The HEP-ACRONYM package*

Acronyms based on glossaries-extra with sentence-aware capitalisation and first-use links

Jan Hajer†

2026/11/13

Abstract

The HEP-ACRONYM package provides sentence-aware acronym helpers based on GLOSSARIES-EXTRA. It defines declaration commands for regular, always-short, always-long, and alternative long-form acronyms, optional links back to the first use, reset and dummy-use helpers, and support for several acronym groups.

Acronyms are implemented using the GLOSSARIES-EXTRA package [1] which is an extension of the GLOSSARIES package [2] and must be loaded after the HYPERREF package [3]. It can be loaded using `\usepackage{hep-acronym}`.

`hyper` The `hyper` option controls whether later acronym uses link back to their first appearance. It is enabled by default and can be deactivated with `hyper=false`.

`titlesec` The TITLESEC package [4] is loaded so that acronyms at the beginning of section titles are capitalised. Classes with their own sectioning code, such as AMSART and PoS, are incompatible with TITLESEC; `titlesec=false` does not load it. This is the default for the KOMA-Script classes. Acronyms at the beginning of numbered section titles are then still capitalised. The HEP-PAPER package sets this option in its `amsart` and `pos` compatibility modes.

`\acronym` The `\acronym`⟨*⟩`[`⟨*typeset abbreviation*⟩`]{`⟨*abbreviation*⟩`}`⟨*⟩`{`⟨*definition*⟩`}[`⟨*plural definition*⟩`]` macro generates the singular `\`⟨*abbreviation*⟩ and plural `\`⟨*abbreviation*⟩`s` macros.

The first star prevents the addition of an 's' to the abbreviation plural. The second star restores the TEX default of swallowing subsequent white space. The long form is only shown at the first appearance of these macros, later appearances generate the abbreviation with a hyperlink to the long form. Capitalisation at the beginning of paragraphs and sentences is (mostly) ensured. The long form is never used in math mode which can be exploited to enforce the short form. In order to enforce the long form use `\`⟨*abbreviation*⟩`long`. As there can be no fixed rule whether to use the long form or the short form in section headers the user is left to their own devices e.g.

`\sentence` `\glsdesc{`⟨*abbreviation*⟩`}`, `\Glsdesc{`⟨*abbreviation*⟩`}`. The `\sentence` macro ensures that the directly following abbreviation is capitalised.

In portable document format (PDF) strings, such as bookmarks and the document information, the acronym macros expand to their short form, or to their long form for the `long` variants. An acronym macro at the beginning of a PDF string or directly after `\sentence` is capitalised. Optional arguments of the acronym macros are not supported in PDF strings, and `\textsc` in a typeset abbreviation becomes upper case. In the table of contents and the lists of figures and tables, acronyms are shown in their short form without links, and they change neither the first

*This document corresponds to HEP-ACRONYM v1.6.

†jan.hajer@tecnico.ulisboa.pt

use nor the entry count. Acronyms in the text of links, e.g. of `\href` or `\hyperlink`, do not link to their first use either.

`\shortacronym` `\longacronym` The `\shortacronym` and `\longacronym` macros are drop-in replacements of the `\acronym` macro showing only the short or long form of their acronym.

`\acronymalternative` The `\acronymalternative{`⟨*acronym*⟩`}{`⟨*name*⟩`}`⟨***⟩`{`⟨*alternative long form*⟩`}` macro defines an additional acronym macro that shares the use counter and short form of ⟨*acronym*⟩ but prints the supplied long form at first use.

`\resetacronym` The first use form of the acronym can be enforced by resetting the acronym counter with `\resetacronym{`⟨*key*⟩`}`.

`\dummyacronym` If the acronym counter equals one at the end of the document the short form of the acronym is not introduced. Placing a `\dummyacronym{`⟨*key*⟩`}` at the end of the document ensures that the short form is introduced.

`\acronyms` The `\acronyms[`⟨*key*⟩`]{`⟨*title*⟩`}` macro creates a new acronym glossary and switches subsequent declarations to it. Without the optional argument the mandatory argument is used both as glossary key and printed title.

`warning` In order to reduce the number of potentially conflicting packages the GLOSSARIES package is loaded without any glossary style. In the case that the glossary should be printed additional packages must be loaded via e.g. `\usepackage{glossary-list}`.

A Implementation

<*package>

Define a hepacronym namespace for the options using the KVOPTIONS package [5].

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{
  family=hepacronym,
  prefix=hepacronym@
}
```

`hyper` Define the `hyper` option controlling the hyperlink to the first appearance of the acronyms.

