

An Acronym Environment for L^AT_EX 2_ε*

Tobias Oetiker

2024/03/30

1 Introduction

When writing a paper on cellular mobile radio I started to use a lot of acronyms. This can be very disturbing for the reader, as he might not know all the used acronyms. To help the reader I kept a list of all the acronyms at the end of my paper.

This package makes sure, that all acronyms used in the text are spelled out in full at least once.

2 The user interface

The package provides several commands and one environment for dealing with acronyms. Their appearance can be controlled by two package options and three macros.

2.1 Acronyms in the Text

`\ac` To enter an acronym inside the text, use the

```
\ac[<linebreak penalty>]{<acronym>}
```

command. The first time you use an acronym, the full name of the acronym along with the acronym in brackets will be printed. If you specify the `footnote` option while loading the package, the full name of the acronym is printed as a footnote. The next time you access the acronym only the acronym will be printed.

When an acronym is being used, for the first time (with the `footnote` option not specified), next to the end of the line, a line break between the full name of the acronym and the acronym in brackets can be encountered. The optional variable represents the penalty level of breaking the line at that place, taking integer values between 0 and 4. A higher number corresponds to a higher penalty.

`\Ac` Works in the same way as `\ac`, but starts the long form with an upper case letter. Use case: when the acronym is used for the first time, at the beginning of a sentence.

`\acresetall` The 'memory' of the macro `\ac` can be flushed by calling the macro `\acresetall`. Afterwards, `\ac` will print the full name of any acronym and the acronym in brackets the next time it is used.

- `\acf` If later in the text again the Full Name of the acronym should be printed, use the command
- `\acf[\langle linebreak penalty \rangle]{\langle acronym \rangle}`
- to access the acronym. It stands for “full acronym” and it always prints the full name and the acronym in brackets.
- When an full acronym is being used next to the end of the line, a line break between the full name of the acronym and the acronym in brackets can be encountered. The optional variable represents the penalty level of breaking the line at that place, taking integer values between 0 and 4. A higher number corresponds to a higher penalty.
- `\Acf` Works in the same way as `\acf`, but starts the long form with an upper case letter.
- `\acs` To get the short version of the acronym, use the command
- `\acs{\langle acronym \rangle}`
- `\acl` Gives you the expanded acronym without even mentioning the acronym.
- `\acl{\langle acronym \rangle}`
- `\Acl` Works in the same way as `\acl`, but starts with an upper case letter.
- `\acp` Works in the same way as `\ac`, but makes the short and/or long forms into plurals.
- `\Acp` Works in the same way as `\acp`, but starts the long form with an upper case letter.
- `\acfp` Works in the same way as `\acf`, but makes the short and long forms into plurals.
- `\Acfp` Works in the same way as `\acfp`, but starts the long form with an upper case letter.
- `\acsp` Works in the same way as `\acs`, but makes the short form into a plural.
- `\aclp` Works in the same way as `\acl`, but makes the long form into a plural.
- `\Aclp` Works in the same way as `\aclp`, but starts with an upper case letter.
- `\acfi` Works in the same way as `\acf`, but prints the Full Name acronym (`\acl`) in italics and the abbreviated form (`\acs`) in upshaped form.
- `\Acfi` Works in the same way as `\acfi`, but starts the long form with an upper case letter.
- `\acfip` Works in the same way as `\acfi`, but makes the short and long forms into plurals.
- `\Acfip` Works in the same way as `\acfip`, but starts the long form with an upper case letter.
- `\acused` Marks an acronym as used, as if it had been called with `\ac`, but without printing anything. This means that in the future only the short form of the acronym will be printed.
- `\acsu` Prints the short form of the acronym and marks it as used.
- `\aclu` Prints the long form of the acronym and marks it as used.
- `\Aclu` Works in the same way as `\aclu`, but starts with an upper case letter.
Example: `\acl{lox}/\acl{lh2}` (`\acsu{lox}/\acsu{lh2}`)
- `\iac` Works in the same way as the `\ac` command but prefixes it with an appropriate indefinite article.

`\Iac` Works in the same way as the `\ac` command but prefixes it with an appropriate upper case indefinite article.

`\...*` The following commands do the same as their unstarred forms, except that the acronym will not be marked as used. If you work with the 'onlyused' option then macros which have only been used with starred commands will not show up.
`\ac*`, `\Ac*`, `\acs*`, `\acl*`, `\Acl*`, `\acf*`, `\Acf*`, `\acp*`, `\Acp*`, `\acsp*`, `\aclp*`,
`\Aclp*`, `\acfp*`, `\Acfp*`, `\acfi*`, `\Acfi*`, `\acfip*`, `\Acfip*`, `\acsu*`, `\aclu*`,
`\Aclu*`, `\iac*`, and `\Iac*`.

2.2 Customization

The appearance of `\acs` and `\acf` can be configured in various ways. Of main importance are the package options:

`footnote` makes the full name of the acronym appear as a footnote.

`smaller` lets the acronyms appear a bit smaller than the surrounding text. This is in accord with typographic convention. The `reysize` package is required.

There are three lower-level macros controlling the output. Any acronym printed by `\acs` is formatted by `\acsfont`. Similarly, unless the option `footnote` is specified, `\acffont` handles the output of `\acf`, where the included acronym goes through `\acfsfont` (and `\acsfont`). The plural and upper case forms are treated accordingly. Usually the three macros do nothing. To give an example, the option `smaller` makes `\acsfont` use the command `\textsmaller` from the `reysize` package:

```
\renewcommand*{\acsfont}[1]{\textsmaller{#1}}
```

2.3 Defining Acronyms

Acronyms can either be defined from an environment specifically introduced for that purpose or by direct definitions.

`acronym` The `acronym` environment allows one to define all the acronyms needed by a document at a single place and is self-documenting, since a table of acronyms is automatically produced.

`\acro` In the `acronym` environment, acronyms are defined with the command:

```
\acro{<acronym>}[<short name>]{<full name>}
```

The first argument `<acronym>` is the acronym string itself and is used in the commands of the previous section such as `\ac` or `\acl`, that print the different forms of the acronym.

Because internal commands take `<acronym>` for storing the different forms of the acronym, the T_EX code for the acronym is limited by `\csname`. If the acronym requires problematic or complicated T_EX stuff (font commands, ...), then this code can be given in the optional argument `<short name>`. The first argument `<acronym>` is then a simpler string to identify the acronym. For example, an acronym for water can look like this:

```
\acro{H2O}[$\mathrm{H_2O}] {water}
```

Then `\acs{H2O}` gets “H₂O” and `\acl{H2O}` prints “water”.

`\acroextra` Inside the `acronym` environment additional information can be added to the list of acronyms with the `\acroextra` command that will not be included in the normal inline acronyms.

```
\acroextra{<additional info>}
```

for example:

```
\acro{H2O}[$\mathrm{H_2O}$]
  {Dihydrogen Monoxide\acroextra{ (water)}}}
\acro{NA}[\ensuremath{N_{\mathrm{A}}}]
  {Number of Avogadro\acroextra{ (See \S\protect\ref{A1})}}
```

Note that `\acroextra` must be inserted inside the `\acro` definition and that fragile commands must be protected. Be careful of unnecessary spaces.

The standard format of the acronym list is a `\description` environment. If you pass an optional parameter to the `acronym` environment, the width of the acronym-column will be fitted to the width of the given parameter (which should be the longest acronym). For example, if *HBCI* is the longest acronym used, the list should start with

```
\begin{acronym}[HBCI]
```

`\aclabelfont` The short form of each acronym in the list is formatted using `\aclabelfont`, which typesets its arguments in bold font by default. It can be redefined to produce bold sans-serif labels, for example, with

```
\renewcommand*{\aclabelfont}[1]{\textbf{\textsf{\acsfont{#1}}}}
```

In standard mode, the acronym-list will consist of all defined acronyms, regardless if the the acronym was used in the text before or not. This behavior can be changed by loading the package with the parameter `printonlyused` (used at least once) or `printonlyreused` (use more than once):

```
\usepackage[printonlyused]{acronym}
```

In `printonly(re)used`-mode you can add to each acronym the the page number where it was first used by additionally specifying the option `withpage`.

```
\usepackage[printonlyused,withpage]{acronym}
```

`\newacro` If one does not want an acronym list to be produced at all, acronyms can be defined directly thanks to the two commands

```
\acrodef
\newacro{<acronym>}[<short name>]{<full name>}
\acrodef{<acronym>}[<short name>]{<full name>}
```

the difference between the two consisting in the fact that the latter makes the acronym definition stored in the `.aux` file. Therefore, the acronym becomes available from start-up in the next run.

