

An Acronym Environment for L^AT_EX 2_ε*

Tobias Oetiker

2004/11/10

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 v1.17, last revised 2004/11/10.

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’.

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.

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 T_EX code for the acronym is limited by `\csname`. If the acronym

requires problematic or complicate 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 `\ac1{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}
```

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{ $\chi$ }-faktor}
```

which will then give

```
pdf bookmark: \acf{Nx} → X-factor (Nx)
text:         \acf{Nx} →  $\chi$ -faktor ( $N_{\chi}$ )
```

- For acronyms in sectional headers, the file must be PDF^LA^TE^X'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[printonlyused]{acronym}
4 \begin{document}
5
6 \section{Intro}
7 In the early nineties, \acs{GSM} was deployed in many European
8 countries. \ac{GSM} offered for the first time international
9 roaming for mobile subscribers. The \acs{GSM}'s use of \ac{TDMA} as
10 its communication standard was debated at length. And every now
11 and then there are big discussion whether \ac{CDMA} should have
12 been chosen over \ac{TDMA}.
13
14 \section{Furthermore}
15 \acresetall
16 The reader could have forgotten all the nice acronyms, so we repeat the
17 meaning again.
18
19 If you want to know more about \acf{GSM}, \acf{TDMA}, \acf{CDMA}
20 and other acronyms, just read a book about mobile communication. Just
21 to mention it: There is another \ac{UA}, just for testing purposes!
22
23 \subsection{Some chemistry and physics}
24 \label{Chem}
25 \ac{NAD+} is a major electron acceptor in the oxidation
26 of fuel molecules. The reactive part of \ac{NAD+} is its nictinamide
27 ring, a pyridine derivate.
28
29 One mol consists of \acs{NA} atoms or molecules. There is a relation
30 between the constant of Boltzmann and the \acl{NA}:
31 \begin{equation}
32 k = R/\acs{NA}
33 \end{equation}
34
35 \section{Acronyms}
36 \begin{acronym}[TDMA]
37 \acro{CDMA}{Code Division Multiple Access}
38 \acro{GSM}{Global System for Mobile communication}
39 \acro{NA}[\ensuremath{N_{\mathrm{A}}}]
40 {Number of Avogadro\acroextra{ (see \S\ref{Chem})}}
41 \acro{NAD+}[NAD\textsuperscript{+}]{Nicotinamide Adenine Dinucleotide}
42 \acro{NUA}{Not Used Acronym}
43 \acro{TDMA}{Time Division Multiple Access}
44 \acro{UA}{Used Acronym}
45 \end{acronym}
46
47 \end{document}
48 </acrotest>
```

4 The implementation

49 `{*acronym}`

4.1 Identification

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

```
50 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
51 \ProvidesPackage{acronym}[2004/11/10
52                               v1.17
53                               Support for acronyms (Tobias Oetiker)]
```

4.2 Options

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

```
54 \newif\ifAC@footnote
55 \AC@footnotefalse
56 \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.

```
57 \newif\ifAC@nohyperlinks
58 \AC@nohyperlinksfalse
59 \DeclareOption{nohyperlinks}{\AC@nohyperlinkstrue}
```

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

```
60 \newif\ifAC@printonlyused
61 \AC@printonlyusedfalse
62 \DeclareOption{printonlyused}{\AC@printonlyusedtrue}
```

`\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.

```
63 \newif\ifAC@smaller
64 \AC@smallerfalse
65 \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`.

```
66 \newif\ifAC@dua
67 \AC@duaafalse
68 \DeclareOption{dua}{\AC@duatruetrue}
```

Now we process the options.

```
69 \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
`\acffont` except when the `smaller` option is loaded.
`\acfsfont`

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.

```
70 \ifAC@smaller
71   \RequirePackage{relsize}
72   \newcommand*{\acsfont}[1]{\textsmaller{#1}}
73 \else
74   \newcommand*{\acsfont}[1]{#1}
75 \fi
76 \newcommand*{\acffont}[1]{#1}
77 \newcommand*{\acfsfont}[1]{#1}
```

4.4 Hyperlinks and PDF support

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

```
78 \def\AC@hyperlink#1#2{#2}
79 \def\AC@hypertarget#1#2{#2}
```

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

```
80 \ifAC@nohyperlinks
81 \else
82   \AtBeginDocument{%
83     \ifpackageloaded{hyperref}
84       {\let\AC@hyperlink=\hyperlink
85        \newcommand*{\AC@raisedhypertarget[2]{%
86          \Hy@raisedlink{\hypertarget{#1}{}}#2}%
87         \let\AC@hypertarget=\AC@raisedhypertarget}{}}
88 \fi
```