```
\DeclareBoolOption[true]{hyper}
```

`titlesec` Define the `titlesec` option controlling whether section titles are treated as sentences. The KOMA-SCRIPT classes, which define `\KOMAClassName`, have their own sectioning code, which TITLESEC does not support. For them the option is false unless it is set explicitly.

```
\@ifundefined{KOMAClassName}{%
  \DeclareBoolOption[true]{titlesec}%
}{%
  \DeclareBoolOption[false]{titlesec}%
}
```

Process options.

```
\ProcessKeyvalOptions*
```

Load the base package [1, 2] and set the abbreviation style.

```
\PassOptionsToPackage{nostyles}{glossaries-extra}
\RequirePackage{glossaries-extra}
% \setabbreviationstyle{long-hyphen-short-hyphen}
```

Use the entry count feature.

```
\glsenableentrycount
\glssetcategoryattribute{abbreviation}{entrycount}{1}
```

Provide macros for older `glossaries-extra` installations.

```
\AtBeginDocument{
  \@ifpackageloaded{hyperref}{
    \providecommand*{\glsxtrusefield}[2]{\@gls@entry@field{#1}{#2}}
    \providecommand*{\glsxtrsetfieldifexists}[3]{\glsdoifexists{#1}{#3}}
    \providecommand*{\gGlsXtrSetField}[3]{%
      \glsxtrsetfieldifexists{#1}{#2}{%
        \csgdef{glo@\glsdetoklabel{#1}@#2}{#3}%
      }%
    }
```

Set hyperlinks from the abbreviation to their definition in the text.

```
    \glssetcategoryattribute{abbreviation}{nohyperfirst}{true}
    \renewcommand*{\glsdonohyperlink}[2]{{%
      \glsxtrprotectlinks\edef\fieldvalue{%
        \glsxtrusefield{\glslabel}{hastarget}%
      }%
      \ifdefstring\fieldvalue{true}{#2}{%
        \gGlsXtrSetField{\glslabel}{hastarget}{true}%
        \glsdohypertarget{#1}{#2}%
      }%
    }}
  }{}
}
```

`\hep@disable` `\hep@pdf@definitions` Collect the PDF-string definitions of the acronym macros in a macro that HYPERREF executes before it converts a string. This works independently of the order in which the packages are loaded and also for acronyms declared in the document body.

```
\newcommand*{\hep@pdf@definitions}{}
\newcommand*{\hep@disable}[1]{\g@addto@macro\hep@pdf@definitions{#1}}
```

`\hep@pdf@define` Define the PDF string `{`⟨*text*⟩`}{`⟨*space*⟩`}` of the acronym macro with name `{`⟨*name*⟩`}`. The ⟨*space*⟩ is either empty or `\xspace`, which HYPERREF supports in PDF strings. The capitalised variant is stored under the detokenised macro name and used at the beginning of a PDF string. Plain text has no small capitals, so `\textsc` becomes upper case.

```
\newcommand*{\hep@pdf@define}[3]{%
  \def\hep@pdf@text{#2}%
  \hep@pdf@uppercase@sc
```

```
  \begingroup
    \edef\hep@temp{\endgroup
      \noexpand\hep@disable{%
        \noexpand\@namedef{#1}{\unexpanded\expandafter{\hep@pdf@text#3}}%
      }%
      \global\noexpand\@namedef{hep@pdf@cap@\@backslashchar#1 }{%
        \noexpand\MFUsentencecase{\unexpanded\expandafter{\hep@pdf@text}}%
        \unexpanded{#3}%
      }%
    }%
  \hep@temp
}
\ExplSyntaxOn
\cs_new_protected:Npn \hep@pdf@uppercase@sc
  {
    \regex_replace_all:nnN
      { \c{textsc} } { \c{text_uppercase:n} } \hep@pdf@text
  }
\ExplSyntaxOff
```

`\hep@pdf@sentence` In PDF strings, `\sentence` capitalises the following acronym macro. Other expandable tokens in front of it, such as the conditionals that TITLESEC puts in front of the section titles, `\numberline` or `\textbf`, are expanded first. Any other token ends the sentence beginning, and so does `\@empty`, which `\pdfstringdef` places at the end of every string.

