An Acronym Environment for LaTeX 2$_\varepsilon$[*]

Tobias Oetiker

2009/01/25

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 v1.34, last revised 2009/01/25.

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 English plurals by adding an 's'.

\acfp   Works in the same way as `\acf`, but makes the short and long forms into English plurals by adding an 's'.

\acsp   Works in the same way as `\acs`, but makes the short form into an English plural by adding an 's'.

\aclp   Works in the same way as `\acl`, but makes the long form into an English plural by adding an 's'.

\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 `relsize` package is required.

\acsfont \acffont \acfsfont   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 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 `relsize` package:

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

2.3 Defining Acronyms

acronym   With the `acronym` environment you define all the acronyms in your document.

\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 complicate 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 "H_2O" and `\acl{H2O}` prints "water".

All acronym definitions, made by `\acro` or `\acrodef` are added to the `.aux` file. Therefore they are available from start-up in the next run.

\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 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}`

3

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}
```

2.4 Miscellaneous

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 the uses 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 (N_{χ})

- 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{$\mathrm{H_20}$}{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 \section{Acronyms}
44 \begin{acronym}[TDMA]
45  \acro{CDMA}{Code Division Multiple Access}
46  \acro{GSM}{Global System for Mobile communication}
47  \acro{NA}[\ensuremath{N_{\mathrm A}}]
48      {Number of Avogadro\acroextra{ (see \S\ref{Chem})}}
```

```
49  \acro{NAD+}[NAD\textsuperscript{+}]{Nicotinamide Adenine Dinucleotide}
50  \acro{NUA}{Not Used Acronym}
51  \acro{TDMA}{Time Division Multiple Access}
52  \acro{UA}{Used Acronym}
53  \acro{lox}[\ensuremath{LOX}]{Liquid Oxygen}%
54  \acro{lh2}[\ensuremath{LH_2}]{Liquid Hydrogen}%
55 \end{acronym}
56
57 \end{document}
58 ⟨/acrotest⟩
```

4   The implementation

59 ⟨∗acronym⟩

4.1   Identification

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

```
60 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
61 \ProvidesPackage{acronym}[2009/01/25
62                          v1.34
63                          Support for acronyms (Tobias Oetiker)]
64 \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.

```
65 \newif\ifAC@footnote
66 \AC@footnotefalse

67 \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.

```
68 \newif\ifAC@nohyperlinks
69 \AC@nohyperlinksfalse

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

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

```
71 \newif\ifAC@printonlyused
72 \AC@printonlyusedfalse

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

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

```
74 \newif\ifAC@withpage
75 \AC@withpagefalse

76 \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.

```
77 \newif\ifAC@smaller
78 \AC@smallerfalse
79 \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 explicity requested by \acf or \acfp.

```
80 \newif\ifAC@dua
81 \AC@duafalse
82 \DeclareOption{dua}{\AC@duatrue}
```

\ifAC@nolist The option nolist stands for "don't write the list of acronyms".

```
83 \newif\ifAC@nolist
84 \AC@nolistfalse
85 \DeclareOption{nolist}{\AC@nolisttrue}
```

Now we process the options.

```
86 \ProcessOptions\relax
```

4.3 Setup macros

\acsfont
\acffont
\acfsfont
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.

```
87 \ifAC@smaller
88   \RequirePackage{relsize}
89   \newcommand*{\acsfont}[1]{\textsmaller{#1}}
90 \else
91   \newcommand*{\acsfont}[1]{#1}
92 \fi
93 \newcommand*{\acffont}[1]{#1}
94 \newcommand*{\acfsfont}[1]{#1}
```

4.4 Hyperlinks and PDF support

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

```
95 \def\AC@hyperlink#1#2{#2}
96 \def\AC@hypertarget#1#2{#2}
97 \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.

```
98 \ifAC@nohyperlinks
99 \else
100   \AtBeginDocument{%
101     \@ifpackageloaded{hyperref}
102       {\let\AC@hyperlink=\hyperlink
103        \newcommand*\AC@raisedhypertarget[2]{%
```

```
104            \Hy@raisedlink{\hypertarget{#1}{}}#2%
105          \let\AC@hypertarget=\AC@raisedhypertarget
106          \def\AC@phantomsection{%
107            \Hy@GlobalStepCount\Hy@linkcounter
108            \edef\@currentHref{section*.\the \Hy@linkcounter}%
109            \Hy@raisedlink{%
110              \hyper@anchorstart{\@currentHref}\hyper@anchorend
111            }%
112          }%
113        }{}}%
114 \fi
```

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

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

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

```
149 \newtoks\AC@clearlist
```

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

```
150 \newcommand*\AC@addtoAC@clearlist[1]{%
151   \global\AC@clearlist\expandafter{\the\AC@clearlist\AC@reset{#1}}%
152 }
```

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

```
153 \newcommand*\acresetall{\the\AC@clearlist\AC@clearlist={}}
```

```
154 \def\AC@reset#1{%
155   \global\expandafter\let\csname ac@#1\endcsname\relax
156 }
```

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

```
157 \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.