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

```
89 \AtBeginDocument{%
90   \providecommand\texorpdfstring[2]{#1}%
91   \providecommand\pdfstringdefDisableCommands[1]{}%
92 }
```

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

Clearlist

```
93 \newtoks\clearlist

\AC@addtoclearlist Adds acronyms to the clear list
94 \newcommand*\AC@addtoclearlist[1]{%
95   \global\clearlist\expandafter{\the\clearlist\AC@reset{#1}}%
96 }

\acresetall This macro resets the ac@FN - tag of each acronym, therefore \ac will use FullName
\AC@reset (FN) next time it is called
97 \newcommand*\acresetall{\the\clearlist\clearlist={}}
98 \def\AC@reset#1{%
99   \global\expandafter\let\csname ac@#1\endcsname\relax}

\AC@used We also need a markers for 'used'.
100 \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.
101 \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.
102 \newcommand*\AC@logged[1]{%
103   \@bsphack
104   \protected@write\@auxout{}\string\acronymused{#1}}%
105   \@esphack}
Keep it out of bookmarks.
106 \AtBeginDocument{%
107   \pdfstringdefDisableCommands{%
108     \let\AC@logged@gobble
109   }%
110 }
Flag the acronym at the beginning of the document as used (called by the aux
file).
111 \newcommand*\acronymused[1]{%
112   \expandafter\ifx\csname acused@#1\endcsname\AC@used
113     \relax
114   \else
115     \global\expandafter\let\csname acused@#1\endcsname\AC@used
116     \global\let\AC@populated\AC@used
117   \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` The internal macro `\newacro` stores the `<short name>` and the `<full name>` in the command `\fn@<acronym>`.

```
\AC@newacro 118 \newcommand*\newacro[1]{%
119   \ifnextchar[{\AC@newacro{#1}}{\AC@newacro{#1}[\AC@temp]}}
120 \newcommand\AC@newacro{}
121 \def\AC@newacro#1[#2]#3{%
122   \def\AC@temp{#1}%
123   \expandafter\gdef\csname fn@#1\endcsname{{#2}{#3}}%
124 }
```

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

```
\AC@acrodef 125 \newcommand*\acrodef[1]{%
126   \ifnextchar[{\AC@acrodef{#1}}{\AC@acrodef{#1}[\AC@temp]}}
127 \newcommand\AC@acrodef{}
128 \def\AC@acrodef#1[#2]#3{%
129   \def\AC@temp{#1}%
130   \@bsphack
131   \protected@write\@auxout{}{\string\newacro{#1}[#2]{#3}}%
132   \@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.

```
133 \def\bflabel#1{{\textbf{\textsf{#1}}\hfill}}
134 \newenvironment{AC@deflist}[1]%
135   {\raggedright\begin{list}{}%
136     {\settowidth{\labelwidth}{\textbf{\textsf{#1}}}%
137     \setlength{\leftmargin}{\labelwidth}%
138     \addtolength{\leftmargin}{\labelsep}%
139     \renewcommand{\makelabel}{\bflabel}}}%
140   {\end{list}}
```

`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.

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

```

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

```
142 \newenvironment{acronym}[1][1]{%
143   \providecommand*\acro{\AC@acro}%
144   \long\def\acroextra##1{##1}%
145   \ifx1#1
146     \global\expandafter\let\csname ac@des@mark\endcsname\AC@used
147     \begin{description}%
148   \else
149     \begin{AC@deflist}{#1}%
150   \fi%
151 }%
152 {%
153   \ifx\AC@populated\AC@used\else
154     \item[]\relax
155   \fi
156   \expandafter\ifx\csname ac@des@mark\endcsname\AC@used
157     \end{description}%
158   \else
159     \end{AC@deflist}%
160   \fi}

```

`\AC@acro`

```
\AC@@acro
161 \newcommand*\AC@acro[1]{%
162   \@ifnextchar[{\AC@@acro{#1}}{\AC@@acro{#1}[\AC@temp]{}]}
163 \newcommand\AC@@acro{}
164 \def\AC@@acro#1[#2]#3{%
165   \def\AC@temp{#1}%
166   \ifAC@printonlyused
167     \expandafter\ifx\csname acused@#1\endcsname\AC@used%
168     \item[\protect\AC@hypertarget{#1}{\acsfont{#2}}] #3%
169     \fi%
170   \else%
171     \item[\protect\AC@hypertarget{#1}{\acsfont{#2}}] #3%
172   \fi
173   \begingroup
174     \def\acroextra##1{}%
175     \@bsphack
176     \protected@write\@auxout{%
177       {\string\newacro{#1}[\string\AC@hyperlink{#1}{#2}]{#3}}%
178     \@esphack
179   \endgroup}

```

4.7 Using acronyms

`\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>`.

