

An Acronym Environment for L^AT_EX 2_ε*

Tobias Oetiker

2009/10/20

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{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.

`\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{acronym}`

*This file has version number v1.35, last revised 2009/10/20.

to access the acronym. It stands for “full acronym” and it always prints the full name and the acronym in brackets.

- `\acs` To get the short version of the acronym, use the command
- `\acs{<acronym>}`
- `\acl` Gives you the expanded acronym without even mentioning the acronym.
- `\acl{<acronym>}`
- `\acp` Works in the same way as `\ac`, but makes the short and/or long forms into plurals.
- `\acfp` Works in the same way as `\acf`, but makes the short and long forms into plurals.
- `\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.
- `\acfi` Prints the Full Name acronym (`\acl`) in italics and the abbreviated form (`\acs`) in upshaped form.
- `\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.
- Example: `\acl{lox}/\acl{lh2}` (`\acsu{lox}/\acsu{lh2}`)
- `\...*` 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*`, `\acs*`, `\acl*`, `\acf*`, `\acp*`, `\acsp*`, `\aclp*`, `\acfp*`, `\acfi*`, `\acsu*` and `\aclu*`.

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 `\acffont` (and `\acsfont`). The plural 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 TeX code for the acronym is limited by `\csname`. If the acronym requires problematic or complicated TeX 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]
```

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

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

In `printonlyused`-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}
```

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

```
\newacro
\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.

2.3.1 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

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

```

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

```

that allow one to define plural exceptions. The first one 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}-faktor}

```

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{ $\mathrm{H}_2\mathrm{O}$ }{H2O}) ...]
{The \acf{H2O} ...}

```

3 An example file

```
1 \acrotest
2 \documentclass{article}
3 \usepackage{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 \subsection{Some testing fundamentals}
44 When testing \acp{IC}, one typically wants to identify functional
45 blocks to be tested separately. The latter are commonly indicated as
46 \acp{BUT}. To test a \ac{BUT} requires defining a testing strategy\dots
47
48 \section{Acronyms}
```

```

49 \begin{acronym}[TDMA]
50 \acro{CDMA}{Code Division Multiple Access}
51 \acro{GSM}{Global System for Mobile communication}
52 \acro{NA}[\ensuremath{N_{\mathrm{A}}}]
53     {Number of Avogadro\acroextra{ (see \S\ref{Chem})}}
54 \acro{NAD+}[NAD\textsuperscript{+}]{Nicotinamide Adenine Dinucleotide}
55 \acro{NUA}{Not Used Acronym}
56 \acro{TDMA}{Time Division Multiple Access}
57 \acro{UA}{Used Acronym}
58 \acro{lox}[\ensuremath{LOX}]{Liquid Oxygen}%
59 \acro{lh2}[\ensuremath{LH_2}]{Liquid Hydrogen}%
60 \acro{IC}{Integrated Circuit}%
61 \acro{BUT}{Block Under Test}%
62 \acrodefplural{BUT}{Blocks Under Test}%
63 \end{acronym}
64
65 \end{document}
66 </acrotest>

```

4 The implementation

67 `\acronym`)

4.1 Identification

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

```
68 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
69 \ProvidesPackage{acronym}[2009/10/20
70                               v1.35
71                               Support for acronyms (Tobias Oetiker)]
72 \RequirePackage{suffix}
```

4.2 Options

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

```
73 \newif\ifAC@footnote
74 \AC@footnotefalse
75 \DeclareOption{footnote}{\AC@footnotetrue}
```

`\ifAC@nohyperlinks` If `hyperref` is loaded, all acronyms will link to their glossary entry. With the option `nohyperlinks` these links can be suppressed.

```
76 \newif\ifAC@nohyperlinks
77 \AC@nohyperlinksfalse
78 \DeclareOption{nohyperlinks}{\AC@nohyperlinkstrue}
```

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

```
79 \newif\ifAC@printonlyused
80 \AC@printonlyusedfalse
81 \DeclareOption{printonlyused}{\AC@printonlyusedtrue}
```

`\ifAC@withpage` A marker which tells us to print page numbers.