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

\AC@logged \acronymused 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.

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

```
159 \newcommand*{\AC@logged}[1]{%
160   \acronymused{#1}% mark it as used in the current run too
161   \@bsphack
162   \protected@write\@auxout{}{\string\acronymused{#1}}%
163   \@esphack}
```

Keep it out of bookmarks.

```
164 \AtBeginDocument{%
165   \pdfstringdefDisableCommands{%
166     \let\AC@logged\@gobble
167   }%
168 }
```

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

```
169 \newcommand*{\acronymused}[1]{%
170     \expandafter\ifx\csname acused@#1\endcsname\AC@used
171         \relax
172     \else
173         \global\expandafter\let\csname acused@#1\endcsname\AC@used
174         \global\let\AC@populated\AC@used
175     \fi}
```

4.6   Defining acronyms

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

> `\acro{`⟨*acronym*⟩`}[`⟨*short name*⟩`]{`⟨*full name*⟩`}`

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

`\newacro`
`\AC@newacro`
The internal macro `\newacro` stores the ⟨*short name*⟩ and the ⟨*full name*⟩ in the command `\fn@<acronym>`.

```
176 \newcommand*\newacro[1]{%
177     \@ifnextchar[{\AC@newacro{#1}}{\AC@newacro{#1}[\AC@temp]}}
```

```
178 \newcommand\AC@newacro{}
179 \def\AC@newacro#1[#2]#3{%
180     \def\AC@temp{#1}%
181     \expandafter\gdef\csname fn@#1\endcsname{{#2}{#3}}%
182     }
```

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

```
183 \newcommand*\acrodef[1]{%
184     \@ifnextchar[{\AC@acrodef{#1}}{\AC@acrodef{#1}[\AC@temp]}}
```

```
185 \newcommand\AC@acrodef{}
186 \def\AC@acrodef#1[#2]#3{%
187     \def\AC@temp{#1}%
188     \@bsphack
189     \protected@write\@auxout{}{\string\newacro{#1}[#2]{#3}}%
190     \@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 enviroment is empty.

```
191 \def\bflabel#1{{\textbf{\textsf{#1}}\hfill}}
192 \newenvironment{AC@deflist}[1]%
193         {\ifAC@nolist%
```

```
194        \else%
195          \raggedright\begin{list}{}%
196              {\settowidth{\labelwidth}{\textbf{\textsf{#1}}}%
197              \setlength{\leftmargin}{\labelwidth}%
198              \addtolength{\leftmargin}{\labelsep}%
199              \renewcommand{\makelabel}{\bflabel}}%
200          \fi}%
201        {\ifAC@nolist%
202         \else%
203            \end{list}%
204         \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.

```
205 \newcommand{\acroextra}[1]{}
```

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

```
206 \newenvironment{acronym}[1][1]{%
207    \providecommand*{\acro}{\AC@acro}%
208    \long\def\acroextra##1{##1}%
209    \def\@tempa{1}\def\@tempb{#1}%
210    \ifx\@tempa\@tempb%
211       \global\expandafter\let\csname ac@des@mark\endcsname\AC@used%
212       \ifAC@nolist%
213       \else%
214          \begin{description}%
215       \fi%
216    \else%
217       \begin{AC@deflist}{#1}%
218    \fi%
219 }%
220 {%
221    \ifx\AC@populated\AC@used\else%
222       \ifAC@nolist%
223       \else%
224          \item[]\relax%
225       \fi%
226    \fi%
227    \expandafter\ifx\csname ac@des@mark\endcsname\AC@used%
228       \ifAC@nolist%
```

```
229       \else%
230          \end{description}%
231       \fi%
232    \else%
233       \end{AC@deflist}%
234    \fi}%
```

```
235 \newcommand*\AC@acro[1]{%
236    \@ifnextchar[{\AC@@acro{#1}}{\AC@@acro{#1}[\AC@temp]}}

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

4.7 Using acronyms

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.

```
266 \newif\ifAC@starred
```

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

13

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>`.

