
\glxtr@resource: new	198	1.12 – 2017-02-03	
\glxtr@saveentrycounter: new ..	36	General: added target key to	
\glxtr@setup@record: new ..	11	printgloss family	182
\glxtrassignfieldfont: added		\@@glxtr@recordcounter: new ..	10
check for existence	106	\@gls@preglossaryhook: check	
\glxtrresourcefile: new ...	196	for definition	305
\printunsrtglossaries: new ..	205	\@glxtr@counterrecordhook:	
\printunsrtglossary: new ...	204	new	201
record: added record package		\@glxtr@display@loc: new ..	189
option	14	\@glxtr@docounterrecord:	
1.09 – 2016-12-16		new	201
\@glxtr@gettype: new	184	\@glxtr@longnewglossaryentry:	
\@glxtr@mixed@assign@sortkey:		new	58
new	184	\@glxtr@noop@recordcounter:	
\@printglossary: redefined to		new	10
save options	182	\@glxtr@op@recordcounter:	
\glxtr@makeglossaries: new ..	184	new	11
1.10 – 2016-12-17		\@glxtr@provide@storagekey:	
\@GLSpl@: fixed bug caused by		new	39
typo in command name	96	\@glxtr@s@longnewglossaryentry:	
1.11 – 2017-01-19		new	58
\@glxtr@do@redef@forxlsentries:		\@glxtr@entryfmt: new	42
new	4	\@glxtr@indexaliased: new ..	137
\@glxtr@noidx@do: new	213	\@glxtr@setaliasnoindex: new	137
\@glxtr@redef@forxlsentries:		\@newglossaryentryposthook:	
new	4	added check for alias key ...	74
\@glxtr@shortcutsval: new ..	23	\@no@glxtr@indexaliased: new	138
\@glxtr@unsrt@getgrouptitle:		\@printunsrtglossary: new ..	204
new	212	\@ptoglossarypreamble: new ..	55
\@print@noidx@glossary: added		\csGlsXtrLetField: new	50
redefinition	188	\eGlsXtrSetField: new	50
\glxtr@addloclistfield:		\gGlsXtrSetField: new	50
added group key	12	\glsnoidxdisplayloc: added	
added location key	11	redefinition	189
\glxtr@fields: new	198	\glssettoctitle: added patch ..	60
\glxtr@linkprefix: new	199	\glxtr@counterrecord: new ..	201
\glxtr@org@newignoredglossary:		\glxtr@langtag: new	198
new	59	\glxtr@newabbreviation: new	313

\glstr@org@newignoredglossary:	\GlsXtrSetField: new	49
Added check for existence	\glstrsetfieldifexists: new	49
\glstr@pluralsuffixes: new	\glstrunsrtdo: new	212
\glstr@provideignoredglossary:	\Glsxrusefield: new	49
new	\glstrusefield: new	48
\glstr@s@newignoredglossary:	long-postshort-user: new	528
Added check for existence	long-postshort-user-desc:	
\glstr@s@provideignoredglossary:	new	531
new	\longnewglossaryentry: added	
\glstrabbrvpluralsuffix:	starred version	58
new	postdot: new	18
\glstralias: new	\pretoglossarypreamble: new	55
\glstrcopytoglossary: new	\print@noop@unsrtglossaryunit:	
\glstrdeffield: new	new	211
\glstrdisplayendloc: new	\print@op@unsrtglossaryunit:	
\glstrdisplayendloohook:	new	211
new	\printunsrtglossary: added	
\glstrdisplaysingleloc: new	starred form	204
\glstrdisplaystartloc: new	\printunsrtglossaryhandler:	
\glstrdohyperlink: added	new	210
check for alias field	\printunsrtglossaryunit: new	11
\glstredeffield: new	\printunsrtglossaryunitsetup:	
\glstrentryfmt: new	new	211
\glstrfielddolistloop: new	\provideignoredglossary: new	60
\glstrfieldforlistloop: new	\s@glstr@provide@storagekey:	
\glstrfieldifinlist: new	new	40
\glstrfieldlistadd: new	\s@printunsrtglossary: new	204
\glstrfieldlistead: new	short-postlong-user: new	533
\glstrfieldlistgadd: new	short-postlong-user-desc:	
\glstrfieldlistxadd: new	new	534
\glstrfieldxifinlist: new	\xGlsXtrSetField: new	50
\glstrfmt: new	1.13 – 2017-02-07	
\GlsXtrFmtDefaultOptions: new	\@glsdisp: removed	
\GlsXtrFmtField: new	\@glstr@org@glsdisp	96
\glstrifkeydefined: new	\glstrsetaliasnoindex:	
\glstrindexaliased: new	switched to	
\GlsXtrLetField: new	\providecommand	137
\GlsXtrLetFieldToField: new	1.14 – 2017-04-18	
\GlsXtrLoadResources: removed	General: added \glsadd option	
restriction on only one per	theHvalue	105
document	added \glsadd option	
\glstrlocrangefmt: new	thevalue	105
\glstrpostlongdescription:	\@gls@link: added redefinition	102
new	\@gls@noidx@getgrouptitle:	
\glstrprovidestoragekey: new	new	186
\GlsXtrRecordCounter: new	\@gls@removespaces: new	190
\glstrresourcecount: new	\@glstr@do@automake@err:	
\glstrresourcefile: added	new	200
catcode change for @	\@glstr@org@gloautosee: new	34
\glstrsetaliasnoindex: new		

\@glstr@record: added third arg	5	short-postfootnote-desc: fixed spelling of \@glabbrvfont .	453
\@glstr@recordsee: new	11	short-postlong-user: fixed spelling of \@glabbrvfont .	533
\gl:disablehyper: added redefinition	144	short-postlong-user-desc: fixed spelling of \@glabbrvfont	535
\gl:enableentrycount: fixed assignment of \@cGls@ . . .	159	1.16 – 2017-06-15	
\gl:enableentryunitcount: fixed assignment of \@cGls@	167	\@glo@autosee: added redefinition	35
\gl:navigation: new	188	\@gl:noidx@getgrouptitle: fixed bug	186
\glstr@org@getgrouptitle: new	187	\@glstr@addunusedxrefs: added check for seealso field .	75
\glstr@recordsee: new	5	\@glstr@checkgroup: use \csuse instead of \csname .	212
\glstr@writefields: added check for automake	200	\@glstr@dorecordnodefer: new	9
\glstrdisplayendloc: added check for empty format . . .	190	\@glstr@record@only@setup: added check for \@gl:setupsort@none	13
\glstrgetgrouptitle: new . .	187	\@glstr@unsrt@gloss@init: corrected misspelt command	206
\glstrinitwrgloss: new	97	\@printunsrt@glossary@handler: new	210
\glstrlocationhyperlink: new	191	autoseeindex: new	17
\glstrsetgrouptitle: new . .	188	\gl:checkseeallowed: added redefinition	35
\glstrsupphypertextnumber: new .	191	\glstr@writefields: added \providecommand lines . . .	199
\ifglstrwrglossbefore: new .	97	\glstrautoindex: new	299
1.15 – 2017-05-10		\glstrautoindexassignsort: new	299
\@glstr@dorecord: corrected premature expansion of \@gl:locref	8	\glstrautoindexentry: new .	299
footnote: fixed spelling of \@glabbrvfont	448	\glstrindexseealso: new . . .	72
long-em-short-em: fixed spelling of \@glabbrvfont	501	\glstrseealsolabels: new . .	74
long-postshort-user: fixed spelling of \@glabbrvfont .	528	\glstrseelist: new	70
long-postshort-user-desc: fixed spelling of \@glabbrvfont	532	\glstruseseealso: new	69
long-short: fixed spelling of \@glabbrvfont	443	\glstruseseealsoformat: new	69
long-short-user: fixed spelling of \@glabbrvfont	527	\seealso: new	72
postfootnote: fixed spelling of \@glabbrvfont	451	1.17 – 2017-08-09	
short-em-long-em: fixed spelling of \@glabbrvfont	505	General: removed some inconsistencies in the abbreviation styles	442
short-long: fixed spelling of \@glabbrvfont	445	\@glstr@mark@wordseps: new .	313
short-long-user: fixed spelling of \@glabbrvfont	536	\@glstr@markwordseps: new .	312
		\@glstr@noidx@displaynumberlist: replace hard-coded ?? with \glstrundeftag	185

\@glxstr@noidx@entrynumberlist: replace hard-coded ?? with \glxstrundeftag 186	\glxstrshorthyphennoinsert: new 557
\@glxstr@noidx@numberlistloop: replace hard-coded ?? with \glxstrundeftag 186	\glxstrshortlongdescname: new 446
\@glxstrifhyphenstart: new .. 538	\glxstrshortlongdescsort: new 446
\glsabbrvhyphenfont: new ... 540	\Glsxtrsubsequentfmt: new .. 338
\glsabbrvonlyfont: new 577	\glxtrsubsequentfmt: new .. 337
\glsabbrvscfont: new 463	\Glsxtrsubsequentplfmt: new . 338
\glsabbrvsmfont: new 481	\glxtrsubsequentplfmt: new . 337
\glsabbrvuserfont: initialised to default font 526	\glxtrword: new 312
\glsfirstabbrvhyphenfont: new 540	\glxtrwordsep: new 312
\glsfirstabbrvonlyfont: new . 577	long-em-noshort-em-desc-noreg: new 519
\glsfirstabbrvscfont: new .. 463	long-em-noshort-em-noreg: new 515
\glsfirstabbrvsmfont: new .. 481	long-hyphen-noshort-desc-noreg: new 547
\glsfirstlonghyphenfont: new 540	long-hyphen-noshort-noreg: new 554
\glsfirstlongonlyfont: new . 577	long-hyphen-postshort-hyphen: new 557
\glslonghyphenfont: new 540	long-hyphen-postshort-hyphen-desc: new 562
\glslongonlyfont: new 577	long-hyphen-short-hyphen: new 540
\glslonguserfont: initialised to default font 526	long-hyphen-short-hyphen-desc: new 545
\glxstr@newabbreviation: added \glxstrorgshort and \glxstrorglong 313	long-noshort-desc-noreg: new 462
\GlsXtrDefineAcShortcuts: new 21	long-noshort-noreg: new 463
\glxstrgenabbrvfmt: added check for \ifglxstrinsertinside .. 334	long-only-short-only: new .. 577
\glxstrhyphensuffix: new ... 540	long-only-short-only-desc: new 579
\glxstrifhyphenstart: new .. 538	long-short-user-desc: corrected first forms 535
\glxstrlonghyphen: new 554	short-hyphen-long-hyphen: new 564
\glxstrlonghyphennoshort: new 546	short-hyphen-long-hyphen-desc: new 569
\glxstrlonghyphenshort: new . 538	short-hyphen-postlong-hyphen: new 571
\glxstrlongshortdescname: new 444	short-hyphen-postlong-hyphen-desc: new 576
\glxstronlydescname: new ... 579	short-long-user-desc: corrected first forms 537
\glxstronlydescsort: new ... 579	short-nolong-desc-noreg: new 458
\glxstronlysuffix: new 577	short-nolong-noreg: new 456
\glxstrparen: new 317	
\glxstrposthyphenlong: new . 570	
\glxstrposthyphenshort: new . 555	
\glxstrposthyphensubsequent: new 556	
\glxstrshortdescname: new .. 456	
\glxstrshorthyphen: new 569	
\glxstrshorthyphenlong: new . 563	