Note that all the acronym definitions made by `\acro` in the `acronym` environment are also similarly added to the `.aux` file.

*This file has version number v1.49, last revised 2024/03/30.

2.3.1 Non standard indefinite articles

Sometimes the indefinite article of an acronym differs between its short form and its long form, for example “a Federal Bureau of Investigation (FBI) agent” and “an FBI agent”. To deal with this, the package provides the following three commands

```
\newacroindefinite
\acrodefindefinite
\acroindefinite      \acroindefinite{<acronym>}{<short indefinite article>}{<long indefinite
article>}
\newacroindefinite{<acronym>}{<short indefinite article>}{<long
indefinite article>}
\acrodefindefinite{<acronym>}{<short indefinite article>}{<long
indefinite article>}
```

that allow one to define indefinite articles. The `\acroindefinite` command is meant to be used in the `acronym` environment. The difference among the latter two is that `\acrodefindefinite` puts the acronym definition in the `.aux` file, so that the acronym exception is available at the next run from start-up.

When using `\iac` and `\Iac` without first defining an article, the default article is “a”.

2.3.2 Non standard and foreign plural forms

When the plural form of an acronym is required, the package typically obtains it as an English plural, by adding an ‘s’. This happens both for long and short forms. For instance, for an acronym defined as

```
\newacro{IC}{Integrated Circuit}
```

the `\acsp{IC}` command produces “ICs”, and the `\aclp{IC}` command produces “Integrated Circuits”.

Unfortunately, this is generally not suitable for typesetting in languages different from English, and at times it is not correct even for English. For instance consider the “MP” acronym, commonly used to refer to a “Member of the Parliament”. Of course, its long form plural is not “Member of the Parlements”, but “Members of the Parliament”. For the short form plural, “MPs” is anyway commonly accepted. The same happens with “SOC (System on a Chip)” or “BUT (Block Under Test)”.

In foreign languages, things can be even more complicated. For instance, in Italian, there are different rules for English acronyms used in Italian text and Italian acronyms used in Italian text. The former do not get a plural at all, neither for the long, nor for the short form as in “Un paio di *Integrated Circuit (IC)*”. The latter get a plural long form following the natural Italian rules for plurals, and a plural short form that can either be the same as the singular short form, or — at times — a form obtained by doubling those letter of the short form that correspond to words that get a plural in the long form. For instance: “Nucleo Investigativo (NI)” could take a plural as in “Nuclei Investigativi (NNII)”, although in modern texts one is more likely to find “Nuclei Investigativi (NI)”.

To deal with all these different situations, the package (since version 1.35) has been enriched with the following three commands

```
\acroplural
\newacroplural
\acrodefplural
```

```

\acroplural{<acronym>}[<short plural>]{<long plural>}
\newacroplural{<acronym>}[<short plural>]{<long plural>}
\acrodefplural{<acronym>}[<short plural>]{<long plural>}

```

that allow one to define plural exceptions. The `\acroplural` command is meant to be used in the `acronym` environment. The difference among the latter two is that `\acrodefplural` puts the acronym definition in the `.aux` file, so that the acronym exception is available at the next run from start-up. When the optional short form is not provided, the acronym name plus an ‘s’ is used.

Plural exceptions are never reported in tables of acronyms.

2.4 Miscellaneous

2.4.1 Sectioning and pdf marks

Acronyms are robust (since version 1.12) and can be used in sectional headers such as `\chapter`, `\section`, etc., but please note the following:

- Do not use the general form (`\ac` or `\acp`) in sectional headers, because it will use the full name the first time, that is in the table of contents, and the short form further on.
- The text of `<acronym>` is used verbatim in bookmarks and not `<short name>` for pdfTeX with `hyperref`.
- When the long form of the acronym is used in sectional headers (for pdfTeX with `hyperref`), it will end up in the pdf bookmarks. In that case it is good to hide unusual text such as math inside the `\texorpdfstring` defined by `hyperref`, for example:

```

\acro{Nx}[\ensuremath{N_{\chi}}]
{\texorpdfstring{$\chi$}{X}-factor}

```

which will then give

```

pdf bookmark: \acf{Nx} → X-factor (Nx)
text:         \acf{Nx} → χ-factor (Nχ)

```

- For acronyms in sectional headers, the file must be PDFLaTeX'ed 3 times before the bookmarks are correct.
- Acronyms in sectional headers together with the `footnote` option will not give reliable results, because it will end up in the running heads and table of contents. If you really need it, use the optional argument of the sectioning commands. For example:

```

\chapter[The water \texorpdfstring{$\mathrm{H_2O}$}{H2O}) ...]
{The \acf{H2O} ...}

```

3 An example file

```
1 (*acrotest)
2 \documentclass{article}
3 \usepackage[colorlinks]{hyperref}
4 \usepackage[printonlyused,withpage]{acronym}
5 \begin{document}
6
7 \section{Intro}
8 In the early nineties, \acs{GSM} was deployed in many European
9 countries. \ac{GSM} offered for the first time international
10 roaming for mobile subscribers. The \acs{GSM}'s use of \ac{TDMA} as
11 its communication standard was debated at length. And every now
12 and then there are big discussion whether \ac{CDMA} should have
13 been chosen over \ac{TDMA}.
14
15 \section{Furthermore}
16 \acresetall
17 The reader could have forgotten all the nice acronyms, so we repeat the
18 meaning again.
19
20 If you want to know more about \acf{GSM}, \acf{TDMA}, \acf{CDMA}
21 and other acronyms, just read a book about mobile communication. Just
22 to mention it: There is another \ac{UA}, just for testing purposes!
23
24 \begin{figure}[h]
25 Figure
26 \caption{A float also admits references like \ac{GSM} or \acf{CDMA}.}
27 \end{figure}
28
29 \subsection{Some chemistry and physics}
30 \label{Chem}
31 \ac{NAD+} is a major electron acceptor in the oxidation
32 of fuel molecules. The reactive part of \ac{NAD+} is its nictinamide
33 ring, a pyridine derivate.
34
35 One mol consists of \acs{NA} atoms or molecules. There is a relation
36 between the constant of Boltzmann and the \acl{NA}:
37 \begin{equation}
38 k = R/\acs{NA}
39 \end{equation}
40
41 \acl{lox}/\acl{lh2} (\acsu{lox}/\acsu{lh2})
42
43 \Acp{LFVP} are processes in which the lepton number of the initial
44 and final states are different. An example for \iac{LFVP} is
45 neutrinoless double beta decay.
46
47 \subsection{Some testing fundamentals}
48 When testing \acp{IC}, one typically wants to identify functional
49 blocks to be tested separately. The latter are commonly indicated as
50 \acp{BUT}. To test a \ac{BUT} requires defining a testing strategy\dotsc{}
51 \Iac{IC} popped up unexpectedly.
52
```

```

53 \section{Acronyms}
54 \begin{acronym}[TDMA]
55 \acro{CDMA}{Code Division Multiple Access}
56 \acro{GSM}{Global System for Mobile communication}
57 \acro{NA}[\ensuremath{N_{\mathrm{A}}}]
58     {Number of Avogadro\acroextra{ (see \S\ref{Chem})}}
59 \acro{NAD+}[NAD\textsuperscript{+}]{Nicotinamide Adenine Dinucleotide}
60 \acro{LFVP}{lepton flavor violating process}
61 \acroindefinite{LFVP}{an}{a}
62 \acro{NUA}{Not Used Acronym}
63 \acro{TDMA}{Time Division Multiple Access}
64 \acro{UA}{Used Acronym}
65 \acro{lox}[\ensuremath{LOX}]{Liquid Oxygen}%
66 \acro{lh2}[\ensuremath{LH_2}]{Liquid Hydrogen}%
67 \acro{IC}{Integrated Circuit}%
68 \acro{BUT}{Block Under Test}%
69 \acrodefplural{BUT}{Blocks Under Test}%
70 \acroindefinite{IC}{an}{an}
71 \end{acronym}
72
73 \end{document}
74 </acrotest>

```


4 The implementation

75 `*acronym`

4.1 Identification

First we test that we got the right format and name the package.