```
267 \newcommand*\AC@get[3]{%
268     \ifx#1\relax
269         \PackageWarning{acronym}{Acronym '#3' is not defined}%
270         \textbf{#3!}%
271     \else
272         \expandafter#2#1\null
273     \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>`.

```
274 \newcommand*\AC@acs[1]{%
275     \expandafter\AC@get\csname fn@#1\endcsname\@firstoftwo{#1}}
```

```
276 \newcommand*\AC@acl[1]{%
277     \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
\acsa by `\acsfont`.
\@acs
```
278 \newcommand*{\acs}{\AC@starredfalse\protect\acsa}%
279 \WithSuffix\newcommand\acs*{\AC@starredtrue\protect\acsa}%
```

```
280 \newcommand*{\acsa}[1]{%
281     \texorpdfstring{\protect\@acs{#1}}{#1}}
```

```
282 \newcommand*{\@acs}[1]{%
283     \acsfont{\AC@acs{#1}}%
284 %% having a footnote on acs sort of defetes the purpose
285 %%    \ifAC@footnote
286 %%        \footnote{\AC@acl{#1}{}}%
287 %%    \fi
288     \ifAC@starred\else\AC@logged{#1}\fi}
```

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

```
291 \newcommand*{\@acl}[1]{%
292     \AC@acl{#1}%
293     \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)

```
294 \newcommand*\@verridelabel[1]{%
```

```
295    \@bsphack
296    \protected@write\@auxout{}{\string\undonewlabel{#1}}%
297    \label{#1}%
298    \@overriddenmessage rs{#1}%
299    \@esphack
300  }%
301  \newcommand*\undonewlabel{\@und@newl@bel rs}%
302  \newcommand*\@und@newl@bel[3]{%
303    \@ifundefined{#1@#3}%
304    {%
305      \global\expandafter\let\csname#2@#3\endcsname\@nnil
306    }%
307    {%
308      \global\expandafter\let\csname#1@#3\endcsname\relax
309    }%
310  }%
311  \newcommand*\@overriddenmessage[3]{%
312    \expandafter\ifx\csname#2@#3\endcsname\@nnil
313      \expandafter\@firstoftwo
314    \else
315      \@ifundefined{#1@#3}%
316      {%
317        \@ifundefined{#2@#3}%
318        {\expandafter\@firstoftwo}%
319        {\expandafter\@secondoftwo}%
320      }%
321      {\expandafter\@secondoftwo}%
322    \fi
323    {%
324      \PackageInfo{acronym}{Label '#3' newly defined as it
325      shall be overridden^^Jalthough it is yet undefined}%
326      \global\expandafter\let\csname#2@#3\endcsname\empty
327    }%
328    {%
329      \PackageInfo{acronym}{Label '#3' overridden}%
330      \@ifundefined{#2@#3}{%
331        \global\expandafter\let\csname#2@#3\endcsname\empty}{}%
332      \expandafter\g@addto@macro\csname#2@#3\endcsname{i}%
333    }%
334  }%
335  \newcommand*\ac@testdef[3]{%
336    \@ifundefined{s@#2}\@secondoftwo\@firstofone
337    {%
338      \expandafter\ifx\csname s@#2\endcsname\empty
339        \expandafter\@firstofone
340      \else
341        \expandafter\xdef\csname s@#2\endcsname{%
342          \expandafter\expandafter
343          \expandafter\@gobble
344          \csname s@#2\endcsname
```

```
345        }%
346        \expandafter\@gobble
347      \fi
348    }%
349    {%
350      \@testdef{#1}{#2}{#3}%
351    }%
352 }%
353 \protected@write\@auxout{}{%
354    \string\reset@newl@bel
355 }%
356 \newcommand*\reset@newl@bel{%
357    \ifx\@newl@bel\@testdef
358      \let\@newl@bel\ac@testdef
359      \let\undonewlabel\@gobble
360    \fi
361 }%
362 \newcommand*\AC@placelabel[1]{%
363    \expandafter\ifx\csname ac@#1\endcsname\AC@used
364    \else
365      {\AC@phantomsection\@verridelabel{acro:#1}}%
366      \global\expandafter\let\csname ac@#1\endcsname\AC@used
367      \AC@addtoAC@clearlist{#1}%
368    \fi
369 }%
```

\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 \acfsfont.

```
370 \newcommand*{\acf}{\AC@starredfalse\protect\acfa}%
371 \WithSuffix\newcommand\acf*{\AC@starredtrue\protect\acfa}%

372 \newcommand*{\acfa}[1]{%
373    \texorpdfstring{\protect\@acf{#1}}{\AC@acl{#1} (#1)}}

374 \newcommand*{\@acf}[1]{%
375      \ifAC@footnote
376        \acsfont{\AC@acs{#1}}%
377        \footnote{\AC@placelabel{#1}\AC@acl{#1}{}}%
378      \else
379        \acffont{%
380          \AC@placelabel{#1}\AC@acl{#1}%
381          \nolinebreak[3] %
382          \acfsfont{(\acsfont{\AC@acs{#1}})}%
383        }%
384      \fi
385      \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.