1.18 – 2017-08-10

stylemods: changed default value
 to “default” 25

1.19 – 2017-09-09		
General: added <code>\glslink</code> option		
<code>theHvalue</code>	99	
added <code>\glslink</code> option		
<code>thevalue</code>	99	
<code>\@glsxtr@defaultnumberformat:</code>		
new	5	
<code>\@glsxtr@dorecord:</code> Use		
<code>\@glsrecordloc</code> instead of		
<code>\@glsloc</code>	8	
<code>\@glsxtr@dorecordnodefer:</code> Use		
<code>\theGlsentrycounter</code> for the		
location rather than		
<code>\@glsloc</code>	9	
<code>\@glsxtr@record@setting:</code> new	12	
<code>\@glsxtr@record@setting@alsoindex:</code>		
new	12	
<code>\@glsxtrifhasfield:</code> new	47	
<code>\glsxtr@writefields:</code> removed		
double-quotes around		
<code>\jobname</code>	200	
<code>\glsxtrdoautoindexname:</code>		
changed format test	298	
<code>\glsxtrhyperlink:</code> new	144	
<code>\glsxtrifhasfield:</code> new	46	
<code>\GlsXtrSetDefaultNumberFormat:</code>		
new	5	
<code>\s@glsxtrifhasfield:</code> new ...	47	
1.20 – 2017-09-11		
<code>\@glsxtrhypernameprefix:</code> new	182	
<code>\glsdohypertarget:</code> added		
redefinition	184	
<code>\printunsrtglossaryunitsetup:</code>		
switched from redefining		
<code>\glslinkprefix</code> to		
<code>\@glsxtrhypernameprefix</code>	211	
1.21 – 2017-11-03		
General: adjusted <code>mcalttree</code> ..	685	
modified index to remove hard		
coded <code>\space</code>	657	
modified list to remove hard		
coded <code>\space</code>	644	
moved conditional outside of		
<code>\glsgrpskip</code>		
.....	648, 649, 651, 653, 655	
new	689	
redefined <code>altlistgroup</code> to		
discourage breaks after group		
headings	647	
redefined <code>altlisthypergroup</code>		
to discourage breaks after		
group headings	647	
redefined <code>alttreegroup</code> to		
discourage breaks after group		
headings	677	
redefined <code>alttreehypergroup</code>		
to discourage breaks after		
group headings	678	
redefined <code>indexgroup</code> to		
discourage breaks after group		
headings	658	
redefined <code>indexhypergroup</code> to		
discourage breaks after group		
headings	659	
redefined <code>listgroup</code> to		
discourage breaks after group		
headings	646	
redefined <code>listhypergroup</code> to		
discourage breaks after group		
headings	646	
redefined <code>mcalttreegroup</code> to		
discourage breaks after group		
headings	685	
redefined		
<code>mcalttreehypergroup</code> to		
discourage breaks after group		
headings	686	
redefined <code>mcalttreespannav</code>		
to discourage breaks after		
group headings	688	
redefined <code>mcindexgroup</code> to		
discourage breaks after group		
headings	680	
redefined		
<code>mcindexhypergroup</code> to		
discourage breaks after group		
headings	680	
redefined <code>mcindexspannav</code> to		
discourage breaks after group		
headings	681	
redefined <code>mcaltreegroup</code> to		
discourage breaks after group		
headings	682	
redefined <code>mcalttreehypergroup</code>		
to discourage breaks after		
group headings	682	
redefined		
<code>mcaltreenonamegroup</code> to		

discourage breaks after group headings	683	\@glxstrsetaliasnoindex: changed to use \glxstrifhasfield instead of \ifglshasfield	137
redefined mcoltreenonamehypergroup to discourage breaks after group headings	684	\@glxstrwrglossmark: new . . .	27
redefined mcoltreenonamespannav to discourage breaks after group headings	684	\@rGLS: new	224
redefined mcoltreespannav to discourage breaks after group headings	683	\@rGLS@: new	224
redefined treegroup to discourage breaks after group headings	662	\@rGLSpl: new	224
redefined treehypergroup to discourage breaks after group headings	662	\@rGLSpl@: new	224
redefined treenonamegroup to discourage breaks after group headings	664	\@rGls: new	223
redefined treenonamehypergroup to discourage breaks after group headings	665	\@rGls@: new	223
\@glxstr@record: added check for default options	7	\@rGlspl: new	223
\@glxstrwrglossmark: new . .	27	\@rgls: new	222
\@glslink: changed \let to \def	145	\@rgls@: new	222
\@glxstr@checkgroup: new . .	212	\@rglspl: new	223
\@glxstr@defpostpunc: new . .	18	\@rglspl@: new	223
\@glxstr@do@record@wrglossary: new	5	\@rgls: new	222
\@glxstr@dosee@alsoindex@glossary: new	34	\@rgls@: new	222
\@glxstr@doseeglossary: new .	34	\@rglspl: new	223
\@glxstr@noidx@do: removed code dealing with the group	214	\@rglspl@: new	223
\@glxstr@printunsrtglossaryskipentry: new	210	all: new	642
\@glxstr@record@setting@off: new	13	debug: new	28
\@glxstr@record@setting@only: new	12	\@glssetwidest: new	666
\@glxstr@rglstrigger@record: new	221	\@glsdisablehyper: added check for existence	144
\@glxstrglossentry: new . . .	202	changed to use \def rather than \let	145
\@glxstrnewgls: new	218	\@glsenablehyper: changed to use \def rather than \let	145
		\Glsfmtname: new	363
		\glsfmtname: new	363
		\glshex: new	584
		\glslistchildpostlocation: new	644
		\glslistchildprelocation: new	644
		\glslistprelocation: new . . .	644
		\glsnavhyperlink: patched . . .	142
		\glsseeitemformat: new	66
		\glsshowtarget: new	33
		\glstreechildprelocation: new	657
		\glstreeprelocation: new . . .	657
		\glstriggerrecordformat: new	222
		\glsuseabbrvfont: new	334
		\glsuselongfont: new	334
		\glxstr@do@alsoindex@wrglossary: new	6
		\glxstr@org@@do@wrglossary: new	36
		\glxstr@org@dohyperlink: new	142
		\glxstr@setbookindexmark: new	697

\glxtrbookindexatendgroup:		\glxtrifrecordtrigger: new .	221
new	691	\glxtrindexseealso: added	
\glxtrbookindexbetween: new	690	check that the entry exists ..	72
\glxtrbookindexbookmark:		\glxtrinithyperoutside: new	100
new	692	\GlsXtrLocationRecordCount:	
\glxtrbookindexcols: new ..	690	new	220
\glxtrbookindexcolspread:		\glxtrnewgls: new	215, 219
new	692	\glxtrnewGLSlike: new	219
\glxtrbookindexfirstmark:		\glxtrnewglslike: new	219
new	697	\glxtrnewrgls: new	219
\glxtrbookindexfirstmarkfmt:		\glxtrnewrglslike: new	219
new	697	\glxtrprelocation: new	643, 690
\glxtrbookindexformatheader:		\GlsXtrRecordCount: new	220
new	691	\glxtrrecordtriggervalue:	
\glxtrbookindexgroupskip:		new	220
new	691	\glxtrresourcefile: now	
\glxtrbookindexlastmark:		disables record key	196
new	697	\glxtrresourceinit: new ...	197
\glxtrbookindexlastmarkfmt:		\GlsXtrSetRecordCountAttribute:	
new	697	new	221
\glxtrbookindexmarkentry:		\GlsXtrtitlename: new	353
new	696	\glxtrtitlename: new	353
\glxtrbookindexname: new ..	690	\glxtrtitleorpdforheading:	
\glxtrbookindexparentchildsep:		new	347
new	690	\GlsXtrTotalRecordCount: new	220
\glxtrbookindexparentschildsep:		\glxtrwrglossmark: new	27
new	690	\ifglxtr@hyperoutside: new .	99
\glxtrbookindexprelocation:		nolong-short: new	458
new	690	nolong-short-em: new	510
\glxtrbookindexsubatendgroup:		nolong-short-noreg: new	459
new	691	nolong-short-sc: new	471
\glxtrbookindexsubbetween:		nolong-short-sm: new	489
new	691	nopostdot: new	18
\glxtrbookindexsubname: new	690	postpunc: new	18
\glxtrbookindexsubprelocation:		\printunsrtglossaryentryprocesshook:	
new	690	new	210
\glxtrbookindexsubsubatendgroup:		\printunsrtglossarypredoglossary:	
new	691	new	210
\glxtrbookindexsubsubbetween:		\printunsrtglossaryskipentry:	
new	691	new	210
\glxtrbookindexthepage: new	696	\rGLS: new	224
\glxtrdetoklocation: new ..	220	\rGls: new	223
\glxtrenablerecordcount:		\rgls: new	222
new	220	\rGLSformat: new	225
\glxtrglossentry: new	201	\rGlsformat: new	225
\glxtrgroupfield: new	212	\rglsformat: new	225
\GlsXtrheadname: new	353	\rGLSpl: new	224
\glxtrheadname: new	352	\rGlspl: new	223
\GlsXtrIfFieldEqStr: new ...	50	\rglspl: new	222
\glxtriflabelinlist: new ..	211		

\rGLSplformat: new	225	\GlsXtrDefineAcShortcuts:	
\rGLsplformat: new	225	changed \newabbr definition	
\rglsplformat: new	225	to use \providecommand	22
\s@glxstrifhasfield: switched		\glxstrfmtdisplay: new	42
from \ifdef to \ifundef	47	\glxstrifcustomdiscardperiod:	
short-sc: corrected first letter		new	307
uppercasing	469	\GlsXtrIfFieldUndef: new	48
short-sm: corrected first letter		\glxstrrestorepostpunc: new	182
uppercasing	486	\s@glxstrfmt: new	41
shortcuts: ac	24	\s@glxstrfmt: new	41
1.22 – 2017-11-08		\xglsupdatewidest: new	667
\@glxstr@nopostpunc: new	181	1.24 – 2017-11-14	
\@glxstr@orgprintglossary:		\gl sadd: added \@gl s@setsort	106
changed explicit \let for		\glxstrforcsvfield: new	44
\nopostdesc to		\glxstrlocalsetgrouptitle:	
\glxstractivatenopost	180	new	188
\@glxstrglossentryother: new	203	1.25 – 2017-11-14	
\glossentrynameother: new	296	\glxstrbookindexmulticol env:	
\gl sseeitemformat: switched		new	692
check from regular to short	66	1.25 – 2017-11-24	
\glxstr@setaccessdisplay:		\gl sextrapostnamehook: new	295
new	296	\glxstrfootnotename: new	448
\glxstr@writefields: provide		\glxstrlongnoshortdescname:	
\glxstr@record in aux file	199	new	460
\glxstractivatenopost: new	181	\glxstrlongnoshortname: new	462
\glxstrbookindexprelocation:		\glxstrlongshortname: new	423
removed check for no post		\glxstrlongshortuserdescname:	
dot	690	new	531
\glxstrglossentryother: new	203	\glxstronlyname: new	577
\glxstrnopostpunc: new	181	\glxstrpostlinkAddDescOnFirstUse:	
1.23 – 2017-11-12		changed to use	
\@@glxstrfmt: added check for		\glxstrparen	308
indexing	41	\glxstrpostlinkAddSymbolOnFirstUse:	
added grouping	41	changed to use	
new	41	\glxstrparen	308
\@glxstr@nopostpunc@postdesc:		\glxstrshortlongname: new	445
new	181	\glxstrshortlonguserdescname:	
\@glxstr@restore@postpunc:		new	534
new	181	\glxstrshortnolongname: new	454
\@glxstrentryfmt: fixed missing		1.26 – 2018-01-05	
label argument	42	\@glxstr@do@inc@linkcount:	
\@glxstrfmt: new	41	new	226
\eglsupdatewidest: new	667	\gl slinkpresetkeys: new	101
\ggl supdatewidest: new	667	\glxstr@inc@linkcount: new	100
\gl supdatewidest: new	667	\GlsXtrEnableLinkCounting:	
\GlsXtrDefineAbbreviationShortcuts:		new	227
changed \newabbr definition		\GlsXtrIfLinkCounterDef: new	227
to use \providecommand	21	\glxstrinclinkcounter: new	226
		\GlsXtrLinkCounterName: new	227
		\GlsXtrLinkCounterValue: new	226

\GlsXtrTheLinkCounter: new .	226	\glxtrLatinAA: new	625
1.27 – 2018-02-26		\glxtrLatinAELigature: new .	625
General: added		\glxtrLatinE: new	623
glossaries-extra-bib2gls.sty .	582	\glxtrLatinEszettSs: new ..	624
\@glxtrdialecthook: new . . .	37	\glxtrLatinEszettSz: new ..	625
\Alpha: new	607	\glxtrLatinEth: new	625
\Beta: new	607	\glxtrLatinH: new	623
\Chi: new	608	\glxtrLatinI: new	623
\Digamma: new	608	\glxtrLatinInsularG: new ..	625
\Epsilon: new	607	\glxtrLatinK: new	623
\Eta: new	607	\glxtrLatinL: new	624
\glxtr@loaddialect: new . . .	422	\glxtrLatinLslash: new	626
\glxtrBasicDigitrules: new .	639	\glxtrLatinM: new	624
\glxtrcombiningdiacriticIIIrules:		\glxtrLatinN: new	624
new	612	\glxtrLatinO: new	624
\glxtrcombiningdiacriticIIrules:		\glxtrLatinOELigature: new .	625
new	612	\glxtrLatinOslash: new	625
\glxtrcombiningdiacriticIrules:		\glxtrLatinP: new	624
new	611	\glxtrLatinS: new	624
\glxtrcombiningdiacriticIVrules:		\glxtrLatinSchwa: new	624
new	613	\glxtrLatinT: new	624
\glxtrcombiningdiacriticrules:		\glxtrLatinThorn: new	625
new	611	\glxtrLatinWynn: new	625
\glxtrcontrolrules: new . . .	610	\glxtrLatinX: new	624
\glxtrcurrencyrules: new ..	615	\glxtrMathGreekIIrules: new	632
\glxtrdigitrules: new	639	\glxtrMathGreekIrules: new .	631
\glxtrfractionrules: new ..	640	\glxtrMathItalicAlpha: new .	636
\glxtrGeneralLatinIIrules:		\glxtrMathItalicBeta: new .	636
new	617	\glxtrMathItalicChi: new ..	639
\glxtrGeneralLatinIrules:		\glxtrMathItalicDelta: new .	636
new	617	\glxtrMathItalicEpsilon:	
\glxtrGeneralLatinIrules:		new	636
new	616	\glxtrMathItalicEta: new ..	637
\glxtrGeneralLatinIVrules:		\glxtrMathItalicGamma: new .	636
new	618	\glxtrMathItalicGreekIIrules:	
\glxtrGeneralLatinVIIrules:		new	627
new	621	\glxtrMathItalicGreekIrules:	
\glxtrGeneralLatinVIIrules:		new	627
new	620	\glxtrMathItalicIota: new .	637
\glxtrGeneralLatinVrules:		\glxtrMathItalicKappa: new .	637
new	619	\glxtrMathItalicLambda: new	637
\glxtrGeneralLatinVrules:		\glxtrMathItalicLowerGreekIIrules:	
new	619	new	630
\glxtrgeneralpuncIIrules:		\glxtrMathItalicLowerGreekIrules:	
new	616	new	629
\glxtrgeneralpuncIrules:		\glxtrMathItalicMu: new . . .	637
new	614	\glxtrMathItalicNabla: new .	639
\glxtrgeneralpuncrules: new	614	\glxtrMathItalicNu: new . . .	637
\glxtrhyphenrules: new	613	\glxtrMathItalicOmega: new .	639
\glxtrLatinA: new	623		