```
82 \newif\ifAC@withpage
83 \AC@withpagefalse
84 \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 variable.

```
85 \newif\ifAC@smaller
86 \AC@smallerfalse
87 \DeclareOption{smaller}{\AC@smallertrue}
```


`\ifAC@dua` The option `dua` stands for “don’t use acronyms”. It leads to a redefinition of `\ac` and `\acp`, making the full name appear all the time and suppressing all acronyms but the explicitly requested by `\acf` or `\acfp`.

```
88 \newif\ifAC@dua
89 \AC@duaafalse
90 \DeclareOption{dua}{\AC@duatruer}
```

`\ifAC@nolist` The option `nolist` stands for “don’t write the list of acronyms”.

```
91 \newif\ifAC@nolist
92 \AC@nolistfalse
93 \DeclareOption{nolist}{\AC@nolisttrue}
```

Now we process the options.

```
94 \ProcessOptions\relax
```

4.3 Setup macros

`\acsfont` The appearance of the output of the commands `\acs` and `\acf` is partially controlled by `\acsfont`, `\acffont`, and `\acfsfont`. By default, they do nothing 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.

```
95 \ifAC@smaller
96   \RequirePackage{relsize}
97   \newcommand*{\acsfont}[1]{\textsmaller{#1}}
98 \else
99   \newcommand*{\acsfont}[1]{#1}
100 \fi
101 \newcommand*{\acffont}[1]{#1}
102 \newcommand*{\acfsfont}[1]{#1}
```

4.4 Hyperlinks and PDF support

`\AC@hyperlink` Define dummy hyperlink commands
`\AC@hypertarget`

```
103 \def\AC@hyperlink#1#2{#2}
104 \def\AC@hypertarget#1#2{#2}
105 \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.

```
106 \ifAC@nohyperlinks
107 \else
108   \AtBeginDocument{%
109     \ifpackageloaded{hyperref}
110       {\let\AC@hyperlink=\hyperlink
111        \newcommand*\AC@raisedhypertarget[2]{%
```

```

112         \Hy@raisedlink{\hypertarget{#1}{}}#2}%
113     \let\AC@hypertarget=\AC@raisedhypertarget
114     \def\AC@phantomsection{%
115         \Hy@GlobalStepCount\Hy@linkcounter
116         \edef\@currentHref{section*.\the \Hy@linkcounter}%
117         \Hy@raisedlink{%
118             \hyper@anchorstart{\@currentHref}\hyper@anchorend
119         }%
120     }%
121     }{}%
122 \fi

```

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

```

123 \AtBeginDocument{%
124     \providecommand\texorpdfstring[2]{#1}%
125     \providecommand\pdfstringdefDisableCommands[1]{}%
126     \pdfstringdefDisableCommands{%
127         \csname AC@starredfalse\endcsname
128         \csname AC@footnotefalse\endcsname
129         \let\AC@hyperlink\@secondoftwo
130         \let\acsfont\relax
131         \let\acffont\relax
132         \let\acfsfont\relax
133         \let\acused\relax
134         \let\null\relax
135         \def\AChy@call#1#2{%
136             \ifx*#1\@empty
137                 \expandafter #2%
138             \else
139                 #2{#1}%
140             \fi
141         }%
142         \def\acs#1{\AChy@call{#1}\AC@acs}%
143         \def\acl#1{\AChy@call{#1}\AC@acl}%
144         \def\acf#1{\AChy@call{#1}\AChy@acf}%
145         \def\ac#1{\AChy@call{#1}\AC@ac}%
146         \def\acsp#1{\AChy@call{#1}\AC@acsp}%
147         \def\aclp#1{\AChy@call{#1}\AC@aclp}%
148         \def\acfp#1{\AChy@call{#1}\AChy@acfp}%
149         \def\acp#1{\AChy@call{#1}\AC@acp}%
150         \def\acfi#1{\AChy@call{#1}\AChy@acf}%
151         \let\acsu\acs
152         \let\aclu\acl
153         \def\AChy@acf#1{\AC@acl{#1} (\AC@acs{#1})}%
154         \def\AChy@acfp#1{\AC@aclp{#1} (\AC@acsp{#1})}%
155     }%
156 }

```

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`