```
\ExplSyntaxOn
\cs_new:Npn \hep@pdf@sentence #1
  {
    \cs_if_exist:cTF { hep@pdf@cap@ \tl_to_str:n {#1} }
      { \use:c { hep@pdf@cap@ \tl_to_str:n {#1} } }
      {
        \bool_lazy_all:nTF
          {
            { \tl_if_single_token_p:n {#1} }
            { \token_if_expandable_p:N #1 }
            { ! \token_if_eq_meaning_p:NN #1 \@empty }
          }
          { \exp_after:wN \hep@pdf@sentence #1 }
          {#1}
      }
  }
\ExplSyntaxOff
```

`\hep@pdf@numberline` In the standard classes, the bookmarks of sectioning commands begin with `\protect` followed by `\numberline{`⟨*number*⟩`}`. Skip `\protect`, which is not expandable here, and treat the text after the number as the beginning of a sentence.

```
\@namedef{hep@pdf@cap@\@backslashchar protect }{\sentence}
\newcommand*{\hep@pdf@numberline}[1]{\hep@pdf@@numberline{#1}\sentence}
\hep@disable{%
  \let\hep@pdf@@numberline\numberline
```

```
  \let\numberline\hep@pdf@numberline
}
```

`\pdfstringdef` Once HYPERREF is loaded, register the PDF-string definitions and treat every PDF string, e.g. a bookmark, as the beginning of a sentence. The `\@empty` ensures that `\sentence` has an argument.

```
\newcommand*{\hep@pdf@register}{%
  \@ifpackageloaded{hyperref}{%
    \pdfstringdefDisableCommands{\hep@pdf@definitions}%
    \global\let\hep@pdfstringdef\pdfstringdef
    \gdef\pdfstringdef##1##2{\hep@pdfstringdef##1{\sentence##2\@empty}}%
    \global\let\hep@pdf@register\relax
  }{}%
}
\hep@pdf@register
\AtBeginDocument{\hep@pdf@register}
```

`\hyper@link` `\hyper@linkurl` `\hyper@linkfile` Acronyms in the text of a link, e.g. of `\href` or `\hyperlink`, do not link to their first use, since links cannot be nested. The drivers of HYPERREF define these macros, so they are wrapped at the end of `\begin{document}`.

```
\AddToHook{begindocument/end}{%
  \@ifpackageloaded{hyperref}{%
    \let\hep@hyper@link\hyper@link
    \def\hyper@link#1#2#3{\hep@hyper@link{#1}{#2}{{\glsdisablehyper#3}}}%
    \let\hep@hyper@linkurl\hyper@linkurl
    \def\hyper@linkurl#1#2{\hep@hyper@linkurl{{\glsdisablehyper#1}}{#2}}%
    \let\hep@hyper@linkfile\hyper@linkfile
    \def\hyper@linkfile#1#2#3{%
      \hep@hyper@linkfile{{\glsdisablehyper#1}}{#2}{#3}%
    }%
  }{}%
}
```

Ignore only hyperref's `math shift` PDF-string warning.

```
\AtBeginDocument{%
  \@ifpackageloaded{hyperref}{%
    \let\hep@HyPsd@CatcodeWarning\HyPsd@CatcodeWarning
    \renewcommand*{\HyPsd@CatcodeWarning}[1]{%
      \ifstrequal{#1}{math shift}{}{%
        \hep@HyPsd@CatcodeWarning{#1}%
      }%
    }%
  }{}%
}
```

Deactivate hyperrefs if requested.

```
\ifhepacronym@hyper\else
  \glsdisablehyper
\fi
```

Load the TITLESEC package [4] in order to ensure that section headers are treated as sentences.

```
\ifhepacronym@titlesec\RequirePackage{titlesec}\fi
```

`\begin@sentence` Mark the beginning of a paragraph as if it would follow a full stop using the EVERYHOOK package [6]. The space factor is a `\mathchardef` constant like `\@m`, so that digits following `\sentence`, for example in the label `\item[12]`, do not extend the number.

```
\PassOptionsToPackage{excludeor}{everyhook}
\RequirePackage{everyhook}
\@ifdefinable\begin@sentence{\mathchardef\begin@sentence=1001\relax}
\newcommand*{\sentence}{\ifhmode\spacefactor=\begin@sentence\fi}
\hep@disable{\let\sentence\hep@pdf@sentence}
\PushPostHook{par}{{\sentence}}
```

`\ttl@case` `\@hangfrom` Treat section titles as sentences using the ETOOLBOX package [7]. With TITLESEC its case hook `\ttl@case` marks the beginning of the title. Without it, the beginning of a sentence is marked after `\@hangfrom`, which sets the number of displayed section headings in the standard classes, AMSART, and PoS. Starred sections in the latter two remain lower case if HYPERREF is loaded, as its anchor restores the previous space factor.