\@gls@removespaces: added	added \glsaddpresetkeys ..	105
expansion	\glsaddpostsetkeys: new	105
\@glsxtr@wrglossary@locationhyperlink:	\glsaddpresetkeys: new	105
new	\glsuserdescription: new ...	526
\glsxtr@inc@wrglossaryctr:	\glsxtrabbreviationfont: new	87
new	\GlsXtrDualBackLink: new ...	585
\glsxtr@wrglossarylocation:	\GlsXtrDualField: new	585
new	\GlsXtrExpandedFmt: new	101
\GlsXtrBibTeXEntryAliases:	\GLSxtrlong: added	
new	\@glsxtr@record	330
\glsxtrfieldforlistloop:	\Glsxtrlong: added	
corrected argument order in	\@glsxtr@record	329
\forlistcsloop	\glsxtrlong: added	
\GlsXtrIndexCounterLink: new	\@glsxtr@record	328
\GlsXtrInternalLocationHyperlink:	\GLSxtrlongpl: added	
new	\@glsxtr@record	334
\GlsXtrProvideBibTeXFields:	\Glsxtrlongpl: added	
new	\@glsxtr@record	333
indexcounter: new	\glsxtrlongpl: added	
\setentrycounter: new	\@glsxtr@record	332
1.30 – 2018-04-25	\GLSxtrshort: added	
\@glsxtr@dorecord: don't	\@glsxtr@record	327
suppress expansion of	\Glsxtrshort: added	
\@glsrecordlocref	\@glsxtr@record	327
\@glsxtr@record: added check	\glsxtrshort: added	
for post-key hook	\@glsxtr@record	326
added check for pre-key hook .	\GLSxtrshortpl: added	
\@GLSxtr@fullpl: added	\@glsxtr@record	332
\@glsxtr@record	\Glsxtrshortpl: added	
\@GlsXtrStopUnsetBuffering:	\@glsxtr@record	331
new	\glsxtrshortpl: added	
\@Glsxtr@fullpl: added	\@glsxtr@record	330
\@glsxtr@record	\GlsXtrStartUnsetBuffering:	
\@glsxtr@full: added	new	153
\@glsxtr@record	\GlsXtrStopUnsetBuffering:	
\@glsxtr@fullpl: added	new	155
\@glsxtr@record	indexcounter: added check for	
\@glsxtr@glossadd@postkeys:	wrglossary counter	27
new	\s@GlsXtrStopUnsetBuffering:	
\@glsxtr@glossadd@prekeys:	new	155
new	1.31 – 2018-05-09	
\@glsxtr@glslink@postkeys:	General: added prefix key for	
new	glslink	100
\@glsxtr@glslink@prekeys: new	added prefix key for	
\@glsxtr@local@textformat:	printgloss	182
new	changed \let to \def	182
\@glsxtr@unset: new	\@GlsXtrStartUnsetBuffering:	
\@glsxtrbuffer@unset: new ..	new	153
\glsadd: added	\@gls@ifaccessattribute@set:	
\glsaddpostsetkeys	new	260

\@gls@initaccesskeys:		\GlsXtrStandaloneGlossaryType:	
new	259, 279	new	202
\@gls@setup@default@short@access:		\GlsXtrStandaloneSubEntryItem:	
new	261	new	203
\@glsxtr@record@noglossarywarning:		\s@GlsXtrStartUnsetBuffering:	
new	196	new	154
\@glsxtrbuffer@nodup@unset:		1.32 – 2018-05-24	
new	154	\GlsXtrForeignText: new	52
\glsaddeach: new	106	\GlsXtrForeignTextField: new ..	54
\glsapturedgroup: new	584	\GlsXtrUnknownDialectWarning:	
\glsdefpostdesc: new	306	new	54
\glsdefpostlink: new	307	1.33 – 2018-07-26	
\glsdefpostname: new	295	\ifglsused: added redefinition .	56
\glsdohypertarget: bug fix:		1.34 – 2018-07-29	
ensure that new version is		docdef: atom	16
picked up	184	\gls@begindocdefs: atom	77
\glslistdesc: new	644	\GlsXtrIfUnusedOrUndefined:	
\glslocalreseteach: new	156	new	38
\glslocalunseteach: new	156	\glsxtrNoGlossaryWarning:	
\glsrechilddesc: new	661	added package warning	24
\glsrechildsymbol: new	661	\ifglsxtrdocdefrestricted:	
\glsredefaultnamefmt: new ..	656	changed to allow for atom as	
\glsredesc: new	660	well	17
\glsregrgroupheaderfmt: added		1.35 – 2018-08-13	
redefinition	656	\@gls@@link: initialise post-link	
\glsreenamefmt: added		hook commands	97
redefinition	656	1.36 – 2018-08-18	
\glsreenavigationfmt: added		\glsxtrautoindexesc: new ...	299
redefinition	656	\glsxtrdisplaysupplc: new ..	587
\glsreenonamechilddesc: new	663	\glsxtrmultisupplcation:	
\glsreenonamedesc: new	663	new	586
\glsreenonamesymbol: new ..	663	1.37 – 2018-11-30	
\glsreesymbol: new	660	General: new	697
\glsxtr@newabbreviation: added		\@@glsxtr@dorecord: nameref ..	9
\ExtraCustomAbbreviationFields		\@@glsxtr@record: added check	
.....	313	for auto-add	7
\GlsXtrForUnsetBufferedList:		\@dGLS: new	598
new	156	\@dGLSpl: new	598
\GlsXtrIfFieldCmpNum: new ..	48	\@dGLs: new	598
\GlsXtrIfFieldEqNum: new ...	47	\@dGLspl: new	598
\GlsXtrIfFieldEqXpStr: new ..	51	\@dglS: new	597
\GlsXtrIfFieldNonZero: new ..	47	\@dglSpl: new	598
\GlsXtrIfHasNonZeroChildCount:		\@gls@getcounterprefix: new ..	36
new	584	\@gls@longextrawidestname:	
\GlsXtrIfXpFieldEqXpStr: new	51	new	702
\glsxtrpostlinkAddSymbolDescOnFirstUse		\@glsxtr@bibgls@removespaces:	
new	308	new	589
\GlsXtrRecordWarning: new ..	194	\@glsxtr@check@bibgls@nameref:	
\glsxtrRevertTocMarks: new ..	347	new	197

\@glstr@do@nameref@record:		\glslongextraLocationDescNameTabularHeader:	
new	10	new	708
\@glstr@get@prefixedlabel:		\glslongextraLocationDescSymNameHeader:	
new	596	new	720
\@glstr@if@record@only: new	12	\glslongextraLocationDescSymNameTabularFooter:	
\@glstr@ifnum@mmode: new	9	new	721
\@glstr@labelprefixes: new	593	\glslongextraLocationDescSymNameTabularHeader:	
\@glstr@prefixlabellist:		new	721
new	594	\glslongextraLocationFmt:	
\@glstr@providenewgls: new	215	new	698
\@glstr@record@only@setup:		\glslongextraLocationSymDescNameHeader:	
new	13	new	717
\@glstr@record@setting@nameref:		\glslongextraLocationSymDescNameTabularFooter:	
new	12	new	718
\@glstr@use@equation@counter@or:		\glslongextraLocationSymDescNameTabularHeader:	
new	101	new	717
\dGLS: new	598	\glslongextraLocSetDescWidth:	
\dglS: new	597	new	703
\dglSdisp: new	599	\glslongextraNameAlign: new	700
\dglSlink: new	598	\glslongextraNameDescHeader:	
\dGLSpl: new	598	new	701
equations: new	17	\glslongextraNameDescLocationHeader:	
floats: new	17	new	705
\glsadd: added grouping	105	\glslongextraNameDescLocationTabularFooter:	
ensure that \glsadd performs		new	705
indexing	106	\glslongextraNameDescLocationTabularHeader:	
\glslongextraDescAlign: new	700	new	705
\glslongextraDescFmt: new	698	\glslongextraNameDescSymHeader:	
\glslongextraDescNameHeader:		new	709
new	706	\glslongextraNameDescSymLocationHeader:	
\glslongextraDescNameTabularFooter:		new	711
new	707	\glslongextraNameDescSymLocationTabularFooter:	
\glslongextraDescNameTabularHeader:		new	711
new	706	\glslongextraNameDescSymLocationTabularHeader:	
\glslongextraDescSymNameHeader:		new	711
new	719	\glslongextraNameDescSymTabularFooter:	
\glslongextraDescSymNameTabularFooter:		new	710
new	719	\glslongextraNameDescSymTabularHeader:	
\glslongextraDescSymNameTabularHeader:		new	709
new	719	\glslongextraNameDescTabularFooter:	
\glslongextraGroupHeading:		new	701
new	701	\glslongextraNameDescTabularHeader:	
\glslongextraHeaderFormat:		new	701
new	701	\glslongextraNameFmt: new	698
\glslongextraLocationAlign:		\glslongextraNameSymDescHeader:	
new	701	new	712
\glslongextraLocationDescNameHeader:		\glslongextraNameSymDescLocationHeader:	
new	708	new	714
\glslongextraLocationDescNameTabularFooter:		\glslongextraNameSymDescLocationTabularFooter:	
new	708	new	714

\glslongextraNameSymDescLocationTabularHeader:	\glsxtrdisplaylocnameref:
new 714	new 587
\glslongextraNameSymDescTabularFooter:	\glsxtrfmtexternalnameref:
new 713	new 590
\glslongextraNameSymDescTabularHeader:	\glsxtrfmtinternalnameref:
new 713	new 590
\glslongextraSetDescWidth:	\GLSxTRhiername: new 68
new 702	\GLSxtrhiername: new 68
\glslongextraSetWidest: new . 701	\GlsXtrhiername: new 67
\glslongextraSubDescFmt: new 699	\Glsxtrhiername: new 67
\glslongextraSubLocationFmt:	\glsxtrhiername: new 66
new 700	\glsxtrhiernamesep: new 69
\glslongextraSubNameFmt: new 699	\glsxtridentifyglslike: new . 215
\glslongextraSubSymbolFmt:	\glsxtrifinlabelprefixlist:
new 699	new 594
\glslongextraSymbolAlign:	\GlsXtrLocationField: new . . 213
new 700	\glsxtrnameloclink: new 589
\glslongextraSymbolFmt: new . 698	\glsxtrnamereflink: new 588
\glslongextraSymDescNameHeader:	\glsxtrprependlabelprefix:
new 716	new 593
\glslongextraSymDescNameTabularFooter:	\GlsXtrSetAltModifier: write
new 716	modifier to aux 141
\glslongextraSymDescNameTabularHeader:	\glsxtrSetWidest: new 590
new 716	\glsxtrSetWidestFallback:
\glslongextraSymLocSetDescWidth:	new 592
new 703	\GlsXtrStandaloneEntryName:
\glslongextraSymSetDescWidth:	new 202
new 702	\GlsXtrStandaloneEntryOther:
\glslongextraTabularVAlign:	new 204
new 704	\GLSxtrusefield: new 49
\glslongextraUpdateWidest:	\GLSxtrusefield: fixed internal
new 702	command and added check
\glslongextraUpdateWidestChild:	for \texorpdfstring 49
new 702	\ifGlsLongExtraUseTabular:
\glsrenewcommand: new 584	new 704
\glsseeitemformat: removed	long-desc-name: new 707
reference to \glslabel 66	long-desc-sym-name: new 719
\glsxtr@dblfloat: new 17	long-loc-desc-name: new 708
\glsxtr@do@autoadd: new 101	long-loc-desc-sym-name: new . 721
\glsxtr@float: new 17	long-loc-sym-desc-name: new . 718
\glsxtr@record@nameref: new . 201	long-name-desc: new 704
\glsxtr@renewcommand: new . . 584	long-name-desc-loc: new 705
\glsxtr@writefields: provide	long-name-desc-sym: new 710
\glsxtr@record@nameref in	long-name-desc-sym-loc: new . 711
aux file 199	long-name-sym-desc: new 713
\glsxtraddlabelprefix: new . 593	long-name-sym-desc-loc: new . 714
\GlsXtrAutoAddOnFormat: new . 101	long-sym-desc-name: new 716
\glsxtrclearlabelprefixes:	1.38 – 2018-12-01
new 593	all: added glossary-longextra . . 642