```
157 \newtoks\AC@clearlist
```

`\AC@addtoAC@clearlist` Adds acronyms to the clear list

```
158 \newcommand*\AC@addtoAC@clearlist[1]{%
159   \global\AC@clearlist\expandafter{\the\AC@clearlist\AC@reset{#1}}%
160 }
```

`\acresetall` This macro resets the `ac@FN` - tag of each acronym, therefore `\ac` will use FullName (FN) next time it is called

`\AC@reset`

```
161 \newcommand*\acresetall{\the\AC@clearlist\AC@clearlist={}}
162 \def\AC@reset#1{%
163   \global\expandafter\let\csname ac@#1\endcsname\relax
164 }
```

`\AC@used` We also need a markers for 'used'.

```
165 \newcommand*\AC@used{@<>@<>@}
```

`\AC@populated` An on/off flag to note if any acronyms were logged. This is needed for the first run with `printonlyused` option, because the acronym list are then empty, resulting in a "missing item" error.

```
166 \newcommand{\AC@populated}{}
```

`\AC@logged` Log the usage by writing the `\acronymused` to the `aux` file and by reading it back 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.

`\acronymused`

This methodology is needed when the list of acronyms is in the front matter of the document.

```
167 \newcommand*\AC@logged}[1]{%
168   \acronymused{#1}% mark it as used in the current run too
169   \@bsphack
170   \protected@write\@auxout{}\string\acronymused{#1}%
171   \@esphack}
```

Keep it out of bookmarks.

```
172 \AtBeginDocument{%
173   \pdfstringdefDisableCommands{%
174     \let\AC@logged@gobble
175   }%
176 }
```

Flag the acronym at the beginning of the document as used (called by the `aux` file).

```

177 \newcommand*\acronymused}[1]{%
178   \expandafter\ifx\csname acused@#1\endcsname\AC@used
179   \relax
180   \else
181     \global\expandafter\let\csname acused@#1\endcsname\AC@used
182     \global\let\AC@populated\AC@used
183   \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>`.

```

184 \newcommand*\newacro[1]{%
185   \@ifnextchar[{\AC@newacro{#1}}{\AC@newacro{#1}[#1]}]
186 \newcommand\AC@newacro{}
187 \def\AC@newacro#1[#2]#3{%
188   \expandafter\gdef\csname fn@#1\endcsname{#2}#3}%
189   }

```

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

```

\AC@acrodef 190 \newcommand*\acrodef[1]{%
191   \@ifnextchar[{\AC@acrodef{#1}}{\AC@acrodef{#1}[#1]}]
192 \newcommand\AC@acrodef{}
193 \def\AC@acrodef#1[#2]#3{%
194   \@bsphack
195   \protected@write\@auxout{\string\newacro{#1}[#2]#3}%
196   \@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.

```

197 \def\bflabel#1{{\textbf{\textsf{#1}}\hfill}}
198 \newenvironment{AC@deflist}[1]%
199   {\ifAC@nolist%
200    \else%
201    \raggedright\begin{list}{}%

```

```

202             {\settoheight{\labelwidth}{\textbf{\textsf{#1}}}%
203             \setlength{\leftmargin}{\labelwidth}%
204             \addtolength{\leftmargin}{\labelsep}%
205             \renewcommand{\makeatlabel}{\bflabel}}%
206     \fi}%
207     {\ifAC@nolist%
208     \else%
209     \end{list}%
210     \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.

```

211 \newcommand{\acroextra}[1]{}