```
\ifhepacronym@titlesec
  \gdef\ttl@case{\leavevmode\sentence}
\else
  \RequirePackage{etoolbox}
  \apptocmd{\@hangfrom}{\sentence}{}{}
\fi
```

`\frenchspacing` Adjust the `\frenchspacing` macro to be compatibel with this idea.

```
\def\frenchspacing{%
  \sfcode`\.\begin@sentence \sfcode`\?\begin@sentence
  \sfcode`\!\begin@sentence \sfcode`\:\begin@sentence
  \sfcode`\;\@m \sfcode`\,\@m
}
```

`if@begin@of@sentence` Provide a macro checking for the beginning of a sentence by examining the length of the preceeding space.

```
\newcommand*{\hep@if@begin@of@sentence}[2]{\leavevmode\protecting{%
  \ifboolexpr{ test {\ifnumcomp{\spacefactor}{=}{3000}} or%
               test {\ifnumcomp{\spacefactor}{=}{2000}} or%
               test {\ifnumcomp{\spacefactor}{=}{\begin@sentence}}%
  }{#1}{#2}%
}}
```

`\@starttoc` Typeset the table of contents and the lists of figures and tables, which all use `\@starttoc`, with the short form of the acronyms. Their entries begin sentences. Acronyms are marked as used only locally, so that the lists change neither the first use nor the entry count. Links and link targets are deactivated, since the entries are links themselves.

```
\let\hep@starttoc\@starttoc
```

```
\renewcommand*{\@starttoc}[1]{%
  \begingroup
    \let\@glsunset\@glslocalunset
    \let\@glsreset\@glslocalreset
    \let\@gls@increment@currcount\@gls@local@increment@currcount
    \glslocalunsetall
    \glsdisablehyper
    \let\@glslink\@secondoftwo
    \let\glsdonohyperlink\@secondoftwo
    \let\hep@contents@line\contentsline
    \let\hep@number@line\numberline
    \renewcommand*{\contentsline}[2]{%
      \hep@contents@line{##1}{\sentence ##2}%
    }%
    \renewcommand*{\numberline}[1]{\hep@number@line{##1}\sentence}%
    \hep@starttoc{#1}%
  \endgroup
}
```

`\hep@item@sentence` `hep@description@item` `\descriptionsitem` Ensure that acronyms in description items get capitalised using the ENUMITEM package [8]. The sentence-aware `\item` is installed only if it is not active yet, because in nested description lists saving the active `\item` would make it call itself.

```
\RequirePackage{enumitem}
\newcommand*\hep@item@sentence{\leavevmode\sentence}
\NewDocumentCommand{\hep@description@item}{o}{%
  \IfNoValueTF{#1}{%
    \hep@latex@item\hep@item@sentence
  }{%
    \hep@latex@item[\hep@item@sentence#1]\hep@item@sentence
  }%
}
\newcommand*\descriptionsitem{%
  \ifx\item\hep@description@item\else
    \let\hep@latex@item\item
    \let\item\hep@description@item
  \fi
}
\setlist[description]{before=\descriptionsitem}
```

`\mathdef` Provide a macro that allows to extend a macro definition into math mode using the XPARSE package [9].

```
\RequirePackage{xparse}
\ProvideDocumentCommand{\mathdef}{mO{0}m}{%
  \expandafter\let\csname hep@text\string#1\endcsname=#1
  \expandafter\newcommand\expandafter*%
    \csname hep@math\string#1\endcsname[#2]{#3}
  \DeclareRobustCommand#1{%
    \ifmmode
      \expandafter\let\expandafter\next\csname%
```

```
      hep@math\string#1\endcsname%
    \else
      \expandafter\let\expandafter\next\csname%
      hep@text\string#1\endcsname%
    \fi
    \next
  }%
}
```

`\hep@if@short` Check whether `\cgls{`⟨*label*⟩`}` prints the short form. It does not if the entry is unused or if the entry count of the previous run is one.