\glslongextraNameFmt: bug fix: removed double param . . .	698	short-sc-desc: bug fix: omit description key as advertised in the manual	469
1.39 – 2019-03-22		short-sm-desc: corrected to omit description key as advertised in the manual	487
General: added label key for printgloss	182	1.40 – 2019-03-22	
\@@glxtr@dorecord: added protection for fragile commands	8	General: new	729
\@GlsXtrIfFieldCmpNum: new .	48	all: added glossary-topic	642
\@GlsXtrIfFieldEqNum: new . .	47	\glstopicAssignSubIndent: new	732
\@GlsXtrIfFieldEqStr: new . .	50	\glstopicAssignWidest: new .	732
\@GlsXtrIfFieldEqXpStr: new .	51	\glstopicCols: new	733
\@GlsXtrIfFieldNonZero: new .	47	\glstopicColsEnv: new	734
\@GlsXtrIfXpFieldEqXpStr: new	51	\glstopicDesc: new	731
\@gls@removespaces: changed \x to \@glo@tmp	190	\glstopicGroupHeading: new .	730
\glxtrbookindexlocation: new	690	\glstopicInit: new	731
\glxtrbookindexsublocation: new	690	\glstopicItem: new	731
\glxtrentryparentname: new .	49	\glstopicLoc: new	731
\GlsXtrIfFieldCmpNum: added starred version	48	\glstopicMarker: new	731
\GlsXtrIfFieldEqNum: added starred version	47	\glstopicMidSkip: new	733
\GlsXtrIfFieldEqStr: added starred form	50	\glstopicName: new	731
\GlsXtrIfFieldEqXpStr: added starred form	51	\glstopicParIndent: new	731
\GlsXtrIfFieldNonZero: added starred version	47	\glstopicPostSkip: new	733
\GlsXtrIfXpFieldEqXpStr: added starred form	51	\glstopicPreSkip: new	733
\glxtrsetglossarylabel: new	183	\glstopicSubIndent: new	731
\glxtrshortdescname: corrected to show long form as advertised in the manual . .	456	\glstopicSubItem: new	733
\s@GlsXtrIfFieldCmpNum: new .	48	\glstopicSubItemBox: new . . .	733
\s@GlsXtrIfFieldEqNum: new .	47	\glstopicSubItemSep: new . . .	733
\s@GlsXtrIfFieldEqStr: new .	51	\glstopicSubLoc: new	733
\s@GlsXtrIfFieldEqXpStr: new	51	\glstopicSubNameFont: new . .	733
\s@GlsXtrIfXpFieldEqXpStr: new	52	\glstopicTitleFont: new	731
short-desc: corrected to omit description key as advertised in the manual	457	\glstopicwidest: new	732
short-em-desc: bug fix: omit description key as advertised in the manual	508	topic: new	729
		topicmcols: new	734
		1.40 – 2019-03-31	
		\glsfirstabbrvdefaultfont: changed definition from \glsabbrvfont to \glsabbrvdefaultfont for consistency	320
		\GlsXtrDefaultResourceOptions: new	196
		long-hyphen-noshort-noreg: corrected formatting commands	554
		\printunsrtabbreviations: new	583
		\printunsrtacronyms: new . . .	583
		\printunsrtindex: new	583

\printunsrtnumbers: new	583	\@gls@setup@default@access	
\printunsrtsymbols: new	583	261
1.41 – 2019-04-09		\@glslink: switched from	
bookindex: changed		\glsdohyperlink to	
\thisgrptitle to		\glsxtrdohyperlink	145
\glsxtrcurrentgrptitle .	695	\@glsxtr@abbrlists: new	171
\glslistgroupskip: new	644	\@glsxtr@acronymlists: new .	171
\glstopicAssignSubIndent:		\@glsxtr@addabbreviationlist:	
moved \par from		new	172
\glstopicSubItem	732	\@glsxtr@base@acrcmd: new ..	127
\glstopicSubItem: added check		\@glsxtr@doloadprefix: new .	24
for description	733	\@glsxtr@org@addtoacronymlists:	
moved \par to		new	171
\glstopicAssignSubIndent	733	\@glsxtr@org@setacronymlists:	
\glstopicSubLoc: moved \space		new	171
to \glstopicSubPreLocSep	733	\@glsxtrentryfmt: added	
\glstopicSubPreLocSep: new .	733	\glslabel and scope	42
\glstreeChildDescLoc: new ..	661	debug: showaccsupp	28
\glstreeDescLoc: new	660	footnote: added missing text	
\glstreegroupskip: new	657	key	448
\glstreePreHeader: new	657	footnote-desc: new	451
\glsxtralttreeSymbolDescLocation:		\forallabbbrlists:	
added check for description	666	new	172
topic: added penalty if no		\forallacronyms: new	172
description	730	\glsdefaultshortaccess: new .	260
topicmcols: added penalty if no		\glsdisplaynumberlist: added	583
description	734	\glsenablehyper: switched from	
1.42 – 2020-02-03		\glsdohyperlink to	
General: added \@afterheading	681	\glsxtrdohyperlink	145
\@glsxtr@record: moved label		\glsentrynumberlist: added ..	584
definition outside of		\GLSfmtfirst: new	365
conditional	6	\GLSfmtfirsttpl: new	365
\@ACRfull: added redefinition .	132	\GLSfmtfull: new	367
\@ACRfullpl: added redefinition	132	\Glsfmtfull: switched pdf case	
\@Acrfull: added redefinition .	132	to use \glspdfmtfull ...	367
\@Acrfullpl: added redefinition	132	\glsfmtfull: switched pdf case	
\@GlsXtrIfFieldValueInCsvList:		to use \glspdfmtfull ...	367
new	45	\GLSfmtfullpl: new	368
\@acrfull: added redefinition .	132	\Glsfmtfullpl: switched pdf case	
\@acrfullpl: added redefinition	132	to use \glspdfmtfullpl .	367
\@domakeglossaries: provided		\glsfmtfullpl: switched pdf case	
definition for		to use \glspdfmtfullpl .	367
\@domakeglossaries	175	\GLSfmtlong: new	366
\@gls@assign@actual: new ...	261	\GLSfmtlongpl: new	366
\@gls@entry@field: redefined .	56	\GLSfmtname: new	364
\@gls@setup@default@access:		\GLSfmtplural: new	365
added		\GLSfmttext: new	364
\glsdefaultshortaccess .	261	\glspdfmtfull: new	367
\@gls@setup@default@short@access:		\glspdfmtfullpl: new	367
renamed to			

\glseeitemformat: switched to using \glsfmttext and \glsfmtname	66	\GLSXTRhiername: switched to using \GLSfmttext and \GLSfmtname	68
\glsshowtarget: added check for \glsshowtargetouter	33	\GLSXtrhiername: switched to using \glsfmttext, \glsfmtname, \GLSfmttext and \GLSfmtname	68
\glstreeChildDescLoc: added \glstreeNoDescSymbolPreLocation	661	\GlsXtrhiername: switched to using \Glsfmttext and \Glsfmtname	67
\glstreegroupheaderskip: new	657	\GLsxtrhiername: switched to using \glsfmttext and \glsfmtname	67
\glstreeNoDescSymbolPreLocation: new	660	\glsxtrhiername: switched to using \glsfmttext and being obtained	313
\glsxtr@newabbreviation: moved apply abbreviation style to after category key has removed \relax and updated \@gls@short instead of \glsshorttok	314	\GlsXtrIfFieldValueInCsvList: new	45
replaced explicit \spacefactor with \@	314	\glsxtrpdfentryfmt: new	42
\glsxtr@writefields: added check for order=letter	200	\glsxtrprovideaccsuppcmd: new	263
\glsxtrAccSuppAbbrSetFirstLongAttrs: new	264, 280	\glsxtrscs suffix: added \protect	463
\glsxtrAccSuppAbbrSetNameLongAttrs: new	264, 280	\GlsXtrSetAltModifier: added check	141
\glsxtrAccSuppAbbrSetNameShortAttrs: new	264, 280	\GLSXtrtitlefirst: new	357
\glsxtrAccSuppAbbrSetNoLongAttrs: new	264, 280	\GLSXtrtitlefirstplural: new	358
\glsxtrAccSuppAbbrSetTextShortAttrs: new	264, 280	\GLSXtrtitlefull: new	361
\glsxtralttreeSymbolDescLocation: switched to using \glstreeDescLoc	666	\GLSXtrtitlefullpl: new	362
\glsxtrassignactualsetup: new	260	\GLSXtrtitlelong: new	359
\glsxtrbookindexbookmarkprefix: new	692	\GLSXtrtitlelongpl: new	360
\GlsXtrDiscardUnsetBuffering: new	155	\GLSXtrtitlelongpl: new	353
\glsxtrdohyperlink: new (was former redefinition of \glsdohyperlink)	143	\GLSXtrtitlename: new	355
\glsxtrequationlocfmt: new .	588	\GLSXtrtitleplural: new	352
\glsxtrfieldformatcsvlist: new	44	\GLSXtrtitleshort: new	352
\glsxtrfieldformatlist: new .	43	\GLSXtrtitleshortpl: new	354
\glsxtrfootnotedesname: new	450	\GLSXtrtitletext: new	69
\glsxtrfootnotedes sort: new	450	\glsxtrusealias: new	69
		long-em-noshort-em: removed \protect from \glsxtremsuffix	513
		long-em-noshort-em-desc: removed \protect from \glsxtremsuffix	517
		long-em-short-em: added missing text key	501
		removed \protect from \glsxtremsuffix	502
		long-hyphen-noshort-desc-noreg: added missing text key	547

long-hyphen-postshort-hyphen:		\makenoidxglossaries: added	
added missing text key ...	557	\@domakeglossaries	77
long-hyphen-short-hyphen:		postfootnote: added missing	
added missing text key ...	540	text key	451
long-noshort-em: removed		prefix: new	24
\protect from		\RestoreAcronyms: added display	
\glxstremsuffix	511	style	173
long-noshort-em-desc: removed		\s@GlsXtrIfFieldValueInCsvList:	
\protect from		new	45
\glxstremsuffix	515	\seealso: add check for	
long-noshort-sc: moved		\also:	72
\protect inside		short-em: removed \protect	
\glxtrscsuffix	472	from \glxstremsuffix ...	507
long-noshort-sc-desc: moved		short-em-desc: removed	
\protect inside		\protect from	
\glxtrscsuffix	474	\glxstremsuffix	509
long-noshort-sm: removed		short-em-footnote: added	
\protect from		missing text key	519
\glxtrmsuffix	490	removed \protect from	
long-noshort-sm-desc: removed		\glxstremsuffix	520
\protect from		short-em-footnote-desc: new .	521
\glxtrmsuffix	491	short-em-long: added missing	
long-only-short-only: added		text key	503
missing text key	577	removed \protect from	
removed \protect from		\glxstremsuffix	504
\glxtronlysuffix	578	short-em-long-em: added	
long-postshort-user: added		missing text key	505
missing text key	528	removed \protect from	
long-short: added missing text		\glxstremsuffix	506
key	443	short-em-postfootnote: added	
long-short-em: added missing		missing text key	522
text key	499	removed \protect from	
removed \protect from		\glxstremsuffix	523
\glxstremsuffix	499	short-em-postfootnote-desc:	
long-short-sc: added missing		new	524
text key	464	short-footnote-desc: new ...	450
moved \protect inside		short-hyphen-long-hyphen:	
\glxtrscsuffix	464	added missing text key ...	564
long-short-sm: added missing		short-hyphen-postlong-hyphen:	
text key	481	added missing text key ...	571
removed \protect from		short-long: added missing text	
\glxtrmsuffix	482	key	445
long-short-user: added missing		short-long-user: added missing	
text key	527	text key	536
\makeglossaries: added		short-postfootnote-desc:	
\@domakeglossaries	175	added missing text key ...	453
let \@makeglossary to		new	453
\@gobble instead of \relax	176	short-postlong-user: added	
removed redefinition of		missing text key	533
\makeglossary	176		

short-sc: moved \protect inside		added leveloffset key	183
\glxtrscsuffix	468	\@glxtr@assign@leveloffset:	
short-sc-desc: moved \protect		new	183
inside \glxtrscsuffix	470	\@glxtr@leveloffset: new	183
short-sc-footnote: added		\@glxtr@noidx@do: replaced	
missing text key	475	\ifglshasparent with	
moved \protect inside		\@glxtr@ifischild	214
\glxtrscsuffix	476	\@print@unsrt@innnerglossary:	
short-sc-footnote-desc: new	477	new	209
short-sc-long: added missing		\doifglossarynoexistsordo:	
text key	466	switched to starred form of	
moved \protect inside		\ifglossaryexists	63
\glxtrscsuffix	466	\glswriteentry: replaced	
short-sc-postfootnote: added		\ifglused with	
missing text key	478	\GlsXtrIfUnusedOrUndefined	
moved \protect inside			139
\glxtrscsuffix	479	\glxtr@printgloss@checkexists:	
short-sc-postfootnote-desc:		new	180
new	480	\glxtralttreeSymbolDescLocation:	
short-sm: removed \protect		removed duplicate	
from \glxtrsmsuffix	486	description	666
short-sm-desc: removed		\ifglossaryexists: added check	
\protect from		for starred form	38
\glxtrsmsuffix	487	\np@glxtr@assign@leveloffset:	
short-sm-footnote: added		new	183
missing text key	493	\p@glxtr@assign@leveloffset:	
removed \protect from		new	183
\glxtrsmsuffix	494	\pp@glxtr@assign@leveloffset:	
short-sm-footnote-desc: new	495	new	183
short-sm-long: added missing		\printunsrtglossary: added	
text key	483	check for	
removed \protect from		\@printgloss@checkexists	204
\glxtrsmsuffix	484	printunsrtglossarywrap: new	207
short-sm-postfootnote: added		\printunsrtinnnerglossary:	
missing text key	496	new	207
removed \protect from		1.45 – 2020-04-01	
\glxtrsmsuffix	496	General: removed duplicate	
short-sm-postfootnote-desc:		description	664
new	498	\glstreenonameChildDescLoc:	
1.42 – 2020-02-13		new	664
\@glossentrysymbol: new	303	\glstreenonameDescLoc: new	663
\glsentrypdsymbol: new	302	1.46 – 2021-09-18	
1.42 – ?		\@glxtrsetaliasnoindex:	
postfootnote-desc: new	454	changed to use starred version	
1.43 – 2020-02-28		of \glxtrifhasfield	137
\@glxtrentryfmt: changed \def		1.46 – 2021-09-20	
to \edef to avoid infinite		General: changed \edef to	
recursion	42	\protected@edef	74, 311
1.44 – 2020-03-23		\@@glxtr@record: changed	
General: added groups key	183	\edef to \protected@edef	6