```

\acro Acronyms can be defined with the user command `\acro` in side the `acronym` environment.

```

212 \newenvironment{acronym}[1][1]{%
213   \providecommand*{\acro}{\AC@acro}%
214   \providecommand*{\acroplural}{\AC@acroplural}%
215   \long\def\acroextra##1{##1}%
216   \def\@tempa{1}\def\@tempb{#1}%
217   \ifx\@tempa\@tempb%
218     \global\expandafter\let\csname ac@des@mark\endcsname\AC@used%
219     \ifAC@nolist%
220     \else%
221     \begin{description}%
222     \fi%
223   \else%
224     \begin{AC@deflist}{#1}%
225     \fi%
226   }%
227   {%
228   \ifx\AC@populated\AC@used\else%
229     \ifAC@nolist%
230     \else%
231     \item[]\relax%
232     \fi%
233   \fi%
234   \expandafter\ifx\csname ac@des@mark\endcsname\AC@used%
235     \ifAC@nolist%
236     \else%

```

```

237         \end{description}%
238     \fi%
239 \else%
240     \end{AC@deflist}%
241 \fi}%

\AC@acro
\AC@@acro
242 \newcommand*\AC@acro[1]{%
243     \@ifnextchar[{\AC@@acro{#1}}{\AC@@acro{#1}[#1]}}
244 \newcommand\AC@@acro{}
245 \def\AC@@acro#1[#2]#3{%
246     \ifAC@nolist%
247     \else%
248     \ifAC@printonlyused%
249         \expandafter\ifx\csname acused@#1\endcsname\AC@used%
250             \item[\protect\AC@hypertarget{#1}{\acsfont{#2}}] #3%
251             \ifAC@withpage%
252                 \expandafter\ifx\csname r@acro:#1\endcsname\relax%
253                     \PackageInfo{acronym}{%
254                         Acronym #1 used in text but not spelled out in
255                         full in text}%
256                 \else%
257                     \dotfill\pageref{acro:#1}%
258                 \fi\%
259             \fi%
260         \fi%
261     \else%
262         \item[\protect\AC@hypertarget{#1}{\acsfont{#2}}] #3%
263     \fi%
264 \fi%
265 \begingroup
266     \def\acroextra##1{}%
267     \@bsphack
268     \protected@write\@auxout{%
269         {\string\newacro{#1}{\string\AC@hyperlink{#1}{#2}}{#3}}%
270     \@esphack
271 \endgroup}

```

4.6.1 Non standard or foreign plural forms

```

\newacroplural Sets up a non standard plural form for a given acronym.
\AC@newacroplurali
\AC@newacropluralii
272 \newcommand*\newacroplural[1]{%
273     \@ifnextchar[%
274         {\AC@newacroplurali{#1}}{\AC@newacropluralii{#1}}%
275     }
276 \newcommand\AC@newacroplurali{}
277 \def\AC@newacroplurali#1[#2]#3{%
278     \expandafter\gdef\csname fn@#1@PS\endcsname{#2}%
279     \expandafter\gdef\csname fn@#1@PL\endcsname{#3}%

```

```

280 }
281 \newcommand\AC@newacropluralii[2]{%
282   \expandafter\gdef\csname fn@#1@PL\endcsname{#2}%
283 }

```

\acrodefplural Same as above, storing content in aux file.

```

\AC@acrodefplurali 284 \newcommand*\acrodefplural[1]{%
\AC@acrodefpluralii 285   \ifnextchar[%
286     {\AC@acrodefplurali{#1}}{\AC@acrodefpluralii{#1}}%
287 }
288 \newcommand\AC@acrodefplurali{}
289 \def\AC@acrodefplurali#1[#2]#3{%
290   \@bsphack
291   \protected@write\@auxout{}\string\newacroplural{#1}[#2]{#3}%
292   \@esphack
293 }
294 \newcommand\AC@acrodefpluralii[2]{%
295   \@bsphack
296   \protected@write\@auxout{}\string\newacroplural{#1}{#2}%
297   \@esphack
298 }