```
\newcommand*{\hep@if@short}[1]{%
  \ifglsused{#1}{%
    \ifnum\glsentryprevcount{#1}=\@ne
      \expandafter\@secondoftwo
    \else
      \expandafter\@firstoftwo
    \fi
  }{\@secondoftwo}%
}
```

`\acronym` Define the `\acronym`⟨*⟩`[`⟨*typeset abbreviation*⟩`]{`⟨*abbreviation*⟩`}`⟨*⟩`{`⟨*definition*⟩`}[`⟨*plural definition*⟩`]` macro using the xspace [10] and amstext [11] packages.

- #1 star for omitting the 's' in the short plural
- #2 optional typeset abbreviation
- #3 mandatory abbreviation
- #4 star for restoring the TeX default for space after text macros
- #5 mandatory long form
- #6 optional plural long form

```
\RequirePackage{xspace}
\RequirePackage{amstext}
\NewDocumentCommand{\acronym}{somsmo}{
  \newabbreviation[
    type=\acronymtype,
    sort=#3,
    \glsshortpluralkey=\IfBooleanTF{#1}{#3}{\IfNoValueTF{#2}{#3s}{#2s}},
    longplural=\IfNoValueTF{#6}{#5s}{#6}
  ]{#3}{\IfNoValueTF{#2}{#3}{#2}}{#5}
```

Provide the singular acronym macro.

```
  \expandafter\newcommand\expandafter*\csname#3\endcsname[1][]{%
    \hep@if@begin@of@sentence{%
      \hep@if@short{#3}{\cgls{#3}[##1]}{\cGls{#3}[##1]}%
    }{\cgls{#3}[##1]}%
    \ifnum\glsentrycurrcount{#3}>1\relax
      \IfBooleanTF{#4}{}{\@\xspace}%
    \else\@\xspace\fi
  }
```

Expand the singular acronym macro in PDF labels.

```
\hep@pdf@define{#3}{\IfNoValueTF{#2}{#3}{#2}}{\IfBooleanF{#4}\xspace}%
```

Provide the singular acronym macro in math mode.

```
\expandafter\mathdef\csname#3\endcsname{%
  \mathinner{\text{\glsxtrshort{#3}}}\@gls@increment@currcount{#3}%
}
```

Provide the plural acronym macro.

```
\expandafter\newcommand\expandafter*\csname#3s\endcsname[1][]{%
  \hep@if@begin@of@sentence{\cGlspl{#3}[##1]}{\cglspl{#3}[##1]}%
  \IfBooleanTF{#4}{}{\@\xspace}%
}
```

Expand the plural acronym macro in PDF labels.

```
\hep@pdf@define{#3s}%
  {\IfBooleanTF{#1}{#3}{\IfNoValueTF{#2}{#3s}{#2s}}}%
  {\IfBooleanF{#4}\xspace}%
```

Provide the plural acronym macro in math mode.

```
\expandafter\mathdef\csname#3s\endcsname{%
  \mathinner{\text{\glsxtrshortpl{#3}}}\@gls@increment@currcount{#3}%
}
```

Provide an enforced long form.

```
\expandafter\newcommand\expandafter*\csname#3long\endcsname[1][]{%
  \hep@if@begin@of@sentence{\Glsdesc*{#3}[##1]}{\glsdesc*{#3}[##1]}%
  \IfBooleanTF{#4}{}{\@\xspace}%
}
```

Ensure that it works in PDFs.

```
\hep@pdf@define{#3long}{#5}{\IfBooleanF{#4}\xspace}%
```

Provide an enforced long plural form.

```
\expandafter\newcommand\expandafter*\csname#3slong\endcsname[1][]{%
  \hep@if@begin@of@sentence{%
    \Glsxtrlongpl*{#3}[##1]}{\glsxtrlongpl*{#3}[##1]%
  }%
  \IfBooleanTF{#4}{}{\@\xspace}%
}
```

Ensure that it works in PDFs.

```
\hep@pdf@define{#3slong}{\IfNoValueTF{#6}{#5s}{#6}}%
  {\IfBooleanF{#4}\xspace}%
```

Provide an enforced short form.

```
\expandafter\newcommand\expandafter*\csname#3short\endcsname[1][]{%
  \glsxtrshort{#3}\@gls@increment@currcount{#3}%
  \IfBooleanTF{#4}{}{\@\xspace}%
}
\hep@pdf@define{#3short}{\IfNoValueTF{#2}{#3}{#2}}%
  {\IfBooleanF{#4}\xspace}%
```

Provide an enforced first form.

```
\expandafter\newcommand\expandafter*\csname#3first\endcsname[1][]{%
  \glsreset{#3}%
  \hep@if@begin@of@sentence{%
    \hep@if@short{#3}{\cgls{#3}[##1]}{\cGls{#3}[##1]}%
  }{\cgls{#3}[##1]}%
  \ifnum\glsentrycurrcount{#3}>1\relax
    \IfBooleanTF{#4}{}{\@\xspace}%
  \else\@\xspace\fi
}
\hep@pdf@define{#3first}{#5 (\IfNoValueTF{#2}{#3}{#2})}{\xspace}%
```

End of `\acronym`.