\@newglossaryentry@defunitcounters:	\@glxstr@rglstrigger@record:
changed \edef to	changed \edef to
\protected@edef 164	\protected@edef 221
\@glossentrysymbol: changed	\@glxstr@warn@hybrid@noprintgloss:
\edef to \protected@edef 303	new 13
\@gls@increment@currunitcount:	\@glxstr@entryfmt: changed
changed \edef to	\edef to \protected@edef . 42
\protected@edef 164	\@glxstr@glossentry: changed
\@gls@link: changed \edef to	\edef to \protected@edef 202
\protected@edef 102, 104	\@glxstr@glossentryother:
\@gls@link@checkfirsthyper:	changed \edef to
changed \edef to	\protected@edef 204
\protected@edef 135	\@glxstr@indexaliased: changed
\@gls@local@increment@currunitcount:	\edef to \protected@edef 137
changed \edef to	\@makeglossaries@warn@noprintglossary:
\protected@edef 165	new 174
\@gls@setup@default@access:	\@newglossaryentryposthook:
changed \edef to	changed \edef to
\protected@edef 261, 262	\protected@edef 75
\@glxstr@addabbreviationlist:	\@print@unsrt@glossary:
changed \eappto to	changed \eappto to
\protected@eappto 172	\protected@eappto 206
changed \edef to	\@print@unsrt@innerglossary:
\protected@edef 172	changed \eappto to
\@glxstr@bibgls@removespaces:	\protected@eappto 209
changed \x to \@glo@tmp .. 589	\@print@unsrt@glossary@handler:
\@glxstr@do@inc@linkcount:	changed \xdef to
changed \x to \@glo@tmp .. 226	\protected@xdef 210
\@glxstr@do@record@wrglossary:	\glossentrydesc: changed \edef
changed \edef to	to \protected@edef .. 290, 291
\protected@edef 5	\Glossentryname: changed \edef
\@glxstr@do@redef@for@gl@sentries:	to \protected@edef .. 294, 295
changed \edef to	\glossentryname: changed \edef
\protected@edef 4	to \protected@edef .. 291, 293
\@glxstr@get@prefixedlabel:	\glossentrynameother: changed
changed \edef to	\edef to \protected@edef 296
\protected@edef 596	\glsadd: changed \edef to
changed \x to \@glo@tmp ... 597	\protected@edef 105
\@glxstr@get@prefixedlabel@field:	\glsalttreechildpredesc: new 665
changed \x to \@glo@tmp .. 601	\glsalttreepredesc: new 665
\@glxstr@mixed@assign@sortkey:	\glsdisablehyper: changed
changed \edef to	\edef to \protected@edef 145
\protected@edef 184	\glsdoifexists: changed \edef
\@glxstr@op@recordcounter:	to \protected@edef 62
changed \eappto to	\glsenableentryunitcount:
\protected@eappto 11	changed \edef to
\@glxstr@orgprintglossary:	\protected@edef 167
changed \xdef to	\glsFindWidestLevelTwo:
\protected@xdef 180	changed \edef to
	\protected@edef 671

<code>\glsFindWidestUsedLevelTwo:</code>	changed <code>\edef</code> to		changed <code>\edef</code> to	
<code>\protected@edef</code>	670		<code>\protected@edef</code>	176, 177
<code>\glsnavhyperlink:</code> changed			<code>\makenoidxglossaries:</code> changed	
<code>\edef</code> to <code>\protected@edef</code>	142		<code>\edef</code> to <code>\protected@edef</code> .	78
<code>\glstopicAssignSubIndent:</code> bug			<code>printunsrtglossarywrap:</code>	
182 maintain hangindent for			changed <code>\xdef</code> to	
multiple paragraphs	732		<code>\protected@xdef</code>	208
<code>\glstopicSubItemParIndent:</code>			<code>record:</code> added hybrid	14
new	732		<code>\setabbreviationstyle:</code> changed	
<code>\glstopicSubItemParIndent:</code>			<code>\edef</code> to <code>\protected@edef</code>	340
new	732		<code>topic:</code> added <code>\par</code> (bug 176) . .	730
<code>\glsxtr@org@newignoredglossary:</code>			grouping added to scope	
changed <code>\eappto</code> to			<code>\everypar</code> (bug 182)	730
<code>\protected@eappto</code>	59	1.47 –	<code>\@GlsXtrIfValueInFieldCsvList:</code>	
changed <code>\edef</code> to			new	45
<code>\protected@edef</code>	59		<code>\@xGlsXtrIfValueInFieldCsvList:</code>	
<code>\glsxtr@provideignoredglossary:</code>			new	46
changed <code>\eappto</code> to			<code>\s@GlsXtrIfValueInFieldCsvList:</code>	
<code>\protected@eappto</code>	61		new	46
changed <code>\edef</code> to			<code>\s@xGlsXtrIfValueInFieldCsvList:</code>	
<code>\protected@edef</code>	61		new	46
<code>\glsxtr@s@newignoredglossary:</code>		1.47 – 2021-11-04	<code>\@GlsXtrIfHasNonZeroChildCount:</code>	
changed <code>\edef</code> to			new	584
<code>\protected@edef</code>	60		<code>\@glsxtrcopytoglossary:</code>	
<code>\glsxtr@s@provideignoredglossary:</code>			replaced <code>\cseappto</code> with	
changed <code>\edef</code> to			<code>\protected@cseappto</code>	62
<code>\protected@edef</code>	61		<code>\@glsxtrforcsvfield:</code> new . . .	44
<code>\glsxtr@setaccessdisplay:</code>			<code>\@glsxtrsetaliasnoindex:</code>	
changed <code>\edef</code> to			changed to use <code>\ifcvoid</code> .	137
<code>\protected@edef</code>	296		<code>\glsaltlistitem:</code> new	645
<code>\glsxtralttreeSymbolDescLocation:</code>			<code>\glslistexpandedname:</code> new . .	645
switch to using			<code>\glslistgroupafterheader:</code>	
<code>\glsalttreepredesc</code> and			new	646
<code>\glsalttreechildpredesc</code>	665		<code>\glslistgroupheaderitem:</code> new	646
<code>\glsxtrdisplayendloc:</code> changed			<code>\glslistinit:</code> new	644
<code>\edef</code> to <code>\protected@edef</code>	190		<code>\glslistitem:</code> new	644
<code>\glsxtrdisplaystartloc:</code>			<code>\glsseefirstitem:</code> new	71
changed <code>\edef</code> to			<code>\glsseelastoxfordsep:</code> new . .	72
<code>\protected@edef</code>	190		<code>\glsseelist:</code> redefined	70
<code>\glsxtrdoautoindexname:</code>			<code>\glsunsetcategoryattribute:</code>	
changed <code>\eappto</code> to			new	283
<code>\protected@eappto</code>	298		<code>\glsxtrapptocsvfield:</code> new . .	49
<code>\glsxtrseelist:</code> changed <code>\edef</code>			<code>\glsxtrfieldtitlecasecs:</code>	
to <code>\protected@edef</code>	70		added check for	
<code>\glsxtrtreechildpredesc:</code> new	660		<code>\glscapitalisewords</code>	289
<code>\glsxtrtreepredesc:</code> new	660		<code>\GlsXtrIfHasNonZeroChildCount:</code>	
<code>\makeglossaries:</code> adjust warning			added starred version	584
on missing glossary for				
“alsoindex”	175			

\GlsXtrIfValueInFieldCsvList:	\@gls@combined@postlinks:
new 45	new 378
\s@glstrforcsvfield: new .. 44	\@gls@combined@textformat:
\s@GlsXtrIfFieldNonZero: new 47	new 376
\s@GlsXtrIfHasNonZeroChildCount:	\@gls@combined@usedprefix:
new 584	new 377
\xGlsXtrIfValueInFieldCsvList:	\@gls@combined@usedskipmain:
new 46	new 377
1.48 – 2021-11-22	\@gls@combined@usedskipothers:
\@@gls@navhypertarget: new . 142	new 377
\@mgls@hyperlink: new 388	\@gls@combined@usedsuffix:
\@GlsXtrMglsOrGls: new 413	new 377
\@Glsfieldorgls: new 418	\@gls@do@glunset: new 136
\@Glsfullorfirst@: new 416	\@gls@restore@glsllocal: new . 136
\@Glslongortext@: new 416	\@gls@save@glsllocal: new ... 136
\@Glsshortortext@: new 416	\@glsfieldorgls: new 418
\@PGLSorgls: new 420	\@glsfullorfirst@: new 416
\@PGLSorglspl: new 420	\@glslongortext@: new 415
\@Pglsorgls: new 419	\@glsnavhypertarget: added
\@Pglsorglspl: new 420	patch 142
\@alt@GlsXtrMglsOrGls: new . 414	\@glsshortortext@: new 416
\@def@multi@glossaryentry:	\@glsshowtarget: new 33
new 379	\@glsshowtargetmarkfmt: new . 34
\@defmultiglossaryentry: new 379	\@glssymbolorgls: new 418
\@firstofthree: new 394	\@glssymbolorgls: new 418
\@gls@combined@category: new 376	\@glstr@addunused: added
\@gls@combined@encapmain:	check for multientry labels .. 76
new 375	\@glstr@do@org@target: new . 184
\@gls@combined@encapothers:	\@glstr@doshowtarget: new . 28
new 376	\@glstr@mglslake: new 413
\@gls@combined@firstprefix:	\@glstr@mglssrefs: new 411
new 377	\@glstr@mglswrite: new 411
\@gls@combined@firstskipmain:	\@glstr@multientry: new ... 383
new 377	\@glstr@seefirstitem: new . 71
\@gls@combined@firstskipothers:	\@glstr@seeitem: new 71
new 377	\@glstrmultientryadjustedname:
\@gls@combined@firstsuffix:	new 605
new 377	\@glstrshowtargetleft: new . 29
\@gls@combined@hyper: new .. 375	\@glstrshowtargetmark: new . 29
\@gls@combined@indexmain:	\@glstrshowtargetright: new 29
new 375	\@mgls@all: new 387
\@gls@combined@indexothers:	\@mgls@disable@writeseparateref@cond:
new 375	new 412
\@gls@combined@mglsopts: new 376	\@mgls@hyper: new 388
\@gls@combined@mglsopts@do:	\@mgls@hyperlink: new 388
new 376, 377	\@mgls@main: new 387
\@gls@combined@mpostlink:	\@mgls@others: new 387
new 378	\@mgls@resetall: new 389
\@gls@combined@mpostlinkelement:	\@mgls@resetmain: new 390
new 378	\@mgls@resetothers: new 390