```
386 \newcommand*{\ac}{\AC@starredfalse\protect\@ac}%
387 \WithSuffix\newcommand\ac*{\AC@starredtrue\protect\@ac}%

388 \newcommand{\@ac}[1]{%
389   \ifAC@dua
390     \ifAC@starred\acl*{#1}\else\acl{#1}\fi%
391   \else
392     \expandafter\ifx\csname ac@#1\endcsname\AC@used%
393     \ifAC@starred\acs*{#1}\else\acs{#1}\fi%
394   \else
395     \ifAC@starred\acf*{#1}\else\acf{#1}\fi%
396   \fi
397 \fi}
```

\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 defi-
\@acsp   nition of the acronym plus an 's'.

```
398 \newcommand*{\acsp}{\AC@starredfalse\protect\acspa}%
399 \WithSuffix\newcommand\acsp*{\AC@starredtrue\protect\acspa}%

400 \newcommand*{\acspa}[1]{%
401   \texorpdfstring{\protect\@acsp{#1}}{#1s}}

402 \newcommand*{\@acsp}[1]{%
403   \acsfont{\AC@acs{#1}s}%
404   \ifAC@starred\else\AC@logged{#1}\fi}
```

\aclp   The user macro `\aclp` prints the plural full name of the acronym.
\@aclp
```
405 \newcommand*{\aclp}{\AC@starredfalse\protect\@aclp}%
406 \WithSuffix\newcommand\aclp*{\AC@starredtrue\protect\@aclp}%

407 \newcommand*{\@aclp}[1]{%
408   \AC@acl{#1}s%
409   \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.
    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 `\acfsfont`.

```
410 \newcommand*{\acfp}{\AC@starredfalse\protect\acfpa}%
411 \WithSuffix\newcommand\acfp*{\AC@starredtrue\protect\acfpa}%

412 \newcommand*{\acfpa}[1]{%
413   \texorpdfstring{\protect\@acfp{#1}}{\AC@acl{#1}s (#1s)}}

414 \newcommand*{\@acfp}[1]{%
415   \ifAC@footnote
416     \acsfont{\AC@acs{#1}s}%
```

```
417        \footnote{\AC@placelabel{#1}\AC@acl{#1}s{}}}%
418    \else
419        \acffont{%
420            \AC@placelabel{#1}\AC@acl{#1}s%
421            \nolinebreak[3] %
422            \acfsfont{(\acsfont{\AC@acs{#1}s})}%
423            }%
424    \fi
425    \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).

```
426 \newcommand*{\acp}{\AC@starredfalse\protect\@acp}%
427 \WithSuffix\newcommand\acp*{\AC@starredtrue\protect\@acp}%

428 \newcommand{\@acp}[1]{%
429    \ifAC@dua
430        \ifAC@starred\aclp*{#1}\else\aclp{#1}\fi%
431    \else
432    \expandafter\ifx\csname ac@#1\endcsname\AC@used
433        \ifAC@starred\acsp*{#1}\else\acsp{#1}\fi%
434    \else
435        \ifAC@starred\acfp*{#1}\else\acfp{#1}\fi%
436    \fi
437    \fi}
```

\acfi   The Full Name is printed in italics and the abbreviated is printed in upshape.
\acfia
```
438 \newcommand*{\acfi}{\AC@starredfalse\protect\acfia}%
439 \WithSuffix\newcommand\acfi*{\AC@starredtrue\protect\acfia}%

440 \newcommand{\acfia}[1]{%
441    {\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!
```
442 \newcommand{\acused}[1]{%
443 \global\expandafter\let\csname ac@#1\endcsname\AC@used%
444 \AC@addtoAC@clearlist{#1}}
```

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

447 \newcommand{\acsua}[1]{%
448    \ifAC@starred\acs*{#1}\else\acs{#1}\fi\acused{#1}}
```

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

451 \newcommand{\aclua}[1]{%
452    \ifAC@starred\acl*{#1}\else\acl{#1}\fi\acused{#1}}
```

```
453 \endinput
454 ⟨/acronym⟩
```

That's it.