76 `\NeedsTeXFormat{LaTeX2e}[1999/12/01]`

77 `\ProvidesPackage{acronym}[2024/03/30]`

78 `v1.49`

79 `Support for acronyms (Tobias Oetiker)`

80 `\RequirePackage{suffix,xstring}`

4.2 Options

`\ifAC@footnote` The option `footnote` leads to a redefinition of `\acf`, `\Acf`, `\acfp`, and `\Acfp`, making the full name appear as a footnote.

81 `\newif\ifAC@footnote`

82 `\AC@footnotefalse`

83 `\DeclareOption{footnote}{\AC@footnotetrue}`

`\ifAC@nohyperlinks` If `hyperref` is loaded, all acronyms will link to their glossary entry and the glossary entries to the first in-text use. With the option `nohyperlinks` these links can be suppressed.

84 `\newif\ifAC@nohyperlinks`

85 `\AC@nohyperlinksfalse`

86 `\DeclareOption{nohyperlinks}{\AC@nohyperlinkstrue}`

`\ifAC@noforwardlinks` If `hyperref` is loaded and the `nohyperlinks` option is not selected, the option `noforwardlinks` suppresses the links from the glossary entries to the in-text use.

87 `\newif\ifAC@noforwardlinks`

88 `\AC@noforwardlinksfalse`

89 `\DeclareOption{noforwardlinks}{\AC@noforwardlinkstrue}`

`\ifAC@noacroprefix` With the `noacroprefix` option the acronym commands are not prefixed. This reproduces the old behavior of version 1.43, but can cause collisions between user-defined acronyms and commands of this package.

90 `\newif\ifAC@noacroprefix`

91 `\AC@noacroprefixfalse`

92 `\DeclareOption{noacroprefix}{\AC@noacroprefixtrue}`

`\ifAC@printonlyused` We need a marker which is set if the option `printonlyused` was used.

93 `\newif\ifAC@printonlyused`

94 `\AC@printonlyusedfalse`

95 `\DeclareOption{printonlyused}{\AC@printonlyusedtrue}`

`\ifAC@printonlyreused` With the `printonlyreused` option, only those acronyms are included in the list of acronyms that have been used more than once, i.e. at least twice.

96 `\newif\ifAC@printonlyreused`

97 `\AC@printonlyreusedfalse`

```

98 \DeclareOption{printonlyreused}{\AC@printonlyreusedtrue}

\ifAC@withpage A marker which tells us to print page numbers.
99 \newif\ifAC@withpage
100 \AC@withpagefalse
101 \DeclareOption{withpage}{\AC@withpagetrue}

\ifAC@smaller The option smaller leads to a redefinition of \acsfont. We want to make
the acronym appear smaller. Since this should be done in a context-sensitive
way, we rely on the macro \textsmaller provided by the relsize package. As
\RequirePackage cannot be used inside \DeclareOption, we need a boolean vari-
able.
102 \newif\ifAC@smaller
103 \AC@smallerfalse
104 \DeclareOption{smaller}{\AC@smallertrue}

\ifAC@dua The option dua stands for “don’t use acronyms”. It leads to a redefinition of \ac,
\Ac, \acp, and \Acp, making the full name appear all the time and suppressing
all acronyms but the explicitly requested by \acf, \Acf, \acfp or \Acfp.
105 \newif\ifAC@dua
106 \AC@duafalse
107 \DeclareOption{dua}{\AC@duatrue}

\ifAC@nolist The option nolist stands for “don’t write the list of acronyms”.
108 \newif\ifAC@nolist
109 \AC@nolistfalse
110 \DeclareOption{nolist}{\AC@nolisttrue\AC@nohyperlinkstrue}

\ifAC@nolinebreak The option nolinebreak dictates whether to forbid, by default, a line break between
the full name and the short name, when they are presented together.
111 \newif\ifAC@nolinebreak
112 \AC@nolinebreakfalse
113 \DeclareOption{nolinebreak}{\AC@nolinebreaktrue}

Now we process the options.
114 \ProcessOptions\relax

```

4.3 Setup macros

`\acsfont` The appearance of the output of the commands `\acs` and `\acf` is partially con-
`\acffont` trolled by `\acsfont`, `\acffont`, and `\acfsfont`. By default, they do nothing
`\acfsfont` except when the `smaller` option is loaded.

The option `smaller` leads to a redefinition of `\acsfont`. We want to make the
acronym appear smaller. Since this should be done in a context-sensitive way, we
rely on the macro `\textsmaller` provided by the `relsize` package.

```

115 \ifAC@smaller
116 \RequirePackage{relsize}
117 \newcommand*{\acsfont}[1]{\textsmaller{#1}}
118 \else
119 \newcommand*{\acsfont}[1]{#1}
120 \fi
121 \newcommand*{\acffont}[1]{#1}
122 \newcommand*{\acfsfont}[1]{#1}

```

`\AC@linebreakpenalty` When the option `nolinebreak` is specified, the default penalty for a line break is being set to the maximum. Otherwise, the default penalty is one level below the maximum, meaning that most of the times, by default, the line will not get broken.

```

123 \ifAC@nolinebreak
124   \def\AC@linebreakpenalty{4}
125 \else
126   \def\AC@linebreakpenalty{3}
127 \fi

```

4.4 Hyperlinks and PDF support

`\AC@hyperlink` Define dummy hyperlink commands

```

\AC@hyperref 128 \def\AC@hyperlink#1#2{#2}
\AC@hypertarget 129 \def\AC@hyperref[#1]#2{#2}
\AC@phantomsection 130 \def\AC@hypertarget#1#2{#2}
131 \def\AC@phantomsection{}

```

`\AC@raisedhypertarget` Make sure that hyperlink processing gets enabled before we process the document if `hyperref` has been loaded in the mean time.

```

132 \ifAC@nohyperlinks
133 \else
134   \AtBeginDocument{%
135     \ifpackageloaded{hyperref}
136       {\let\AC@hyperlink=\hyperlink
137        \ifAC@noforwardlinks\else\let\AC@hyperref=\hyperref\fi
138        \newcommand*\AC@raisedhypertarget[2]{%
139          \Hy@raisedlink{\hypertarget{#1}{}}#2}%
140        \let\AC@hypertarget=\AC@raisedhypertarget
141        \def\AC@phantomsection{%
142          \Hy@GlobalStepCount\Hy@linkcounter
143          \edef\@currentHref{section*.\the \Hy@linkcounter}%
144          \Hy@raisedlink{%
145            \hyper@anchorstart{\@currentHref}\hyper@anchorend
146            }%
147          }%
148          }{}%
149 \fi

```

`\AC@pageref` Use `\pageref*` instead of `\pageref` when the `hyperref` package is used.

```

150 \AtBeginDocument{%
151   \ifpackageloaded{hyperref}{%
152     \let\AC@pageref=\pagerefstar%
153   }{%
154     \let\AC@pageref=\pageref%
155   }%
156 }

```

The `hyperref` package defines `\pdfstringdefDisableCommands` and `\texorpdfstring` for text in bookmarks. If undefined, then provide them it at the beginning of the document.

```

157 \AtBeginDocument{%