\@mgl@setup: new	387	\glsshowtarget: removed check	
\@mgl@setup@do: new	387	for \glsshowtargetouter	33
\@mgl@setup@do@not: new	387	\glsshowtargetfont: new	33
\@mgl@unsetaction: new	388	\glsshowtargetinner: new	33
\@mgl@unsetall: new	390	\glsshowtargetinnersymleft:	
\@mgl@unsetmain: new	390	new	33
\@mgl@unsetothers: new	391	\glsshowtargetinnersymright:	
\@mgl@localreset: new	385	new	33
\@mgl@localunset: new	385	\glsshowtargetouter: new	33
\@mgl@reset: new	384	\glxtr@mgl@applyopts: new	393
\@mgl@unset: new	384	\glxtr@mgl@checklastelement:	
\@multi@glossary@doifexists:		new	392
new	380	\glxtr@mgl@inner: new	394
\@multi@glossary@entry: new	380	\glxtr@newmgl: new	412
\@multi@glossary@entry: new	379	\glxtr@setup@docurrent: new	391
\@multi@glossary@entry@list:		\glxtr@dohyperlink: added	
new	380	check for multi-entry	144
\@multiglossary@entry: new	379	\glxtr@ifmulti: new	378
\@pglsorgls: new	419	\glxtr@longshortscuserdescname:	
\@pglsorglspl: new	419, 420	new	532
\@provide@multi@glossary@entry@noop:		\glxtr@longshortscuserdescname:	
new	380	new	529
\@secondofthree: new	394	\GlsXtrMglOrGls: new	413
\@thirdofthree: new	394	\glxtr@mgl@WarnAllSkipped:	
\alt@GlsXtrMglOrGls: new	413	new	393
\gl@abbrvs@onlyfont: new	580	\GLSxtr@multientry@adjustedname:	
\gl@abbrvs@userfont: new	529	new	605
\gl@combined@firstsep: new	408	\GlsXtr@multientry@adjustedname:	
\gl@combined@firstsep@first:		new	605
new	408	\Glsxtr@multientry@adjustedname:	
\gl@combined@sep: new	407	new	604
\gl@combined@sep@first: new	408	\glxtr@multientry@adjustedname:	
\gl@doshowtarget: new	33	new	604
\gl@first@abbrvs@onlyfont:		\GLSxtr@multientry@adjustednamefmt:	
new	580	new	607
\gl@first@abbrvs@userfont:		\GlsXtr@multientry@adjustednamefmt:	
new	529	new	607
\gl@link@wrcontent: new	101	\Glsxtr@multientry@adjustednamefmt:	
\gl@nav@hypertarget: new	142	new	606
\gl@set@categories@attribute:		\glxtr@multientry@adjustednamefmt:	
new	282	new	606
\gl@set@categories@attributes:		\GLSxtr@multientry@adjustedname@other:	
new	282	new	607
\gl@set@combined@sep@abbrv@nbsp:		\GlsXtr@multientry@adjustedname@other:	
new	408	new	607
\gl@set@combined@sep@abbrv@none:		\Glsxtr@multientry@adjustedname@other:	
new	409	new	606
\gl@set@combined@sep@narrow:		\glxtr@multientry@adjustedname@other:	
new	409	new	606

\glstrmultientryadjustednamepostsep:	\mglselementprehook: new ...	407
new	\mglselementreset: new	389
\glstrmultientryadjustednamepresep:	\mglselementunset: new	389
new	\mgl\$field: new	418
\glstrmultientryadjustednamesep:	\mgl\$forelements: new	388
new	\mgl\$forotherelements: new ..	388
\glstrmultilasttotherindex:	\Mgl\$full: new	417
new	\mgl\$full: new	416
\glstrmultilist: new	\mgl\$hascategoryprefix: new ..	406
\glstrmultimain: new	\mgl\$hascategorysuffix: new ..	407
\glstrmultimainindex: new ..	\mgl\$lastelementpostlinkhook:	
\glstrmultitotalelements:	new	406
new	\mgl\$lastmainpostlinkhook:	
\glstrsconlydescname: new ..	new	406
\glstrsconlydescsort: new ..	\mgl\$localreset: new	385
\glstrsconlyname: new	\mgl\$localunset: new	384
\glstrsconlysuffix: new ...	\mgl\$localunsettothers: new ..	389
\glstrscusersuffix: new ...	\Mgl\$long: new	417
\glstrshowtargetinner: new ..	\mgl\$long: new	416
\glstrshowtargetouter: new ..	\MGL\$mainpl: new	415
\glstrshowtargetsymbolleft:	\MGL\$mainpl: new	415
new	\Mgl\$mainpl: new	414
\glstrshowtargetsymbolright:	\mgl\$mainpl: new	414
new	\MGL\$name: new	417
\if@mgl\$@writeseparaterefs:	\Mgl\$name: new	417
new	\mgl\$name: new	417
\ifKV@mgl\$@presetlocal: new ..	\MGL\$pl: new	415
\ifmgl\$used: new	\MGL\$pl: new	415
\ifmultiglossaryentryglobal:	\Mgl\$pl: new	414
new	\mgl\$pl: new	414
long-only-short-sc-only: new	\mgl\$prefix: new	407
long-only-short-sc-only-desc:	\mgl\$reset: new	384
new	\mgl\$resetall: new	385
long-postshort-sc-user: new ..	\mgl\$seefirstitem: new	71
long-postshort-sc-user-desc:	\mgl\$seeitem: new	71
new	\mgl\$SetMain: new	385
\MGL\$: new	\mgl\$SetOptions: new	386
\MGLs: new	\Mgl\$short: new	417
\Mgl\$: new	\mgl\$short: new	416
\mgl\$: new	\mgl\$suffix: new	407
\mgl\$@disable@mgl\$opts: new ..	\MGL\$symbol: new	418
\mgl\$@disable@setup: new ...	\Mgl\$symbol: new	418
\mgl\$@enable@mgl\$opts: new ..	\mgl\$symbol: new	418
\mgl\$@enable@setup: new	\mgl\$sunset: new	384
\mgl\$AddOptions: new	\mgl\$sunsetall: new	385
\mgl\$custompostlinkhook: new	\mgl\$sunsettothers: new	389
\mgl\$defcategoryprefix: new ..	\mgl\$usecategoryprefix: new ..	406
\mgl\$defcategorysuffix: new ..	\mgl\$usecategorysuffix: new ..	407
\mgl\$elementindex: new	\MGL\$usefield: new	419
\mgl\$elementposthook: new ..	\Mgl\$usefield: new	419

\mglseusefield: new	419	\@glxstr@dorecord: new	8
\mglswriteSeparateRefsFalse:		\@glxstr@setup@bibglsaux:	
new	412	new	15
\mglswriteSeparateRefsTrue:		\@glxstrbuffer@check@repeats:	
new	412	new	154
\MPGLS: new	421	\@glxstrbuffer@check@repeats@notused:	
\MPGLs: new	421	new	154
\Mpgls: new	420	\@glxstrbuffer@do@check@repeat:	
\mpgls: new	420	new	154
\MPGLSmainpl: new	422	\@glxstrwrglosscountermark:	
\MPGLsmainpl: new	421	new	27
\Mpglsmainpl: new	421	\@newglossaryentry@defcounters:	
\mpglsmainpl: new	420	new	159
\MPGLSpl: new	421	\@GLSXTRhiername: new	69
\MPGLspl: new	421	removed unwanted eol	69
\Mpglspl: new	421	\@GLSXtrhiername: new	68
\mpglspl: new	420	\@GLsXtrhiername: new	68
\mpglsWarning: new	419	\@Glsentryfield: new	31
\multiglossaryentry: new	378	\@Glsxtrfmt: new	42
\multiglossaryentrysetup:		\@Glsxtrhiername: new	67
new	375	\@bibgls@write@aux: new	16
\p@GLsXtrMglsOrGls: new	413	\@d@inner@GLSfield: new	603
\providemultiglossaryentry:		\@d@inner@Glsfield: new	603
new	380	\@d@inner@glsfield: new	603
\s@GLsXtrMglsOrGls: new	413	\@dGLS@field: new	603
showtargets: new	29	\@dGLSfield: new	602
\writemultiglossentry: new	383	\@dGls@field: new	602
1.49 – 2022-10-14		\@dGlsdisp: new	599
General: add gettitlestring patch		\@dGlsfield: new	602
for		\@dGlslink: new	599
\glxstrtitleorpdforheading		\@dgls@field: new	602
.....	347	\@dglsdisp: new	599
added		\@dglsfield: new	602
\glssubgroupheading	645,	\@dglslink: new	598
647–655, 658, 659, 661–664,		\@gls@alt@hyp@opt: changed	
676, 677, 679–683, 685, 687, 688		\let to \def for	
added postamble key for		\@gls@hyp@opt@cs	140
printgloss	183	\@gls@assign@actual: removed	
added preamble key for		use of \pdfstringdef	261
printgloss	182	\@gls@automake@types: new	175
added flatten key	184	\@gls@default@glslink@opts:	
corrected name of		new	136
longpluralaccess field	259	\@gls@default@restore@glslocal:	
new	735	new	136
split shortplural and longplural		\@gls@do@glsprereset: new	97
into separate family	311	\@gls@do@glspreunset: new	98
\@dgls@: new	597	\@gls@glslink@hyper@update@hook:	
\@dgls@@field: new	601	new	98
\@gls@expand@field: added		\@gls@ignore@restore@glslocal:	
redefinition	57	new	136

\@gls@link@postkeys@checkfirsthyper:	\@glsxtr@unsrt@gloss@init:
new 135	new 206
\@gls@noexpand@field: added	save hierarchical level
redefinition 56	information 206
\@gls@warn@makegloss@incompatible:	\@glsxtrbuffer@check@repeats:
new 179	new 154
\@gls@warn@noidx@incompatible:	\@glsxtrcopytoglossary: new . 62
new 79	\@glsxtrglossentryother: bug
\@gls@xtr@initprocess: new . 206	fix: corrected arguments to
\@glslink@prefix@label: new . 102	\GlsXtrStandaloneEntryOther
\@glstable@defaultpreamble: 204
new 756	\@glsxtrhiername: new 67
\@glstable@finishrow: new .. 761	\@glsxtrnewglslink: new 216
\@glsuseri@: added redefinition 119	\@glsxtrnoidxgroups@nomakegloss:
\@glsuserii@: added	new 187
redefinition 120	\@glsxtrundefdebug: new 28
\@glsuseriii@: added	\@glsxtrwrglosscountermark:
redefinition 122	new 27
\@glsuseriv@: added	\@noglslink@prefix@label:
redefinition 123	new 102
\@glsuserv@: added redefinition 124	\@p@glossarysection: moved
\@glsuservi@: added	\phantomsection 55
redefinition 125	\@set@bibgls@write@aux: new . 16
\@glsxtr@assignMakeUppercase:	\@xp@gls@getcounterprefix:
new 32	new 36
\@glsxtr@current@innertextformat@csname:	abbr-long-short: new 727
new 100	abbr-short-long: new 725
\@glsxtr@current@textformat@csname:	bibglsaux: new 15
new 100	bookindex: added
\@glsxtr@dglslnomatch: new .. 594	\gls subgroupheading 696
\@glsxtr@field@linkdefs:	replaced \edef with
removed \glsinsert 106	\protected@edef 694
\@glsxtr@get@prefixedlabel@field:	\d@inner@GLSfield: new 603
new 599	\d@inner@Glsfield: new 603
\@glsxtr@inc@indexcount: new 139	\d@inner@glsfield: new 603
\@glsxtr@local@innertextformat:	\dGlsdisp: new 599
new 100	\dGLSfield: new 602
\@glsxtr@noidx@do: check if	\dGlsfield: new 602
location field has been	\dglslfield: new 602
changed 214	\dglslfieldactualfieldlabel:
\@glsxtr@providenewglsfamily:	new 602
new 216	\dglslfieldcurrentfieldlabel:
\@glsxtr@providenewglslink:	new 601
new 216	\dglslfieldfallbackfieldlabel:
\@glsxtr@restoreMakeUppercase:	new 602
new 32	\dGlslink: new 598
\@glsxtr@saveMakeUppercase:	\GlossariesAbbrStyleTooComplexWarning:
new 32	new 341
\@glsxtr@setup@bibglsaux: new 15	\gls@warn@makegloss@incompatible:
\@glsxtr@truevalue: new 285	new 179