```
}
```

`\shortacronym` The `\shortacronym` never expands into the long form.

```
\NewDocumentCommand{\shortacronym}{somsmo}{
```

Provide the singular acronym macro.

```
\expandafter\newcommand\expandafter*\csname#3\endcsname[1][]{%
  \IfNoValueTF{#2}{#3}{#2}##1\IfBooleanTF{#4}{}{\@\xspace}%
}
```

Expand the singular acronym macro in PDF labels.

```
\hep@pdf@define{#3}{\IfNoValueTF{#2}{#3}{#2}}{\IfBooleanF{#4}\xspace}%
\hep@pdf@define{#3long}{\IfNoValueTF{#2}{#3}{#2}}%
  {\IfBooleanF{#4}\xspace}%
```

Provide the singular acronym macro in math mode.

```
\expandafter\mathdef\csname#3\endcsname{%
  \mathinner{\text{\IfNoValueTF{#2}{#3}{#2}}}%
}
```

Provide the plural acronym macro.

```
\expandafter\newcommand\expandafter*\csname#3s\endcsname[1][]{%
  \IfBooleanTF{#1}{#3}{\IfNoValueTF{#2}{#3s}{#2s}}##1%
  \IfBooleanTF{#4}{}{\@\xspace}%
}
```

Expand the plural acronym macro in PDF labels.

```
  \hep@pdf@define{#3s}%
    {\IfBooleanTF{#1}{#3}{\IfNoValueTF{#2}{#3s}{#2s}}}%
    {\IfBooleanF{#4}\xspace}%
  \hep@pdf@define{#3slong}%
    {\IfBooleanTF{#1}{#3}{\IfNoValueTF{#2}{#3s}{#2s}}}%
    {\IfBooleanF{#4}\xspace}%
```

Provide the plural acronym macro in math mode.

```
  \expandafter\mathdef\csname#3s\endcsname{%
    \mathinner{\text{\IfBooleanTF{#1}{#3}{\IfNoValueTF{#2}{#3s}{#2s}}}}%
  }%
```

Ensure that long form also exists.

```
  \expandafter\let\csname#3long\expandafter\endcsname\csname#3\endcsname
  \expandafter\let\csname#3slong\expandafter\endcsname\csname#3s\endcsname
```

End of `\shoracronym`.

```
}
```

`\longacronym` The `\longacronym` never shows the abbreviated form.

```
\NewDocumentCommand{\longacronym}{somsmo}{
```

Provide the singular acronym macro.

```
  \expandafter\newcommand\expandafter*\csname#3\endcsname[1][]{%
    \hep@if@begin@of@sentence{\MFUsentencecase{#5}}{#5}##1%
    \IfBooleanTF{#4}{}{\@\xspace}%
  }
```

Expand the singular acronym macro in PDF labels.

```
  \hep@pdf@define{#3}{#5}{\IfBooleanF{#4}\xspace}%
  \hep@pdf@define{#3long}{#5}{\IfBooleanF{#4}\xspace}%
```

Provide the plural acronym macro.

```
  \expandafter\newcommand\expandafter*\csname#3s\endcsname[1][]{%
    \hep@if@begin@of@sentence{%
      \IfNoValueTF{#6}{\MFUsentencecase{#5s}}{\MFUsentencecase{#6}}%
    }{%
      \IfNoValueTF{#6}{#5s}{#6}%
    }##1%
    \IfBooleanTF{#4}{}{\@\xspace}%
  }
```

Expand the plural acronym macro in PDF labels.

```
  \hep@pdf@define{#3s}{\IfNoValueTF{#6}{#5s}{#6}}%
    {\IfBooleanF{#4}\xspace}%
```

```
  \hep@pdf@define{#3slong}{\IfNoValueTF{#6}{#5s}{#6}}%
    {\IfBooleanF{#4}\xspace}%
```

Ensure that long form also exists.

```
  \expandafter\let\csname#3long\expandafter\endcsname\csname#3\endcsname
  \expandafter\let\csname#3slong\expandafter\endcsname\csname#3s\endcsname
```

End of `\longacronym`.

```
}
```

`\hep@alternative` Execute `{`⟨*code*⟩`}` with the long form and the description of the abbreviation `{`⟨*label*⟩`}` replaced by its `useri` field. The GLOSSARIES-EXTRA abbreviation styles read these fields directly. The original fields are restored afterwards, before any spacing code is executed.