```

\AC@acroplural Internal commands to set up a plural version of an acronym in the acronym environment.

```

\AC@acroplurali 299 \newcommand*\AC@acroplural[1]{%
\AC@acropluralii 300   \ifnextchar[%
301     {\AC@acroplurali{#1}}{\AC@acropluralii{#1}}%
302 }
303 \newcommand\AC@acroplurali{}
304 \def\AC@acroplurali#1[#2]#3{%
305   \@bsphack
306   \protected@write\@auxout{%
307     {\string\newacroplural{#1}[\string\AC@hyperlink{#1}{#2}]{#3}}%
308   \@esphack
309 }
310 \newcommand\AC@acropluralii[2]{
311   \@bsphack
312   \protected@write\@auxout{%
313     {\string\newacroplural{#1}[\string\AC@hyperlink{#1}{\AC@acs{#1}}]{#2}}%
314   \@esphack
315 }

```

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

```

\AC@acsp 316 \newcommand*\AC@aclp[1]{%
317   \ifcsname fn@#1@PL\endcsname
318   \csname fn@#1@PL\endcsname
319   \else
320   \AC@acl{#1}s%
321   \fi

```

```

322 }
323 \newcommand*\AC@acsp[1]{%
324   \ifcsname fn@#1@PS\endcsname
325   \csname fn@#1@PS\endcsname
326   \else
327   \AC@acs{#1}s%
328   \fi
329 }

```

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.

```
330 \newif\ifAC@starred
```

`\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>`.

```

331 \newcommand*\AC@get[3]{%
332   \ifx#1\relax
333     \PackageWarning{acronym}{Acronym ‘#3’ is not defined}%
334     \textbf{#3!}%
335   \else
336     \expandafter#2#1\null
337   \fi}

```

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

```

338 \newcommand*\AC@acs[1]{%
339   \expandafter\AC@get\csname fn@#1\endcsname\@firstoftwo{#1}}
340 \newcommand*\AC@acl[1]{%
341   \expandafter\AC@get\csname fn@#1\endcsname\@secondoftwo{#1}}

```

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

```

\@acs 342 \newcommand*\acs{\AC@starredfalse\protect\acs}%
343 \WithSuffix\newcommand\acs*\AC@starredtrue\protect\acs}%
344 \newcommand*\acs}[1]{%
345   \texorpdfstring{\protect\@acs{#1}}{#1}}
346 \newcommand*\@acs[1]{%
347   \acsfont{\AC@acs{#1}}%
348 %% having a footnote on acs sort of defetes the purpose
349 %%   \ifAC@footnote
350 %%     \footnote{\AC@acl{#1}}{}}%

```



```

351 %% \fi
352 \ifAC@starred\else\AC@logged{#1}\fi}

\acl The user macro \acl prints the full name of the acronym.
\@acl 353 \newcommand*\acl{\AC@starredfalse\protect\@acl}%
354 \WithSuffix\newcommand\acl*\AC@starredtrue\protect\@acl}%

355 \newcommand*\@acl[1]{%
356 \AC@acl{#1}%
357 \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)

```

358 \newcommand*\@verridelabel[1]{%
359 \@bsphack
360 \protected@write\@auxout{}\string\undonewlabel{#1}}%
361 \label{#1}%
362 \@overriddenmessage rs{#1}%
363 \@esphack
364 }%
365 \newcommand*\undonewlabel{\@und@newl@bel rs}%
366 \newcommand*\@und@newl@bel[3]{%
367 \@ifundefined{#1@#3}%
368 {%
369 \global\expandafter\let\csname#2@#3\endcsname\@nnil
370 }%
371 {%
372 \global\expandafter\let\csname#1@#3\endcsname\relax
373 }%
374 }%
375 \newcommand*\@overriddenmessage[3]{%
376 \expandafter\ifx\csname#2@#3\endcsname\@nnil
377 \expandafter\@firstoftwo
378 \else
379 \@ifundefined{#1@#3}%
380 {%
381 \@ifundefined{#2@#3}%
382 {\expandafter\@firstoftwo}%
383 {\expandafter\@secondoftwo}%
384 }%
385 {\expandafter\@secondoftwo}%
386 \fi
387 {%
388 \PackageInfo{acronym}{Label '#3' newly defined as it
389 shall be overridden^^Jalthough it is yet undefined}%

```