```

```

158 \providecommand\texorpdfstring[2]{#1}%
159 \providecommand\pdfstringdefDisableCommands[1]{}%
160 \pdfstringdefDisableCommands{%
161   \csname AC@starredfalse\endcsname
162   \csname AC@footnotefalse\endcsname
163   \let\AC@hyperlink\@secondoftwo
164   \let\acsfont\relax
165   \let\acffont\relax
166   \let\acfsfont\relax
167   \let\acused\relax
168   \let\null\relax
169   \def\AChy@call#1#2{%
170     \ifx*#1\@empty
171       \expandafter #2%
172     \else
173       #2{#1}%
174     \fi
175   }%
176   \def\acs#1{\AChy@call{#1}\AC@acs}%
177   \def\acl#1{\AChy@call{#1}\@acl}%
178   \def\Acl#1{\AChy@call{#1}\@Acl}%
179   \def\acf#1{\AChy@call{#1}\AChy@acf}%
180   \def\Acf#1{\AChy@call{#1}\AChy@Acf}%
181   \def\ac#1{\AChy@call{#1}\@ac}%
182   \def\Ac#1{\AChy@call{#1}\@Ac}%
183   \def\acsp#1{\AChy@call{#1}\@acsp}%
184   \def\aclp#1{\AChy@call{#1}\@aclp}%
185   \def\Aclp#1{\AChy@call{#1}\@Aclp}%
186   \def\acfp#1{\AChy@call{#1}\AChy@acfp}%
187   \def\Acfp#1{\AChy@call{#1}\AChy@Acfp}%
188   \def\acp#1{\AChy@call{#1}\@acp}%
189   \def\Acp#1{\AChy@call{#1}\@Acp}%
190   \def\acfi#1{\AChy@call{#1}\AChy@acf}%
191   \def\Acfi#1{\AChy@call{#1}\AChy@Acf}%
192   \def\acfip#1{\AChy@call{#1}\AChy@acfp}%
193   \def\Acfip#1{\AChy@call{#1}\AChy@Acfp}%
194   \let\acsu\acs
195   \let\aclu\acl
196   \let\Aclu\Acl
197   \def\AChy@acf#1{\AC@acl{#1} (\AC@acs{#1})}%
198   \def\AChy@Acf#1{\AC@Acl{#1} (\AC@acs{#1})}%
199   \def\AChy@acfp#1{\AC@aclp{#1} (\AC@acsp{#1})}%
200   \def\AChy@Acfp#1{\AC@Aclp{#1} (\AC@acsp{#1})}%
201 }%
202 }

```

4.5 Additional Helper macros

We need a list of the used acronyms after the last `\acresetall` (or since beginning), a token list is very useful for this purpose

```
AC@clearlist
```

```
203 \newtoks\AC@clearlist
```

```

\AC@addtoAC@clearlist Adds acronyms to the clear list
204 \newcommand*\AC@addtoAC@clearlist[1]{%
205   \global\AC@clearlist\expandafter{\the\AC@clearlist\AC@reset{#1}}%
206 }

\acresetall This macro resets the AC@FN - tag of each acronym, therefore \ac will use Full
\AC@reset Name (FN) next time it is called
207 \newcommand*\acresetall{\the\AC@clearlist\AC@clearlist={}}
208 \def\AC@reset#1{%
209   \global\expandafter\let\csname AC@\AC@prefix#1\endcsname\relax
210 }

\AC@used We also need a markers for 'used'.
211 \newcommand*\AC@used{@<>@<>@}

\AC@populated An on/off flag to note if any acronyms were logged. This is needed for the first
run with printonly(re)used option, because the acronym list are then empty,
resulting in a "missing item" error.
212 \newcommand{\AC@populated}{-}

\AC@logged Log the usage by writing the \acronymused to the aux file and by reading it back
\acronymused again at the beginning of the document (performed automatically by LaTeX). This
results in processing the document twice, but it is needed anyway for the rest of
the package.
This methodology is needed when the list of acronyms is in the front matter
of the document.
213 \newcommand*\AC@logged}[1]{%
214   \@bsphack
215   \protected@write\@auxout{}{\string\acronymused{#1}}%
216   \@esphack}

Keep it out of bookmarks.
217 \AtBeginDocument{%
218   \pdfstringdefDisableCommands{%
219     \let\AC@logged@gobble
220   }%
221 }

Flag the acronym at the beginning of the document as used (called by the aux
file).
222 \newcommand*\acronymused}[1]{%
223   \expandafter\ifx\csname acused@#1@once\endcsname\AC@used%
224     \expandafter\ifx\csname acused@#1@twice\endcsname\AC@used%
225     \relax%
226   \else%
227     \global\expandafter\let\csname acused@#1@twice\endcsname\AC@used%
228     \global\let\AC@populated\AC@used%
229   \fi%
230 \else%
231   \global\expandafter\let\csname acused@#1@once\endcsname\AC@used%
232   \ifAC@printonlyreused%
233     \relax%
234   \else%

```

```

235     \global\let\AC@populated\AC@used%
236     \fi%
237 \fi%
238 }

```

`firstupper, \@firstupper@maybe` Internal commands for making a first letter upper case.

```

239 \newcommand{\@firstupper}[1]{%
240   \StrSplit{#1}{1}{\head}{\tail}%
241   \MakeUppercase\head\tail%
242 }
243 \newcommand{\@firstupper@maybe}{%
244   \ifAC@upper
245     \expandafter\@firstupper
246   \else
247     \expandafter\@firstofone
248   \fi
249 }

```

`AC@prefix` Returns the prefix used to build the defined acronym commands as long as the `noacroprefix` option is disabled. Otherwise the output is empty, so the old behaviour from version 1.43 is reproduced.

```

250 \ifAC@noacroprefix
251   \newcommand*{\AC@prefix{}}
252 \else
253   \newcommand*{\AC@prefix{acronyms@}}
254 \fi

```

4.6 Defining acronyms

There are three commands that define acronyms: `\newacro`, `\acrodef`, and `\acro`. They are called with the following arguments:

$$\acro{\langle acronym \rangle}[\langle short name \rangle]{\langle full name \rangle}$$

The mechanism used in this package is to make the optional $\langle short name \rangle$ identical to the $\langle acronym \rangle$ when it is empty (no optional argument), thereby only the second (optional) argument is stored together with the $\langle full name \rangle$.

`\newacro` The internal macro `\newacro` stores the $\langle short name \rangle$ and the $\langle full name \rangle$ in the `\AC@newacro` command `\fn@<acronym>`.

```

255 \newcommand*\newacro[1]{%
256   \@ifnextchar[{\AC@newacro{#1}}{\AC@newacro{#1}[#1]}
257 \newcommand\AC@newacro{}
258 \def\AC@newacro#1[#2]#3{%
259   \expandafter\gdef\csname fn@#1\endcsname{{#2}{#3}}%
260   }

```

`\acrodef` The user command `\acrodef` calls `\newacro` and writes it into the `.aux` file.

```

\AC@acrodef 261 \newcommand*\acrodef[1]{%
262   \@ifnextchar[{\AC@acrodef{#1}}{\AC@acrodef{#1}[#1]}

```

```

263 \newcommand\AC@acrodef{}
264 \def\AC@acrodef#1[#2]#3{%
265     \@bsphack
266     \protected@write\@auxout{}\string\newacro{#1}[#2]{#3}%
267     \@esphack}

```

AC@deflist In standard mode, the acronym list is formatted with a description environment. If an optional argument is passed to the acronym environment, the list is formatted as a AC@deflist, which needs the longest appearing acronym as parameter. If the option 'nolist' is selected the environment is empty.

```

268 \newcommand*\aclabelfont}[1]{\textbf{\acsfont{#1}}}
269 \def\AC@makelabel#1{#1\hfil}
270 \newenvironment{AC@deflist}[1]%
271     {\ifAC@nolist%
272     \else%
273         \raggedright\begin{list}{}%
274             {\settowidth{\labelwidth}{\AC@makelabel{\aclabelfont{#1}}}%
275             \setlength{\leftmargin}{\labelwidth}%
276             \addtolength{\leftmargin}{\labelsep}%
277             \renewcommand{\makelabel}{\AC@makelabel}}%
278         \fi}%
279     {\ifAC@nolist%
280     \else%
281         \end{list}%
282     \fi}%