```
\newrobustcmd*{\hep@alternative}[2]{%
  \csletcs{hep@alternative@long}{glo@\glsdetoklabel{#1}@long}%
  \csletcs{hep@alternative@desc}{glo@\glsdetoklabel{#1}@desc}%
  \csletcs{glo@\glsdetoklabel{#1}@long}{glo@\glsdetoklabel{#1}@useri}%
  \csletcs{glo@\glsdetoklabel{#1}@desc}{glo@\glsdetoklabel{#1}@useri}%
  #2%
  \csletcs{glo@\glsdetoklabel{#1}@long}{hep@alternative@long}%
  \csletcs{glo@\glsdetoklabel{#1}@desc}{hep@alternative@desc}%
}
```

`\acronymalternative` Allow for alternative long text in abbreviation.

```
\NewDocumentCommand{\acronymalternative}{mmsm}{%
```

Store it in the `useri` field.

```
  \GlsXtrSetField{#1}{useri}{#4}%
```

Define the acronym and execute the abbreviation macros with the alternative long form.

```
  \expandafter\newcommand\expandafter*\csname#2\endcsname[1][]{%
    \hep@alternative{#1}{%
      \hep@if@begin@of@sentence{%
        \hep@if@short{#1}{\cgls{#1}[##1]}{\cGls{#1}[##1]}%
      }{\cgls{#1}[##1]}%
    }%
```

Spacing code.

```
    \ifnum\glsentrycurrcount{#1}>1\relax
      \IfBooleanTF{#3}{}{\@\xspace}%
    \else\@\xspace\fi
  }
```

Redirect PDF code to original acronym.

```
  \hep@pdf@define{#2}{\glsentryshort{#1}}{\IfBooleanF{#3}\xspace}%
```

Redirect math macro to original acronym.

```
  \expandafter\mathdef\csname#2\endcsname{%
    \mathinner{\text{\glsxtrshort{#1}}}\@gls@increment@currcount{#1}%
  }
```

Provide the long version of the acronym.

```
  \expandafter\newcommand\expandafter*\csname#2long\endcsname[1][]{%
    \hep@alternative{#1}{%
      \hep@if@begin@of@sentence{\Glsdesc*{#1}[##1]}{\glsdesc*{#1}[##1]}%
    }%
    \IfBooleanTF{#3}{}{\@\xspace}%
  }
```

Provide PDF code.

```
  \hep@pdf@define{#2long}{#4}{\IfBooleanF{#3}\xspace}%
}
```

Silence warning if no acronyms are defined.

```
\renewcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout{%
    \string\providecommand*{\string\@gls@entry@count}[2]{}
  }%
  \count@=0\relax
  \forallglsentries{\@glsentry}{%
    \glshasattribute{\@glsentry}{entrycount}{%
      \ifglsused{\@glsentry}{%
        \immediate\write\@auxout{%
          \string\@gls@entry@count{\@glsentry}{%
            \glsentrycurrcount{\@glsentry}%
          }
        }%
      }{}\advance\count@ by \@ne
    }{}%
  }%
}
```

`\resetacronym` `\dummyacronym` Add two macros for acronym management.

```
\newcommand*{\resetacronym}[1]{\protect\glsreset{#1}}
\newcommand*{\dummyacronym}[1]{\protect\glsunset{#1}}
\hep@disable{\let\resetacronym\@gobble\let\dummyacronym\@gobble}
```

`abstract` Adjust the `abstract` environment to reset all acronym counters.

```
\@ifundefined{endabstract}{}{%
  \let\end@hep@abstract\endabstract%
  \renewcommand*\endabstract{\glsresetall\end@hep@abstract}%
}
```

`\acronyms` Add a possibility to have different groups of acronyms.