```

390   \global\expandafter\let\csname#2@#3\endcsname\empty
391 }%
392 {%
393   \PackageInfo{acronym}{Label '#3' overridden}%
394   \ifundefined{#2@#3}{%
395     \global\expandafter\let\csname#2@#3\endcsname\empty}{}%
396   \expandafter\g@addto@macro\csname#2@#3\endcsname{i}%
397 }%
398 }%
399 \newcommand*\ac@testdef[3]{%
400   \ifundefined{s@#2}\@secondoftwo\@firstofone
401   {%
402     \expandafter\ifx\csname s@#2\endcsname\empty
403     \expandafter\@firstofone
404     \else
405       \expandafter\xdef\csname s@#2\endcsname{%
406         \expandafter\expandafter
407         \expandafter\@gobble
408         \csname s@#2\endcsname
409       }%
410       \expandafter\@gobble
411     \fi
412   }%
413   {%
414     \@testdef{#1}{#2}{#3}%
415   }%
416 }%
417 \protected@write\@auxout{}{%
418   \string\reset@newl@bel
419 }%
420 \newcommand*\reset@newl@bel{%
421   \ifx\@newl@bel\@testdef
422     \let\@newl@bel\ac@testdef
423     \let\undoneNewLabel\@gobble
424   \fi
425 }%
426 \newcommand*\AC@placelabel[1]{%
427   \expandafter\ifx\csname ac@#1\endcsname\AC@used
428   \else
429     {\AC@phantomsection\@verridelabel{acro:#1}}%
430     \global\expandafter\let\csname ac@#1\endcsname\AC@used
431     \AC@addtoAC@clearlist{#1}%
432   \fi
433 }%

```

`\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 is marked as used by by `\gdefining` the `\ac@FN` to be `\AC@used` after its first use.

The option `footnote` leads to a redefinition of `\acf`, making the full name

appear as a footnote. There is then no need for `\acffont` and `\acsfont`.

```

434 \newcommand*{\acf}{\AC@starredfalse\protect\acfa}%
435 \WithSuffix\newcommand\acf*{\AC@starredtrue\protect\acfa}%
436 \newcommand*{\acfa}[1]{%
437   \texorpdfstring{\protect\acf{#1}}{\AC@acl{#1} (#1)}}
438 \newcommand*{\@acf}[1]{%
439   \ifAC@footnote
440     \acsfont{\AC@acs{#1}}%
441     \footnote{\AC@placelabel{#1}\AC@acl{#1}{}}%
442   \else
443     \acffont{%
444       \AC@placelabel{#1}\AC@acl{#1}%
445       \nolinebreak[3] %
446       \acsfont{(\acsfont{\AC@acs{#1}})}%
447     }%
448   \fi
449   \ifAC@starred\else\AC@logged{#1}\fi}

```

`\ac` The first time an acronym is accessed its Full Name (FN) is printed. The next time just (FN). When the footnote option is used the short form (FN) is always used.

```

450 \newcommand*{\ac}{\AC@starredfalse\protect\@ac}%
451 \WithSuffix\newcommand\ac*{\AC@starredtrue\protect\@ac}%
452 \newcommand*{\@ac}[1]{%
453   \ifAC@dua
454     \ifAC@starred\acl*{#1}\else\acl{#1}\fi%
455   \else
456     \expandafter\ifx\csname ac@#1\endcsname\AC@used%
457     \ifAC@starred\acs*{#1}\else\acs{#1}\fi%
458   \else
459     \ifAC@starred\acf*{#1}\else\acf{#1}\fi%
460   \fi
461 \fi}

```

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