```

acronym In the 'acronym' - environment, all acronyms are defined, and printed if they have been used before, which is indicated by the acused-tag.

```

\begin{acronym}
\acro{CDMA}{Code Division Multiple Access\acroextra{\ ...}}
\end{acronym}

```

\acroextra Additional information can be added after to \acro definition for display in the list of acronyms. This command is only active inside the acronym environment. Outside it gobbles up its argument.

```

283 \newcommand\acroextra[1]{ }

```

\acro Acronyms can be defined with the user command \acro inside the acronym environment.

```

284 \newenvironment{acronym}[1][1]{%
285     \providecommand*\acro{\AC@acro}%
286     \providecommand*\acroplural{\AC@acroplural}%
287     \providecommand*\acroindefinite{\AC@acroindefinite}%
288     \long\def\acroextra##1{##1}%
289     \def\@tempa{1}\def\@tempb{#1}%
290     \ifx\@tempa\@tempb%
291         \global\expandafter\let\csname AC@des@mark\endcsname\AC@used%
292         \ifAC@nolist%
293         \else%
294             \begin{description}%
295             \fi%
296     \else%

```

```

297     \begin{AC@deflist}{#1}%
298     \fi%
299 }%
300 {%
301   \ifx\AC@populated\AC@used\else%
302     \ifAC@nolist%
303     \else%
304       \item[]\relax%
305     \fi%
306   \fi%
307   \expandafter\ifx\csname AC@des@mark\endcsname\AC@used%
308     \ifAC@nolist%
309     \else%
310       \end{description}%
311     \fi%
312   \else%
313     \end{AC@deflist}%
314   \fi}%

\AC@acro
\AC@@acro 315 \newcommand* \AC@acro[1]{%
316   \@ifnextchar[ {%
317     \csname AC@\AC@prefix{}@acro\endcsname{#1}%
318   }{%
319     \csname AC@\AC@prefix{}@acro\endcsname{#1}[#1]%
320   }%
321 }

322 \expandafter\newcommand\csname AC@\AC@prefix{}@acro\endcsname{
323 \expandafter\def\csname AC@\AC@prefix{}@acro\endcsname#1[#2]#3{%
324   \ifAC@nolist%
325   \else%
326   \ifnum%
327     \ifAC@printonlyused 1%
328     \else\ifAC@printonlyreused 1%
329     \else 0\fi\fi%
330 =1\relax%
331   \ifnum%
332     \ifAC@printonlyused%
333     \expandafter\ifx\csname acused@#1@once\endcsname\AC@used 1 \else 0 \fi%
334     \else\ifAC@printonlyreused%
335     \expandafter\ifx\csname acused@#1@twice\endcsname\AC@used 1 \else 0 \fi%
336     \else 0 \fi\fi%
337 =1\relax%
338   \item[\protect\AC@hypertarget{#1}]{%
339     \AC@hyperref[acro:#1]{\aclabelfont{#2}\hfill}%
340   }]\AC@hyperref[acro:#1]{#3}%
341   \ifAC@withpage%
342     \expandafter\ifx\csname r@acro:#1\endcsname\relax%
343     \PackageInfo{acronym}{%
344       Acronym #1 used in text but not spelled out in
345       full in text}%
346     \else%
347     \nobreak\leaders\hbox{$\m@th\mkern\@dotsep mu\hbox{.}\mkern\@dotsep mu$}\hfill%
348     \nobreak\hb@xt@\@pnumwidth{%

```



```

349             \hfil\normalfont\normalcolor\AC@pageref{acro:#1}%
350             }%
351             \fi%
352         \fi\\%
353     \fi%
354 \else%
355     \item[\protect\AC@hypertarget{#1}{\AC@hyperref[acro:#1]{\aclabelfont{#2}\hfill}}]\AC@hyper
356 \fi%
357 \fi%
358 \begingroup
359     \def\acroextra##1{%
360     \@bsphack
361     \ifAC@printonlyreused%
362     \protected@write\@auxout{}{%
363     \string\newacro{#1}[%
364     \expandafter\ifx\csname acused@#1@twice\endcsname\AC@used%
365     \string\AC@hyperlink{#1}{#2}%
366     \else%
367     {#2}%
368     \fi%
369     ]{#3}%
370     }%
371     \else%
372     \protected@write\@auxout{}{%
373     \string\newacro{#1}[\string\AC@hyperlink{#1}{#2}]{#3}%
374     }%
375     \fi%
376     \@esphack
377 \endgroup
378 \ignorespaces}

```

4.6.1 Nonstandard indefinite articles

`\newacroindefinite` Sets up a non standard indefinite article for a given acronym.

```

379 \newcommand*\newacroindefinite[3]{%
380 \expandafter\gdef\csname fn@#1@IS\endcsname{#2}%
381 \expandafter\gdef\csname fn@#1@IL\endcsname{#3}%
382 }

```

`\acrodefindefinite` Same as above, storing content in aux file.

```

383 \newcommand*\acrodefindefinite[3]{%
384 \@bsphack
385 \protected@write\@auxout{}{\string\newacroindefinite{#1}{#2}{#3}}%
386 \@esphack
387 }

```

`\AC@acroindefinite` Internal command to set up an indefinite article in the acronym environment.

```

388 \newcommand\AC@acroindefinite[3]{
389 \@bsphack
390 \protected@write\@auxout{}%
391 {\string\newacroindefinite{#1}{#2}{#3}}%
392 \@esphack
393 }

```

4.6.2 Non standard or foreign plural forms

```

\newacroplural Sets up a non standard plural form for a given acronym.
\AC@newacroplurali 394 \newcommand*\newacroplural[1]{%
\AC@newacropluralii 395 \ifnextchar[%
396 {\AC@newacroplurali{#1}}{\AC@newacropluralii{#1}}%
397 }
398 \newcommand\AC@newacroplurali{}
399 \def\AC@newacroplurali#1[#2]#3{%
400 \expandafter\gdef\csname fn@#1@PS\endcsname{#2}%
401 \expandafter\gdef\csname fn@#1@PL\endcsname{#3}%
402 }
403 \newcommand\AC@newacropluralii[2]{%
404 \expandafter\gdef\csname fn@#1@PL\endcsname{#2}%
405 }

\acrodefplural Same as above, storing content in aux file.
\AC@acrodefplurali 406 \newcommand*\acrodefplural[1]{%
\AC@acrodefpluralii 407 \ifnextchar[%
408 {\AC@acrodefplurali{#1}}{\AC@acrodefpluralii{#1}}%
409 }
410 \newcommand\AC@acrodefplurali{}
411 \def\AC@acrodefplurali#1[#2]#3{%
412 \@bsphack
413 \protected@write\@auxout{}\string\newacroplural{#1}[#2]{#3}%
414 \@esphack
415 }
416 \newcommand\AC@acrodefpluralii[2]{%
417 \@bsphack
418 \protected@write\@auxout{}\string\newacroplural{#1}{#2}%
419 \@esphack
420 }

\AC@acroplural Internal commands to set up a plural version of an acronym in the acronym envi-
\AC@acroplurali ronment.
\AC@acropluralii 421 \newcommand*\AC@acroplural[1]{%
422 \ifnextchar[%
423 {\AC@acroplurali{#1}}{\AC@acropluralii{#1}}%
424 }
425 \newcommand\AC@acroplurali{}
426 \def\AC@acroplurali#1[#2]#3{%
427 \@bsphack
428 \protected@write\@auxout{%
429 {\string\newacroplural{#1}[\string\AC@hyperlink{#1}]{#2}]{#3}}%
430 \@esphack
431 }
432 \newcommand\AC@acropluralii[2]{
433 \@bsphack
434 \protected@write\@auxout{%
435 {\string\newacroplural{#1}[\string\AC@hyperlink{#1}{\AC@acs{#1}}]{#2}}%
436 \@esphack
437 }

\AC@aclp Deliver either standard or nonstandard plural form (long and short respectively).
\AC@Aclp
\AC@acsp