```
\NewDocumentCommand{\acronyms}{om}{%
  \IfNoValueTF{#1}{
    \newglossary{#2}{#2.in}{#2.out}{#2}%
    \renewcommand*{\acronymtype}{#2}%
  }{
    \newglossary{#1}{#1.in}{#1.out}{#2}%
    \renewcommand*{\acronymtype}{#1}%
  }
}
```

</package>

B Test

<*test>

```
\documentclass{article}

\usepackage{hyperref}
\usepackage{hep-acronym}
\usepackage{titlesec}

\acronym{PDF}{portable document format}
\acronym{URL}{uniform resource locator}
\acronym{CM}{computer modern}
\acronym{LM}{latin modern}

\begin{document}

\PDF \emph{(Capitalised at beginning of paragraph)}
\PDF \emph{(Second appearance is shortened)}.
\URLs \emph{(Capitalised plural form at beginning of sentence)}
$\URL$ \emph{(Second appearance is forced to be always short)}
\LM \emph{(Not capitalised in middle of sentence)}
\LM \emph{(Second appearance is shorted)}.
$\CM$ \emph{(Enforce short form)}
\CM \emph{(then extend second appearance)}
\URLlong \emph{(enforce long form)}

\section{\PDFlong table of contents is not broken}

\end{document}
```

</test>

C Readme

<*readme>

```
# The ‘hep-acronym‘ package

Acronyms based on glossaries-extra with sentence-aware capitalisation
and first-use links

## Introduction

The ‘hep-acronym‘ package provides an acronym macro based on the
‘glossaries-extra‘ package. It defines declaration commands for regular,
always-short, always-long, and alternative long-form acronyms, keeps the
first-use counters sentence-aware, optionally links later uses back to the
first occurrence, and provides helpers for resetting, dummy-using, and
grouping acronyms.

It can be loaded using ‘\usepackage{hep-acronym}‘.

## Author

Jan Hajer

## License

This file may be distributed and/or modified under the conditions of the
‘LaTeX‘ Project Public License, either version 1.3c of this license or
(at your option) any later version. The latest version of this license is
in ‘http://www.latex-project.org/lppl.txt‘ and version 1.3c or later is
part of all distributions of LaTeX version 2005/12/01 or later.
```

</readme>

References

[1] N. T. Dickimaw Books. ‘The glossaries-extra package: An extension to the glossaries package’ (2015). CTAN: glossaries-extra.

[2] N. Talbot. ‘The glossaries package: Create glossaries and lists of acronyms’ (2007). CTAN: glossaries.

[3] *LaTeX3 Project*. ‘Hypertext marks in LaTeX: a manual for hyperref: Extensive support for hypertext in LaTeX’ (1995). CTAN: hyperref. GitHub: latex3/hyperref.

[4] J. Bezos. ‘The titlesec, titleps and titletoc packages: Select alternative section titles’ (1998). CTAN: titlesec. GitHub: jbezos/titlesec.

[5] H. Oberdiek. ‘The kvoptions package: Key value format for package options’ (2004). CTAN: kvoptions. GitHub: ho-tex/kvoptions.

[6] S. Checkoway. ‘The everyhook package: Hooks for standard TeX token lists’ (2010). CTAN: everyhook.

[7] P. Lehman and J. Wright. ‘The etoolbox package: e-TeX tools for LaTeX’ (2007). CTAN: etoolbox.

[8] J. Bezos. ‘Customizing lists with the enumitem package: Control layout of itemize, enumerate, description’ (2003). CTAN: enumitem. URL: texnia.com/enumitem.html. GitHub: jbezos/enumitem.

[9] *LaTeX3 Project*. 'The `xparse` package: A generic document command parser' (1999). CTAN: `xparse`.
[10] *LaTeX Team*. 'The `xspace` package: Define commands that appear not to eat spaces' (1993). CTAN: `xspace`.
[11] *LaTeX Team*. 'The `amstext` package: Typeset text fragments in mathematics' (1995). CTAN: `amstext`. URL: `ams.org/tex/amslatex`.

Change History

v1.0
General: First release on CTAN: acronyms based on glossaries-extra with sentence-aware use, short and long variants, reset and dummy helpers, and acronym groups. 1
v1.1
General: Added `\sentence` and forced long-form acronym commands, and disabled PDF strings only when hyperref is missing. 1
v1.2
General: Added the hyper option, alternative long-form acronyms, and acronyms in math mode. 1
v1.3
General: Stopped resetting acronyms in tables and lists, and applied starred command definitions. 1
v1.4
General: Released without changes. 1
v1.5
General: Raised the LaTeX format requirement and expanded the documentation, README, and examples. 1
v1.6
General: Added the titlesec option for classes with their own sectioning code, which still capitalise acronyms at the beginning of numbered section titles, fixed an infinite loop in nested description lists, ended the line after `\glsreset` in the enforced first forms, kept `\sentence` from setting the space factor outside horizontal mode, and improved the HYPERREF support: PDF strings are defined independently of the package loading order, for all acronym variants and without stray spaces, capitalised at the beginning of bookmarks, and the contents lists neither consume the first use nor nest links, nor do the texts of links. 1