\gls@warn@noidx@incompatible:	\Glsaccessfmtsymbolplural:
new 79	new 236, 270
\gls@warn@noidxmakegloss@incompatible:\glsaccessfmtsymbolplural:	
new 79	new 236, 269
\glsabsspace: new 174	\GLSaccessfmttext: new . 230, 266
\GLSaccessfmtdesc: new . 238, 271	\Glsaccessfmttext: new . 229, 266
\Glsaccessfmtdesc: new . 237, 270	\glsaccessfmttext: new . 229, 266
\glsaccessfmtdesc: new . 237, 270	\GLSaccessfmtuseri: new 247, 275
\GLSaccessfmtdescplural:	\Glsaccessfmtuseri: new 246, 275
new 239, 271	\glsaccessfmtuseri: new 245, 275
\Glsaccessfmtdescplural:	\GLSaccessfmtuserii: new 249, 276
new 239, 271	\Glsaccessfmtuserii: new 248, 276
\glsaccessfmtdescplural:	\glsaccessfmtuserii: new 247, 275
new 238, 271	\GLSaccessfmtuseriii:
\GLSaccessfmtfirst: new 233, 268	new 251, 277
\Glsaccessfmtfirst: new 232, 267	\Glsaccessfmtuseriii:
\glsaccessfmtfirst: new 232, 267	new 251, 277
\GLSaccessfmtfirstplural:	\glsaccessfmtuseriii:
new 234, 268	new 250, 276
\Glsaccessfmtfirstplural:	\GLSaccessfmtuseriv: new 254, 278
new 233, 268	\Glsaccessfmtuseriv: new 253, 277
\glsaccessfmtfirstplural:	\glsaccessfmtuseriv: new 252, 277
new 233, 268	\GLSaccessfmtuserv: new 256, 279
\GLSaccessfmtlong: new . 243, 274	\Glsaccessfmtuserv: new 255, 278
\Glsaccessfmtlong: new . 243, 273	\glsaccessfmtuserv: new 254, 278
\glsaccessfmtlong: new . 242, 273	\GLSaccessfmtuservi: new 258, 279
\GLSaccessfmtlongpl: new 244, 274	\Glsaccessfmtuservi: new 258, 279
\Glsaccessfmtlongpl: new 244, 274	\glsaccessfmtuservi: new 257, 279
\glsaccessfmtlongpl: new 243, 274	\GLSaccessuseri: new ... 246, 275
\GLSaccessfmtname: new . 229, 265	\Glsaccessuseri: new ... 245, 275
\Glsaccessfmtname: new . 228, 265	\glsaccessuseri: new ... 245, 275
\glsaccessfmtname: new . 228, 265	\GLSaccessuserii: new .. 249, 276
\GLSaccessfmtplural: new 231, 267	\Glsaccessuserii: new .. 248, 276
\Glsaccessfmtplural: new 231, 267	\glsaccessuserii: new .. 247, 275
\glsaccessfmtplural: new 230, 266	\GLSaccessuseriii: new . 251, 277
\GLSaccessfmtshort: new 241, 272	\Glsaccessuseriii: new . 250, 276
\Glsaccessfmtshort: new 240, 272	\glsaccessuseriii: new . 249, 276
\glsaccessfmtshort: new 240, 272	\GLSaccessuseriv: new .. 253, 278
\GLSaccessfmtshortpl:	\Glsaccessuseriv: new .. 252, 277
new 242, 273	\glsaccessuseriv: new .. 252, 277
\Glsaccessfmtshortpl:	\GLSaccessuserv: new ... 256, 278
new 241, 273	\Glsaccessuserv: new ... 255, 278
\glsaccessfmtshortpl:	\glsaccessuserv: new ... 254, 278
new 241, 272	\GLSaccessuservi: new .. 258, 279
\GLSaccessfmtsymbol: new 235, 269	\Glsaccessuservi: new .. 257, 279
\Glsaccessfmtsymbol: new 235, 269	\glsaccessuservi: new .. 256, 279
\glsaccessfmtsymbol: new 234, 269	\glsaddallunindexed: new ... 139
\GLSaccessfmtsymbolplural:	\glsalttreesubgroupitem: new 677
new 237, 270	\glsapptopostlink: new 307

\glsdefaultshortaccess:			
reverted to original definition	260	\glsгенentryfmt: added	
\glsdoifexists: added		redefinition	90
\glstrundefdebug	62	\glshashchar: new	584
\glsenableentryunitcount:		\glsifapplyinnerfmtfield: new	90
added		\glsifattributetrue: new	285
\ifglsresetcurrcount	167	\glsifcategoryattributehasitem:	
\glsencapwrcontent: new	139	new	285
\glsentryindexcount: new	139	\glsifcategoryattributetrue:	
\glsexclapplyinnerfmtfield:		new	285
new	90	\glsifindexed: new	139
\glsfirstinnerfmtabbrvfont:		\glsindexsetting: new	30
new	320	\glsindexsubgroupitem: new	658
\glsfirstinnerfmtlongfont:		\glsinitreunsets: new	99
new	321	\glsinnerfmtabbrvfont: new	321
\glsfirstxplongfont: new	320	\glsinnerfmtlongfont: new	321
\glsfirstxplongfont: new	321	\glslinkwrcontent: removed	
\GLSfmtfield: new	89	grouping	101
\Glsfmtfield: new	88	\glslongextraDescSymHeader:	
\glsfmtfield: new	88	new	725
\Glsfmtfirst: added		\glslongextraDescSymTabularFooter:	
\MFUsentencecase	365	new	725
\Glsfmtfirstpl: added		\glslongextraDescSymTabularHeader:	
\MFUsentencecase	365	new	725
\GLSfmtfull: add upper case		\glslongextraLongFmt: new	699
bookmark	367	\glslongextraLongHeader: new	727
\Glsfmtfull: added		\glslongextraLongShortHeader:	
\MFUsentencecase	367	new	728
\GLSfmtfullpl: add upper case		\glslongextraLongShortTabularFooter:	
bookmark	368	new	729
\Glsfmtfullpl: added		\glslongextraLongShortTabularHeader:	
\MFUsentencecase	367	new	728
\GLSfmtinsert: new	90	\glslongextraShortHeader:	
\glsfmtinsert: new	90	new	727
\Glsfmtlong: added		\glslongextraShortLongHeader:	
\MFUsentencecase	366	new	726
\Glsfmtlongpl: added		\glslongextraShortLongTabularFooter:	
\MFUsentencecase	366	new	727
\Glsfmtname: added		\glslongextraShortLongTabularHeader:	
\MFUsentencecase	363	new	727
\Glsfmtplural: added		\glslongextraShortNoNameSetDescWidth:	
\MFUsentencecase	364	new	703
\GLSfmtshort: new	363	\glslongextraShortTargetFmt:	
\Glsfmtshort: added		new	699
\MFUsentencecase	363	\glslongextraSubGroupHeading:	
\GLSfmtshortpl: new	363	new	701
\Glsfmtshortpl: added		\glslongextraSubLongFmt: new	700
\MFUsentencecase	363	\glslongextraSubShortTargetFmt:	
\Glsfmtttext: added		new	700
\MFUsentencecase	364	\glslongextraSubSymbolOrName:	
		new	700

<code>\glslongextraSubSymbolTargetFmt:</code>	<code>\glstableblockentry: new ...</code>	741
new	<code>\glstableblockheader: new ..</code>	741
<code>\glslongextraSymbolNameAlign:</code>	<code>\glstableblockperrowcount:</code>	
new	new	735
<code>\glslongextraSymbolOrName:</code>	<code>\glstableblocksubentry: new ..</code>	741
new	<code>\glstableblocksubentrysep:</code>	
<code>\glslongextraSymbolTargetFmt:</code>	new	736
new	<code>\glstableblockwidth: new ...</code>	754
<code>\glslongextraSymDescHeader:</code>	<code>\glstablecaption: new</code>	752
new	<code>\glstablecenteralign: new ..</code>	737
<code>\glslongextraSymDescTabularFooter:</code>	<code>\glstableChildEntries: new ..</code>	760
new	<code>\glstablecolsperblock: new ..</code>	741
<code>\glslongextraSymDescTabularHeader:</code>	<code>\glstablecurrentblockindex:</code>	
new	new	735
<code>\glslongextraSymNoNameSetDescWidth:</code>	<code>\glstableDesc: new</code>	740
new	<code>\glstabledescCOLalign: new ..</code>	737
<code>\glslowercase: new</code>	<code>\glstableDescFmt: new</code>	740
31	<code>\glstabledescheader: new ...</code>	735
<code>\glsmfuaddmap: new</code>	<code>\glstabledescwidth: new</code>	754
31	<code>\glstablefinishlengthupdates:</code>	
<code>\glsmfublocker: new</code>	new	741
31	<code>\glstablefinishrow: new</code>	761
<code>\glsmfuexcl: new</code>	<code>\glstablefirstthead: new</code>	753
31	<code>\glstablefoot: new</code>	753
<code>\glspretopostlink: new</code>	<code>\glstableGroupHeaderFmt: new</code>	756
307	<code>\glstablegroupheading: new ..</code>	756
<code>\glssentencecase: new</code>	<code>\glstablehead: new</code>	753
31	<code>\glstableHeaderFmt: new</code>	741
<code>\glsssetcategoryattributes:</code>	<code>\glstableiffilter: new</code>	754
new	<code>\glstableifmeasuring: new ..</code>	754
283	<code>\glstableifpar: new</code>	736
<code>\glsssetcombinedsepabbrvnbs:</code>	<code>\glstableinitlengthupdates:</code>	
corrected spelling of	new	741
<code>\ifglshasshort</code>	<code>\glstablelastfoot: new</code>	753
408	<code>\glstableleftalign: new</code>	736
<code>\glsssetcombinedsepabbrvnone:</code>	<code>\glstablelengthupdate: new ..</code>	741
corrected spelling of	<code>\glstablemeasureandupdate:</code>	
<code>\ifglshasshort</code>	new	754
409	<code>\glstablenameCOLalign: new ..</code>	737
<code>\glssshowtargetfonttext: new ..</code>	<code>\glstableNameFmt: new</code>	737
33	<code>\glstablenameheader: new ...</code>	735
<code>\glssshowtargetinner: added</code>	<code>\glstableNameNoDesc: new ...</code>	739
check for math mode	<code>\glstableNameSingleFmt: new ..</code>	737
33	<code>\glstableNameSinglePostName:</code>	
<code>\glssubgroupheading: new ...</code>	new	738
213	<code>\glstableNameSinglePostSubName:</code>	
<code>\glstable@begin: new</code>	new	739
753	<code>\glstableNameSingleSubSuppl:</code>	
<code>\glstable@blockalignsep: new</code>	new	739
736		
<code>\glstable@child: new</code>		
760		
<code>\glstable@filter: new</code>		
753		
<code>\glstable@finishrow: new ...</code>		
761		
<code>\glstable@groupheading: new ..</code>		
756		
<code>\glstable@ifhaspreamble: new</code>		
755		
<code>\glstable@init: new</code>		
755		
<code>\glstable@n@amps: new</code>		
760		
<code>\glstable@parcase: new</code>		
736		
<code>\glstable@stepentry: new ...</code>		
754		
<code>\glstable@stepsubentry: new ..</code>		
754		
<code>\glstableblockalign: new ...</code>		
741		

\glstableNameSingleSuppl:		
new	738	
\glstableNameSingleSymSep:		
new	738	
\glstableNameTarget: new	737	
\glstableNameWidth: new	754	
\glstableNewStyle: new	742	
\glstableNextCaption: new	752	
\glstableOther: new	737	
\glstableOtherField: new	737	
\glstableOtherHeader: new	735	
\glstableOtherNoDesc: new	740	
\glstableOtherSep: new	738	
\glstablePostNextCaption:		
new	752	
\glstablePreChildren: new	736	
\glstableRightAlign: new	736	
\glstableRowspan: new	753	
\glstableSetStyle: new	741	
\glstableSubDesc: new	740	
\glstableSubDescFmt: new	740	
\glstableSubEntries: new	736	
\glstableSubNameFmt: new	737	
\glstableSubNameNoDesc: new	739	
\glstableSubNameSingleFmt:		
new	738	
\glstableSubNameTarget: new	737	
\glstableSubOtherNoDesc: new	741	
\glstableSubSymbolFmt: new	740	
\glstableSubSymbolNameFmt:		
new	740	
\glstableSubSymbolNameTarget:		
new	740	
\glstableSymbolColAlign: new	737	
\glstableSymbolFmt: new	739	
\glstableSymbolHeader: new	735	
\glstableSymbolNameFmt: new	740	
\glstableSymbolNameTarget:		
new	740	
\glstableSymbolWidth: new	754	
\glstableTotalCols: new	735	
\glstexpdfstring: new	31	
\glstopicSubGroupHeading:		
new	730	
\glstreesubgroupitem: new	662	
\glstreeSubPreHeader: new	657	
\glssupercase: new	31	
\glswrglossdisableanchorcmds:		
new	36	
\glxspabbrvfont: new	321	
\glxplongfont: new	321	
\glxtr@check@complexstyle:		
new	341	
\glxtr@do@ifcomplexstyle@allcaps:		
new	341	
\glxtr@do@ifcomplexstyle@insert:		
new	341	
\glxtr@do@select@nameref@record:		
new	595	
\glxtr@doifexists: new	62	
\glxtr@doifnoexists: new	63	
\GLSxtr@fullformat@fallback:		
new	319	
\GLSxtr@fullplformat@fallback:		
new	319	
\glxtr@mgls@inner: initialise		
hooks	395	
\glxtr@newabbreviation:		
added \glxtrorgkeylist	313	
bug fix: markwords doesn't		
include plural suffix	314	
\glxtr@processunknownoptions:		
new	30	
\glxtr@save@mfu: new	198	
\glxtr@select@entry: new	595	
\glxtr@select@entry@nameref:		
new	595	
\glxtr@shortfieldname: new	94	
\glxtr@title@field: new	350	
\glxtr@wrglossary@encap:		
new	138	
\glxtr@writefields: encoding		
test replaced \ifdef with		
\ifdefvoid and reversed		
args	199	
removed test for fontspec	200	
\glxtr@actualanchor: new	588	
\glxtrAddCounterRecordHook:		
new	201	
\glxtraddgroup: new	210	
\glxtraddunusedxrefs: new	75	
\glxtraliashook: new	75	
\glxtrassignactualsetup:		
added \glstextup	260	
\glxtrassignlinktextfmt:		
new	100	
\glxtrattrentrytextfmt: new	88	
\glxtrbookindexformatsubheader:		
new	692	