```

```

438 \newcommand\AC@aclp[1]{%
439   \ifcsname fn@#1@PL\endcsname
440     \@firstupper@maybe{\csname fn@#1@PL\endcsname}%
441   \else
442     \AC@acl{#1}s%
443   \fi
444 }
445 \newcommand\AC@Aclp[1]{%
446   \AC@uppertrue%
447   \AC@aclp{#1}%
448   \AC@upperfalse%
449 }
450 \newcommand\AC@acsp[1]{%
451   \ifcsname fn@#1@PS\endcsname
452     \csname fn@#1@PS\endcsname
453   \else
454     \AC@acs{#1}s%
455   \fi
456 }

```

4.7 Using acronyms

`\ifAC@starred` Before the macros are defined, we need a boolean variable which will be set to true or false, when the following commands are used in the starred or unstarred form. If it is true, the acronym will be not be logged, otherwise it will be logged.

```
457 \newif\ifAC@starred
```

`\ifAC@upper` If an acronym needs to be capitalized, this flag is used to indicate this at an appropriate point in the code. In that case, the `firstupper` command will be called at a time when the acronym is expandable, otherwise the `xstring` command will not work properly.

```
458 \newif\ifAC@upper
```

`\AC@get` If the acronym is undefined, the internal macro `\AC@get` warns the user by printing the name in bold with an exclamation mark at the end. If defined, `\AC@get` uses the same mechanism used by the LaTeX kernel commands `\ref` and `\pageref` to return the short `\AC@acs` and long forms `\AC@acl` of the acronym saved in `\fn@<acronym>`.

```

459 \newcommand\AC@get[3]{%
460   \ifx#1\relax
461     \PackageWarning{acronym}{Acronym ‘#3’ is not defined}%
462     \textbf{#3!}%
463   \else
464     \@firstupper@maybe{\expandafter#2#1}%
465   \fi
466 }

```

`\AC@acs` The internal commands `\AC@acs` and `\AC@acl` returns the (unformatted) short and the long forms of an acronym as saved in `\fn@<acronym>`. Mbox to prevent `\AC@Acl` hyphenation of short form.

```

467 \newcommand\AC@acs[1]{%
468   \mbox{\expandafter\AC@get\csname fn@#1\endcsname\@firstoftwo{#1}}}

```

```

469 \newcommand*\AC@acl[1]{%
470   \expandafter\AC@get\csname fn@#1\endcsname\@secondoftwo{#1}}
471 \newcommand*\AC@Acl[1]{%
472   \AC@uppertrue%
473   \AC@acl{#1}%
474   \AC@upperfalse%
475 }

```

`\acs` The user macro `\acs` prints the short form of the acronym using the font specified by `\acsfont`.

```

\@acs 476 \newcommand*\acs{\AC@starredfalse\protect\acs}%
477 \WithSuffix\newcommand\acs*\AC@starredtrue\protect\acs}%
478 \newcommand*\acs}[1]{%
479   \texorpdfstring{\protect\acs{#1}}{#1}}
480 \newcommand*\@acs}[1]{%
481   \acsfont{\AC@placelabel@part{#1}\AC@acs{#1}}%
482 %% having a footnote on acs sort of defeats the purpose
483 %%   \ifAC@footnote
484 %%     \footnote{\AC@acl{#1}}}%
485 %%   \fi
486 \ifAC@starred\else\AC@logged{#1}\fi}

```

`\acl` The user macro `\acl` prints the full name of the acronym.

```

\@acl 487 \newcommand*\acl{\AC@starredfalse\protect\@acl}%
\Acl 488 \WithSuffix\newcommand\acl*\AC@starredtrue\protect\@acl}%
\@Acl 489 \newcommand*\Acl{\AC@starredfalse\protect\@Acl}%
490 \WithSuffix\newcommand\Acl*\AC@starredtrue\protect\@Acl}%
491 \newcommand*\@acl}[1]{%
492   \AC@placelabel@part{#1}\AC@acl{#1}%
493   \ifAC@starred\else\AC@logged{#1}\fi}
494 \newcommand*\@Acl}[1]{%
495   \AC@placelabel@part{#1}\AC@Acl{#1}%
496   \ifAC@starred\else\AC@logged{#1}\fi}

```

4.8 Helper functions to unset labels

`\@verridelabel` The internal `\@verridelabel` command lets us 'redefine' an acronym label such that the page reference in the acronym list points where it should be pointing and not just to the very first occurrence of the acronym, where it may not even be expanded. (code by Ulrich Diez)

```

497 \newcommand*\@verridelabel[1]{%
498   \@bsphack
499   \protected@write\@auxout{}\string\AC@undonewlabel{#1}%
500   \label{#1}%
501   \AC@overriddenmessage rs{#1}%
502   \@esphack
503 }%
504 \newcommand*\AC@undonewlabel{\AC@und@newl@bel rs}%
505 \newcommand*\AC@und@newl@bel[3]{%
506   \@ifundefined{#1@#3}%

```

```

507 {%
508   \global\expandafter\let\csname#2@#3\endcsname\@nnil
509 }%
510 {%
511   \global\expandafter\let\csname#1@#3\endcsname\relax
512 }%
513 }%
514 \newcommand*{\AC@overriddenmessage[3]{%
515   \expandafter\ifx\csname#2@#3\endcsname\@nnil
516     \expandafter\@firstoftwo
517   \else
518     \@ifundefined{#1@#3}%
519     {%
520       \@ifundefined{#2@#3}%
521       {\expandafter\@firstoftwo}%
522       {\expandafter\@secondoftwo}%
523     }%
524     {\expandafter\@secondoftwo}%
525   \fi
526   {%
527     \PackageInfo{acronym}{Label ‘#3’ newly defined as it
528     shall be overridden^^Jalthough it is yet undefined}%
529     \global\expandafter\let\csname#2@#3\endcsname\empty
530   }%
531   {%
532     \PackageInfo{acronym}{Label ‘#3’ overridden}%
533     \@ifundefined{#2@#3}{%
534       \global\expandafter\let\csname#2@#3\endcsname\empty}{}%
535     \expandafter\g@addto@macro\csname#2@#3\endcsname{i}%
536   }%
537 }%
538 \newcommand*{\AC@testdef[3]{%
539   \@ifundefined{s@#2}\@secondoftwo\@firstofone
540   {%
541     \expandafter\ifx\csname s@#2\endcsname\empty
542       \expandafter\@firstofone
543     \else
544       \expandafter\xdef\csname s@#2\endcsname{%
545         \expandafter\expandafter
546         \expandafter@gobble
547         \csname s@#2\endcsname
548       }%
549       \expandafter@gobble
550     \fi
551   }%
552   {%
553     \@testdef{#1}{#2}{#3}%
554   }%
555 }%
556 \AtBeginDocument{\immediate\write\@auxout{\string\AC@reset@newl@bel}}
557 \newcommand*{\AC@reset@newl@bel{%
558   \ifx\@newl@bel\@testdef
559     \let\@newl@bel\AC@testdef
560     \let\AC@undonewlabel@gobble

```

```

561 \fi
562 }%
563 \newcommand*\AC@placelabel@part[1]{%
564 \expandafter\ifx\csname AC@\AC@prefix#1\endcsname\AC@used
565 \else
566 {\AC@phantomsection\@verridelabel{acro:#1}}%
567 \fi
568 }%
569 \newcommand*\AC@placelabel[1]{%
570 \expandafter\ifx\csname AC@\AC@prefix#1\endcsname\AC@used
571 \else
572 {\AC@phantomsection\@verridelabel{acro:#1}}%
573 \ifAC@starred\else%
574 \global\expandafter\let\csname AC@\AC@prefix#1\endcsname\AC@used
575 \fi%
576 \AC@addtoAC@clearlist{#1}%
577 \fi
578 }%