```

462 \newcommand*{\acsp}{\AC@starredfalse\protect\acspa}%
463 \WithSuffix\newcommand\acsp*{\AC@starredtrue\protect\acspa}%
464 \newcommand*{\acspa}[1]{%
465   \texorpdfstring{\protect\acsp{#1}}{\AC@acsp{#1}}}
466 \newcommand*{\@acsp}[1]{%
467   \acsfont{\AC@acsp{#1}}%
468   \ifAC@starred\else\AC@logged{#1}\fi}

```

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

```

469 \newcommand*{\aclp}{\AC@starredfalse\protect\@aclp}%
470 \WithSuffix\newcommand\aclp*{\AC@starredtrue\protect\@aclp}%

```

```

471 \newcommand*{\@aclp}[1]{%
472   \AC@aclp{#1}%
473   \ifAC@starred\else\AC@logged{#1}\fi}

```

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

The option `footnote` leads to a redefinition of `\acfp`, making the full name appear as a footnote. There is then no need for `\acffont` and `\acsfont`.

```

474 \newcommand*{\acfp}{\AC@starredfalse\protect\acfp}%
475 \WithSuffix\newcommand\acfp*{\AC@starredtrue\protect\acfp}%
476 \newcommand*{\acfp}[1]{%
477   \texorpdfstring{\protect\@acfp{#1}}{\AC@aclp{#1} (\AC@acsp{#1})}}
478 \newcommand*{\@acfp}[1]{%
479   \ifAC@footnote
480     \acsfont{\AC@acsp{#1}}%
481     \footnote{\AC@placelabel{#1}\AC@aclp{#1}{}}%
482   \else
483     \acffont{%
484       \AC@placelabel{#1}\AC@aclp{#1}%
485       \nolinebreak[3] %
486       \acsfont{(\AC@acsp{#1})}%
487     }%
488   \fi
489   \ifAC@starred\else\AC@logged{#1}\fi}

```

`\acp` The first time an acronym is accessed Full Names (FNs) is printed. The next time `\@acp` just (FNs).

```

490 \newcommand*{\acp}{\AC@starredfalse\protect\@acp}%
491 \WithSuffix\newcommand\acp*{\AC@starredtrue\protect\@acp}%
492 \newcommand{\@acp}[1]{%
493   \ifAC@dua
494     \ifAC@starred\aclp*{#1}\else\aclp{#1}\fi%
495   \else
496     \expandafter\ifx\csname ac@#1\endcsname\AC@used
497     \ifAC@starred\acsp*{#1}\else\acsp{#1}\fi%
498   \else
499     \ifAC@starred\acfp*{#1}\else\acfp{#1}\fi%
500   \fi
501   \fi}

```

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

```

\acfia 502 \newcommand*{\acfi}{\AC@starredfalse\protect\acfia}%
503 \WithSuffix\newcommand\acfi*{\AC@starredtrue\protect\acfia}%
504 \newcommand{\acfia}[1]{%
505   {\itshape \AC@acl{#1} \nolinebreak[3]} (\ifAC@starred\acs*{#1}\else\acs{#1}\fi)}

```

```

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

\acsu Print the short form of the acronym and mark it as used.
\acsua 509 \newcommand*{\acsu}{\AC@starredfalse\protect\acsua}%
510 \WithSuffix\newcommand\acsu*{\AC@starredtrue\protect\acsua}%

511 \newcommand{\acsua}[1]{%
512   \ifAC@starred\acs*{#1}\else\acs{#1}\fi\acused{#1}}

\aclu Print the long form of the acronym and mark it as used.
\aclua 513 \newcommand*{\aclu}{\AC@starredfalse\protect\aclua}%
514 \WithSuffix\newcommand\aclu*{\AC@starredtrue\protect\aclua}%

515 \newcommand{\aclua}[1]{%
516   \ifAC@starred\acl*{#1}\else\acl{#1}\fi\acused{#1}}

517 \endinput
518 </acronym>

That's it.

```