\glstrbookindexpostgroupskip:		\glstrGeneralLatinHtoMrules:	
new	691	new	622
\glstrbookindexpostsubgroupskip:		\glstrGeneralLatinNtoSrules:	
new	691	new	623
\glstrbookindexpregroupskip:		\glstrGeneralLatinNtoZrules:	
new	691	new	622
\glstrbookindexpresubgroupskip:		\glstrGeneralLatinTtoZrules:	
new	691	new	623
\glstrbookindexsubbookmark:		\glstrgeneralpuncaccentsrules:	
new	692	new	614
\GlsXtrClearUnsetBuffer: new	155	\glstrgeneralpuncbracketrules:	
\glstrcopytoglossary: added		new	615
starred form	62	\glstrgeneralpuncmarksrules:	
\glstrcurrentfield: new ...	94	new	614
\glstrdefaultentrytextfmt:		\glstrgeneralpuncquoterules:	
new	88	new	615
\glstrdefaultrevert: new ..	322	\glstrgeneralpuncsignrules:	
\glstrdiscardperiodretainfirstuse:		new	615
new	309	\glstrglossentryother: use	
\glstrdoidentify: new	215	default header if first	
\glstrdopostpunc: made		argument empty	203
robust	311	\GLSxtrheadfirst: new	356
\GlsXtrDualBackLink: corrected		\GLSxtrheadfirstplural: new ..	357
false part	585	\GLSxtrheadfull: new	361
\Glsxtrentryfmt: new	42	\GLSxtrheadfullpl: new	362
\Glsxtrfmt: new	41	\GLSxtrheadlong: new	359
\GLSxtrfullformat: new	318	\GLSxtrheadlongpl: new	360
\Glsxtrfullformat: added check		\GLSxtrheadname: new	353
for insert inside and inner		\GLSxtrheadplural: new	355
fmt	317	\GLSxtrheadshort: new	351
\glstrfullformat: added check		\GLSxtrheadshortpl: new	352
for insert inside and inner		\GLSxtrheadtext: new	354
fmt	317	\GLSxtrhiername: added	
\GLSxtrfullplformat: new ...	319	\expandafters	68
\Glsxtrfullplformat: added		added \glstexorpdfstring ..	68
check for insert inside and		\GLSxtrhiername: added	
inner fmt	318	\expandafters	68
\glstrfullplformat: added		added \glstexorpdfstring ..	68
check for insert inside and		\GlsXtrhiername: added	
inner fmt	318	\expandafters	67
\glstrfullsaveinsert: new ..	107	added \glstexorpdfstring ..	67
\glstrfullsep: added inner		\Glsxtrhiername: added	
fmt	319	\expandafters	67
\glstrgenentrytextfmt: new ..	88	added \glstexorpdfstring ..	67
\glstrGeneralInitRules: new	610	\glstrhiername: added	
\glstrGeneralLatinAtoGrules:		\expandafters	66
new	622	added \glstexorpdfstring ..	66
\glstrGeneralLatinAtoMrules:		\glstridentifyglfamily:	
new	621	new	216
		\glstridentifyglslink: new ..	216

\glxtrifallcaps: new	94	\GlsXtrStandaloneEntryHeadOther:	
\glxtrifheaduc: new	150	new	203
\glxtrifintoc: new	150	\GlsXtrStandaloneEntryPdfName:	
\glxtrifwasglslike: new	94	new	202
\glxtrifwasglslikeandfirstuse:		\GlsXtrStandaloneEntryPdfOther:	
new	94	new	203
\glxtrifwassubsequentorshort:		\GLSxtrsubsequentfmt: new	338
new	94	\GLSxtrsubsequentplfmt: new	339
\glxtrifwassubsequentuse:		\glxtrtaggedlist: new	70
new	94	\glxtrtaggedlistsep: new	70
\glxtrIgnoreableRules: new	610	\glxtrtitlednamerefink:	
\GLSxtrinlinfullformat: new	319	new	588
\GLSxtrinlinfullplformat:		\glxtrtitleopts: new	350
new	320	\glxtrundefdebug: new	28
\GlsXtrLetField: corrected		\GlsXtrUnsetBufferDisableRepeatLocal:	
spelling	50	new	155
\GlsXtrLong: now simulates first		\GlsXtrUnsetBufferEnableRepeatLocal:	
use	329	new	155
\glxtrMFUsave: new	198	\GLSxtrusefield: added	
\GlsXtrMglsOrGls: removed		uppercase PDF bookmark	
spurious \PLUS	413	alternative	49
\glxtrnewglsdisp: new	218	\Glsxtrusefield: now using	
\glxtrnewglslink: new	217	\@Glsentryfield	49
\glxtrnoidxgroups: new	187	\glxtrwordsephyphen: new	312
\Glsxtrpdfentryfmt: new	43	\glxtrwrglossarylocfmt: new	588
\glxtrpostlinkAddDescOnFirstUse:		\glxtrwrglosscountermark:	
added inner formatting	308	new	28
\glxtrpostlinkAddSymbolOnFirstUse:		\ifglsresetcurrcount: new	157
added inner formatting	308	\ifGlsXtrPrefixLabelFallbackLast:	
\glxtrpostlinkSymbolDescSep:		new	596
new	309	\IfTeXParserLib: new	584
\glxtrpreglossarystyle: new	83	long-desc-name: added	
\GlsXtrResetLocalBuffer: new	155	\glssubgroupheading	707
\glxtrrevert: new	322	long-desc-sym: new	724
\glxtrsaveinsert: new	107	long-desc-sym-name: added	
\glxtrseelists: new	65	\glssubgroupheading	720
\glxtrseelistsdelim: new	64	long-loc-desc-name: added	
\glxtrseelistsencap: new	64	\glssubgroupheading	709
\glxtrsetbibglsaux: new	15	long-loc-desc-sym-name: added	
\glxtrsetcomplexstyle: new	340	\glssubgroupheading	722
\glxtrsetlongfirstuse: new	328	long-loc-sym-desc-name: added	
\GlsXtrSetPlusModifier: new	141	\glssubgroupheading	719
\GlsXtrSetStarModifier: new	141	long-name-desc: added	
\glxtrshowtargetsymbolleft:		\glssubgroupheading	704
added check for math mode	33	long-name-desc-loc: added	
\glxtrshowtargetsymbolright:		\glssubgroupheading	706
added check for math mode	33	long-name-desc-sym: added	
\GlsXtrStandaloneEntryHeadName:		\glssubgroupheading	710
new	202	long-name-desc-sym-loc: added	
		\glssubgroupheading	712

long-name-sym-desc: added		\setupglslink: new	97
\glssubgroupheading	714	\shortcut@GLS: new	20
long-name-sym-desc-loc: added		\shortcut@Gls: new	20
\glssubgroupheading	715	\shortcut@gls: new	20
long-sym-desc: new	722	\shortcut@GLSpl: new	20
long-sym-desc-name: added		\shortcut@Glspl: new	20
\glssubgroupheading	717	\shortcut@glspl: new	20
\newdgl\$field: new	604	shortcuts: abother	24
\newdgl\$fieldlike: new	604	acother	24
\PGLSfmtlong: new	373	table: new	761
\Pglsfmtlong: new	373	topic: added	
\pglsfmtlong: new	373	\glssubgroupheading	729
\PGLSfmtlongpl: new	374	topicmccls: added	
\Pglsfmtlongpl: new	374	\glssubgroupheading	734
\pglsfmtlongpl: new	373	1.49 – ?	
\PGLSfmtshort: new	372	\GLSxtrIfInGlossary: new	39
\Pglsfmtshort: new	372	1.49 – ??	
\pglsfmtshort: new	372	General: added glossaries-extra-	
\PGLSfmtshortpl: new	373	abbrstyles.def	423
\Pglsfmtshortpl: new	372	\glxxtremrevert: new	499
\pglsfmtshortpl: new	372	\glxstrfootnotelongformat:	
\PGLSPrefix: new	368	new	438
\Pglsprefix: new	368	\glxstrfootnotelongplformat:	
\pglsprefix: new	368	new	438
\PGLSxtrlong: new	370	\GLSxtrlongformat: new	425
\Pglsxtrlong: new	370	\Glsxtrlongformat: new	424
\pglsxtrlong: new	370	\glxstrlongformat: new	423
\PGLSxtrlongpl: new	371	\GLSxtrlongformatgrp: new	428
\Pglsxtrlongpl: new	371	\Glsxtrlongformatgrp: new	427
\pglsxtrlongpl: new	371	\glxstrlongformatgrp: new	426
\PGLSxtrshort: new	369	\GLSxtrlonghyphennoshort:	
\Pglsxtrshort: new	369	new	547
\pglsxtrshort: new	368	\glxstrlonghyphennoshortdescsort:	
\PGLSxtrshortpl: new	370	new	547
\Pglsxtrshortpl: new	369	\glxstrlonghyphennoshortsort:	
\pglsxtrshortpl: new	369	new	554
\Pglsxtrtitlelong: new	373	\GLSxtrlonghyphenshort: new	539
\Pglsxtrtitlelongpl: new	374	\glxstrlonghyphenshortsort:	
\Pglsxtrtitleshort: new	372	new	540
\Pglsxtrtitleshortpl: new	373	\GLSxtrlongplformat: new	426
\printunsrtglossaryunitpostskip:		\Glsxtrlongplformat: new	425
new	211	\glxstrlongplformat: new	424
\printunsrttable: new	756	\GLSxtrlongplformatgrp: new	428
\prohibit@glxstrnoidxgroups:		\Glsxtrlongplformatgrp: new	428
new	187	\glxstrlongplformatgrp: new	427
\renewabbreviationstyle: reset		\GLSxtrlongshortformat: new	436
subsequent fmts	344	\Glsxtrlongshortformat: new	435
\s@glxstrcopytoglossary: new	62	\glxstrlongshortformat: new	435
\s@GLSxtrfmt: new	42	\GLSxtrlongshortplformat:	
\setupglsadd: new	97	new	436

\GLSxtrlongshortplformat:		\GLSxtrshortplformatgrp: new	434
new	435	\Glsxtrshortplformatgrp: new	433
\glsxtrlongshortplformat:		\glsxtrshortplformatgrp: new	432
new	435	\glsxtrsmrevert: new	481
\glsxtrpostabbrvfootnote:		\glsxtruserfieldfmt: new ...	525
new	448	\GLSxtruserlongformat: new .	439
\glsxtrpostfootnotelongformat:		\glsxtruserlongformat: new .	439
new	438	\GLSxtruserlongplformat: new	440
\GLSxtrpostthyphenlong: new .	570	\glsxtruserlongplformat: new	439
\GLSxtrpostthyphenlongpl: new	571	\GLSxtruserlongshortformat:	
\glsxtrpostthyphenlongpl: new	570	new	441
\GLSxtrpostthyphenshort: new .	555	\Glsxtruserlongshortformat:	
\GLSxtrpostthyphenshortpl:		new	440
new	556	\glsxtruserlongshortformat:	
\glsxtrpostthyphenshortpl:		new	440
new	555	\GLSxtruserlongshortplformat:	
\GLSxtrpostthyphensubsequent:		new	441
new	557	\Glsxtruserlongshortplformat:	
\glsxtrpostuserlongformat:		new	441
new	439	\glsxtruserlongshortplformat:	
\glsxtrpostusersshortformat:		new	440
new	438	\GLSxtruserparen: new	525
\glsxtrsconlyrevert: new ...	580	\glsxtruserparensep: new ...	525
\glsxtrscerevert: new	463	\GLSxtrusersshortformat: new .	439
\glsxtrscuserrevert: new ...	529	\glsxtrusersshortformat: new .	438
\GLSxtrshortformat: new	431	\GLSxtrusersshortlongformat:	
\Glsxtrshortformat: new	430	new	442
\glsxtrshortformat: new	429	\Glsxtrusersshortlongformat:	
\GLSxtrshortformatgrp: new .	434	new	442
\Glsxtrshortformatgrp: new .	433	\glsxtrusersshortlongformat:	
\glsxtrshortformatgrp: new .	432	new	441
\GLSxtrshorthyphenlong: new .	563	\GLSxtrusersshortlongplformat:	
\glsxtrshorthyphenlongsort:		new	442
new	564	\Glsxtrusersshortlongplformat:	
\GLSxtrshorthyphenlong:		new	442
new	540	\glsxtrusersshortlongplformat:	
\glsxtrshorthyphenlong:		new	441
new	539	\GLSxtrusersshortplformat:	
\GLSxtrshortlongformat: new .	437	new	439
\Glsxtrshortlongformat: new .	437	\glsxtrusersshortplformat:	
\glsxtrshortlongformat: new .	436	new	438
\GLSxtrshortlongplformat:		postfootnote: removed redef of	
new	437	\glsxtrsetupfulldefs ...	452
\Glsxtrshortlongplformat:		short-em-postfootnote:	
new	437	removed redef of	
\glsxtrshortlongplformat:		\glsxtrsetupfulldefs ...	522
new	436	short-em-postfootnote-desc:	
\GLSxtrshortplformat: new ..	431	removed redef of	
\Glsxtrshortplformat: new ..	430	\glsxtrsetupfulldefs ...	524
\glsxtrshortplformat: new ..	429		

short-postfootnote-desc:	\glstrsetupfulldefs ...	496
removed redef of		
\glstrsetupfulldefs ...		454
short-sc-postfootnote:	\glstrsetupfulldefs ...	498
removed redef of		
\glstrsetupfulldefs ...		479
short-sc-postfootnote-desc:	\xpglstrpostabbrvfootnote:	
removed redef of	new	448
\glstrsetupfulldefs ...	\xpglstrposthyphenlong: new	571
short-sm-postfootnote:	\xpglstrposthyphenshort:	
removed redef of	new	556
	\xpglstrposthyphensequent:	
	new	557