```

\acf The user macro `\acf` always prints the full name with the acronym. The format
\acfa depends on `\acffont` and `\acfsfont`, and on the option `footnote` handled below.
\@acf The acronym is added to the clear list to keep track of the used acronyms and it
\Acf is marked as used by by `\gdefining` the `\AC@FN` to be `\AC@used` after its first use.
\Acfa The option `footnote` leads to a redefinition of `\acf`, making the full name
\@Acf appear as a footnote. There is then no need for `\acffont` and `\acfsfont`. If the
option `footnote` is not specified, the optional variable determines the penalty for
a line break.

```

579 \newcommand*\acf{\AC@starredfalse\protect\acfa}%
580 \WithSuffix\newcommand\acf*\AC@starredtrue\protect\acfa}%
581 \newcommand*\Acf{\AC@starredfalse\protect\Acfa}%
582 \WithSuffix\newcommand\Acf*\AC@starredtrue\protect\Acfa}%
583 \newcommand*\acfa[2][\AC@linebreakpenalty]{%
584 \texorpdfstring{\protect\@acf[#1]{#2}}{\AC@acl{#2} (#2)}}
585 \newcommand*\Acfa[2][\AC@linebreakpenalty]{%
586 \texorpdfstring{\protect\@Acf[#1]{#2}}{\AC@Acl{#2} (#2)}}
587 \newcommand*\@acf[2][\AC@linebreakpenalty]{%
588 \ifAC@footnote
589 \acsfont{\AC@acs{#2}}%
590 \footnote{\AC@placelabel{#2}\AC@acl{#2}{}}%
591 \else
592 \acffont{%
593 \AC@placelabel{#2}\AC@acl{#2}%
594 \nolinebreak[#1] %
595 \acfsfont{(\acsfont{\AC@acs{#2}})}%
596 }%
597 \fi
598 \ifAC@starred\else\AC@logged{#2}\fi}
599 \newcommand*\@Acf[2][\AC@linebreakpenalty]{%
600 \ifAC@footnote
601 \acsfont{\AC@acs{#2}}%
602 \footnote{\AC@placelabel{#2}\AC@Acl{#2}{}}%

```

```

603 \else
604 \acffont{%
605 \AC@placelabel{#2}\AC@Acl{#2}%
606 \nolinebreak[#1] %
607 \acfsfont{(\acsfont{\AC@acs{#2}})}%
608 }%
609 \fi
610 \ifAC@starred\else\AC@logged{#2}\fi}

```

\ac The first time an acronym is accessed its Full Name (FN) is printed. The next
\@ac time just (FN). When the footnote option is used the short form (FN) is always
\Ac used. The optional variable is being passed to \acf, in case it is used.

```

\@Ac 611 \newcommand*\ac{\AC@starredfalse\protect\@ac}%
612 \WithSuffix\newcommand\ac*\AC@starredtrue\protect\@ac}%

613 \newcommand*\Ac{\AC@starredfalse\protect\@Ac}%
614 \WithSuffix\newcommand\Ac*\AC@starredtrue\protect\@Ac}%

615 \newcommand{\@ac}[2][\AC@linebreakpenalty]{%
616 \ifAC@dua
617 \ifAC@starred\acl*{#2}\else\acl{#2}\fi%
618 \else
619 \expandafter\ifx\csname AC@\AC@prefix#2\endcsname\AC@used%
620 \ifAC@starred\acs*{#2}\else\acs{#2}\fi%
621 \else
622 \ifAC@starred\acf*{#1}{#2}\else\acf{#1}{#2}\fi%
623 \fi
624 \fi}

625 \newcommand{\@Ac}[2][\AC@linebreakpenalty]{%
626 \ifAC@dua
627 \ifAC@starred\Acl*{#2}\else\Acl{#2}\fi%
628 \else
629 \expandafter\ifx\csname AC@\AC@prefix#2\endcsname\AC@used%
630 \ifAC@starred\acs*{#2}\else\acs{#2}\fi%
631 \else
632 \ifAC@starred\Acf*{#1}{#2}\else\Acf{#1}{#2}\fi%
633 \fi
634 \fi}

```

\iac Indefinite article correct expansion. The optional variable is being passed to \ac.

```

\@iac 635 \newcommand*\iac{\AC@starredfalse\protect\@iac}%
\@iaci 636 \WithSuffix\newcommand\iac*\AC@starredtrue\protect\@iac}%
\Iac 637 \newcommand*\Iac{\AC@starredfalse\protect\@Iac}%
\@Iaci 638 \WithSuffix\newcommand\Iaci*\AC@starredtrue\protect\@Iaci}%

639 \newcommand*\@iaci[1]{%
640 \ifcsname fn@#1@IL\endcsname
641 \ifAC@dua
642 \@firstupper@maybe{\csname fn@#1@IL\endcsname}%
643 \else
644 \expandafter\ifx\csname AC@\AC@prefix#1\endcsname\AC@used%
645 \csname fn@#1@IS\endcsname%
646 \else
647 \@firstupper@maybe{\csname fn@#1@IL\endcsname}%
648 \fi

```

```

649   \fi
650   \else
651     a%
652   \fi
653 }
654 \newcommand*{\@iac}[2][\AC@linebreakpenalty]{%
655   \@iaci{#2} \ifAC@starred\ac*{#1}{#2}\else\ac{#1}{#2}\fi%
656 }
657 \newcommand*{\@Iac}[2][\AC@linebreakpenalty]{%
658   \@firstupper{\@iaci{#2}}\space%
659   \ifAC@starred\ac*{#1}{#2}\else\ac{#1}{#2}\fi%
660 }

```

`\acsp` The user macro `\acsp` prints the plural short form of the acronym. This is the `\acspa` acronym itself or the *short name*, if the optional argument is given in the definition of the acronym plus an ‘s’.

```

661 \newcommand*{\acsp}{\AC@starredfalse\protect\acspa}%
662 \WithSuffix\newcommand\acsp*{\AC@starredtrue\protect\acspa}%
663 \newcommand*{\acspa}[1]{%
664   \texorpdfstring{\protect\@acsp{#1}}{\AC@acsp{#1}}}
665 \newcommand*{\@acsp}[1]{%
666   \acsfont{\AC@placelabel@part{#1}\AC@acsp{#1}}%
667   \ifAC@starred\else\AC@logged{#1}\fi}

```

`\aclp` The user macro `\aclp` prints the plural full name of the acronym.

```

\@aclp 668 \newcommand*{\aclp}{\AC@starredfalse\protect\@aclp}%
\Aclp 669 \WithSuffix\newcommand\aclp*{\AC@starredtrue\protect\@aclp}%
\@Aclp 670 \newcommand*{\Aclp}{\AC@starredfalse\protect\@Aclp}%
671 \WithSuffix\newcommand\Aclp*{\AC@starredtrue\protect\@Aclp}%
672 \newcommand*{\@aclp}[1]{%
673   \AC@placelabel@part{#1}\AC@aclp{#1}%
674   \ifAC@starred\else\AC@logged{#1}\fi}
675 \newcommand*{\@Aclp}[1]{%
676   \AC@placelabel@part{#1}\AC@Aclp{#1}%
677   \ifAC@starred\else\AC@logged{#1}\fi}

```

`\acfp` The user macro `\acfp` always prints the plural full name with the plural of the `\acfpa` acronym. The format depends on `\acffont` and `\acfsfont`, and on the option `\@acfp` footnote handled below.

`\Acfp` The option `footnote` leads to a redefinition of `\acfp`, making the full name `\Acfpa` appear as a footnote. There is then no need for `\acffont` and `\acfsfont`. If the option `footnote` is not specified, the optional variable determines the penalty for a line break.

```

678 \newcommand*{\acfp}{\AC@starredfalse\protect\acfpa}%
679 \WithSuffix\newcommand\acfp*{\AC@starredtrue\protect\acfpa}%
680 \newcommand*{\Acfp}{\AC@starredfalse\protect\Acfpa}%
681 \WithSuffix\newcommand\Acfp*{\AC@starredtrue\protect\Acfpa}%
682 \newcommand*{\acfpa}[2][\AC@linebreakpenalty]{%
683   \texorpdfstring{\protect\@acfp{#1}{#2}}{\AC@aclp{#2} (\AC@acsp{#2})}}

```



```

684 \newcommand*{\Acfpa}[2][\AC@linebreakpenalty]{%
685   \texorpdfstring{\protect\@Acfp[#1]{#2}}{\AC@Aclp{#2} (\AC@acsp{#2})}}
686 \newcommand*{\@acfp}[2][\AC@linebreakpenalty]{%
687   \ifAC@footnote
688     \acsfont{\AC@acsp{#2}}%
689     \footnote{\AC@placelabel{#2}\AC@aclp{#2}{}}%
690   \else
691     \acffont{%
692       \AC@placelabel{#2}\AC@aclp{#2}%
693       \nolinebreak[#1] %
694       \acsfont{(\acsfont{\AC@acsp{#2}})}%
695     }%
696   \fi
697   \ifAC@starred\else\AC@logged{#2}\fi}
698 \newcommand*{\@Acfp}[2][\AC@linebreakpenalty]{%
699   \ifAC@footnote
700     \acsfont{\AC@acsp{#2}}%
701     \footnote{\AC@placelabel{#2}\AC@Aclp{#2}{}}%
702   \else
703     \acffont{%
704       \AC@placelabel{#2}\AC@Aclp{#2}%
705       \nolinebreak[#1] %
706       \acsfont{(\acsfont{\AC@acsp{#2}})}%
707     }%
708   \fi
709   \ifAC@starred\else\AC@logged{#2}\fi}

\acp The first time an acronym is accessed Full Names (FNs) is printed. The next time
\@acp just (FNs).The optional variable is being passed to \acfp, in case it is used.
\Acp 710 \newcommand*{\acp}{\AC@starredfalse\protect\@acp}%
\@Acp 711 \WithSuffix\newcommand\acp*{\AC@starredtrue\protect\@acp}%
712 \newcommand*{\Acp}{\AC@starredfalse\protect\@Acp}%
713 \WithSuffix\newcommand\Acp*{\AC@starredtrue\protect\@Acp}%
714 \newcommand{\@acp}[2][\AC@linebreakpenalty]{%
715   \ifAC@dua
716     \ifAC@starred\aclp*{#2}\else\aclp{#2}\fi%
717   \else
718     \expandafter\ifx\csname AC@\AC@prefix#2\endcsname\AC@used
719     \ifAC@starred\acsp*{#2}\else\acsp{#2}\fi%
720   \else
721     \ifAC@starred\acfp*{#1}{#2}\else\acfp{#1}{#2}\fi%
722   \fi
723 \fi}
724 \newcommand{\@Acp}[2][\AC@linebreakpenalty]{%
725   \ifAC@dua
726     \ifAC@starred\Aclp*{#2}\else\Aclp{#2}\fi%
727   \else
728     \expandafter\ifx\csname AC@\AC@prefix#2\endcsname\AC@used
729     \ifAC@starred\Acsp*{#2}\else\Acsp{#2}\fi%
730   \else
731     \ifAC@starred\Acfp*{#1}{#2}\else\Acfp{#1}{#2}\fi%
732   \fi

```

733 \fi}

\acfi The Full Name is printed in italics and the abbreviated is printed in upshape. The

\acfia optional variable determines the penalty for a line break.

```
\Acfi 734 \newcommand*\acfi{\AC@starredfalse\protect\acfia}%
\Acfia 735 \WithSuffix\newcommand\acfi*\AC@starredtrue\protect\acfia}%
736 \newcommand*\Acfi{\AC@starredfalse\protect\Acfia}%
737 \WithSuffix\newcommand\Acfi*\AC@starredtrue\protect\Acfia}%
738 \newcommand{\acfia}[2][\AC@linebreakpenalty]{%
739 \texorpdfstring{\protect\@acfi[#1]{#2}}{\AC@acl{#2}} (#2)}}
740 \newcommand{\Acfia}[2][\AC@linebreakpenalty]{%
741 \texorpdfstring{\protect\@Acfi[#1]{#2}}{\AC@Acl{#2}} (#2)}}
742 \newcommand*\@acfi}[2][\AC@linebreakpenalty]{%
743 \acffont{%
744 \AC@placelabel{#2}{\itshape\AC@acl{#2}}}%
745 \nolinebreak[#1] %
746 \acsfont{\acsfont{\AC@acs{#2}}}}%
747 }%
748 \ifAC@starred\else\AC@logged{#2}\fi}
749 \newcommand*\@Acfi}[2][\AC@linebreakpenalty]{%
750 \acffont{%
751 \AC@placelabel{#2}{\itshape\AC@Acl{#2}}}%
752 \nolinebreak[#1] %
753 \acsfont{\acsfont{\AC@acs{#2}}}}%
754 }%
755 \ifAC@starred\else\AC@logged{#2}\fi}
```

\acfip The plural of the full name is printed in italics and the plural of the acronym is

\acfipa printed in upshape. The optional variable determines the penalty for a line break.

```
\Acfip 756 \newcommand*\acfip{\AC@starredfalse\protect\acfipa}%
\Acfipa 757 \WithSuffix\newcommand\acfip*\AC@starredtrue\protect\acfipa}%
758 \newcommand*\Acfip{\AC@starredfalse\protect\Acfipa}%
759 \WithSuffix\newcommand\Acfip*\AC@starredtrue\protect\Acfipa}%
760 \newcommand{\acfipa}[2][\AC@linebreakpenalty]{%
761 \texorpdfstring{\protect\@acfip[#1]{#2}}{\AC@aclp{#2}} (#2)}}
762 \newcommand{\Acfipa}[2][\AC@linebreakpenalty]{%
763 \texorpdfstring{\protect\@Acfip[#1]{#2}}{\AC@Aclp{#2}} (#2)}}
764 \newcommand*\@acfip}[2][\AC@linebreakpenalty]{%
765 \acffont{%
766 \AC@placelabel{#2}{\itshape\AC@aclp{#2}}}%
767 \nolinebreak[#1] %
768 \acsfont{\acsfont{\AC@acsp{#2}}}}%
769 }%
770 \ifAC@starred\else\AC@logged{#2}\fi}
771 \newcommand*\@Acfip}[2][\AC@linebreakpenalty]{%
772 \acffont{%
773 \AC@placelabel{#2}{\itshape\AC@Aclp{#2}}}%
774 \nolinebreak[#1] %
775 \acsfont{\acsfont{\AC@acsp{#2}}}}%
776 }%
777 \ifAC@starred\else\AC@logged{#2}\fi}
```

```

\acused Marks the acronym as used. Don't confuse this with \acronymused!
778 \newcommand{\acused}[1]{%
779 \global\expandafter\let\csname AC@AC@prefix#1\endcsname\AC@used%
780 \AC@addtoAC@clearlist{#1}}

\acsu Print the short form of the acronym and mark it as used.
\acsua 781 \newcommand*{\acsu}{\AC@starredfalse\protect\acsua}%
782 \WithSuffix\newcommand\acsu*{\AC@starredtrue\protect\acsua}%
783 \newcommand{\acsua}[1]{%
784   \ifAC@starred\acs*{#1}\else\acs{#1}\fi\acused{#1}}

\aclu Print the long form of the acronym and mark it as used.
\aclua 785 \newcommand*{\aclu}{\AC@starredfalse\protect\aclua}%
\Aclu 786 \WithSuffix\newcommand\aclu*{\AC@starredtrue\protect\aclua}%
\Aclua 787 \newcommand*{\Aclu}{\AC@starredfalse\protect\Aclua}%
788 \WithSuffix\newcommand\Aclu*{\AC@starredtrue\protect\Aclua}%
789 \newcommand{\aclua}[1]{%
790   \ifAC@starred\acl*{#1}\else\acl{#1}\fi\acused{#1}}
791 \newcommand{\Aclua}[1]{%
792   \ifAC@starred\Acl*{#1}\else\Acl{#1}\fi\acused{#1}}

793 \endinput
794 </acronym>

That's it.

```