```
180 \newcommand*\AC@get[3]{%
181   \ifx#1\relax
182     \PackageWarning{acronym}{Acronym ‘#3’ is not defined}%
183     \textbf{#3!}%
184   \else
185     \expandafter#2#1\null
186   \fi}
```

`\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>`.

```
187 \newcommand*\AC@acs[1]{%
188   \expandafter\AC@get\csname fn@#1\endcsname\@firstoftwo{#1}}
189 \newcommand*\AC@acl[1]{%
190   \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`.

```
191 \newcommand*\acs[1]{%
192   \texorpdfstring{\protect\@acs{#1}}{#1}}
193 \newcommand*\@acs[1]{%
194   \acsfont{\AC@acs{#1}}%
195 %% having a footnote on acs sort of defetes the purpose
196 %%   \ifAC@footnote
197 %%     \footnote{\AC@acl{#1}}}%
198 %%   \fi
199   \AC@logged{#1}}
```

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

```
\@acl 200 \newcommand*\acl{\protect\@acl}
201 \newcommand*\@acl[1]{%
202   \AC@acl{#1}%
203   \AC@logged{#1}}
```

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

```
204 \newcommand*\acf[1]{%
```

```

205 \texorpdfstring{\protect\@acf{#1}}{\AC@acl{#1} (#1)}%
206 }
207 \newcommand*{\@acf}[1]{%
208 \ifAC@footnote
209 \acsfont{\AC@acs{#1}}%
210 \footnote{\AC@acl{#1}{}}%
211 \else
212 \acffont{%
213 \AC@acl{#1}%
214 \nolinebreak[3] %
215 \acsfont{(\acsfont{\AC@acs{#1}})}%
216 }%
217 \fi
218 \expandafter\ifx\csname ac@#1\endcsname\AC@used
219 \relax%
220 \else
221 \global\expandafter\let\csname ac@#1\endcsname\AC@used
222 \AC@addtoclearlist{#1}%
223 \fi
224 \AC@logged{#1}}

```

`\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.

```

225 \newcommand{\ac}{\protect\@ac}%
226 \newcommand{\@ac}[1]{%
227 \ifAC@dua
228 \acl{#1}%
229 \else
230 \expandafter\ifx\csname ac@#1\endcsname\AC@used
231 \acs{#1}%
232 \else
233 \acf{#1}%
234 \fi
235 \fi
236 }

```

`\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'.

```

237 \newcommand*{\acsp}[1]{%
238 \texorpdfstring{\protect\@acsp{#1}}{#1s}}
239 \newcommand*{\@acsp}[1]{%
240 \acsfont{\AC@acs{#1}}s%
241 \AC@logged{#1}}

```

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

```

\@aclp 242 \newcommand*{\aclp}{\protect\@aclp}

```

```

243 \newcommand*{\@aclp}[1]{%
244   \AC@acl{#1}s%
245   \AC@logged{#1}}

```

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

```

246 \newcommand*{\acfp}[1]{%
247   \texorpdfstring{\protect\@acfp{#1}}{\AC@acl{#1}s (#1s)}%
248   }
249 \newcommand*{\@acfp}[1]{%
250   \ifAC@footnote
251     \acsfont{\AC@acs{#1}}s%
252     \footnote{\AC@acl{#1}s{}}%
253   \else
254     \acffont{%
255       \AC@acl{#1}s%
256       \nolinebreak[3] %
257       \acfsfont{(\acsfont{\AC@acs{#1}}s)}%
258     }%
259   \fi
260   \expandafter\ifx\csname ac@#1\endcsname\AC@used
261     \relax%
262   \else
263     \global\expandafter\let\csname ac@#1\endcsname\AC@used
264     \AC@addtoclearlist{#1}%
265   \fi
266   \AC@logged{#1}}

```

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

```

267 \newcommand{\acp}{\protect\@acp}
268 \newcommand{\@acp}[1]{%
269   \ifAC@dua
270     \aclp{#1}%
271   \else
272     \expandafter\ifx\csname ac@#1\endcsname\AC@used
273       \acsp{#1}%
274     \else
275       \acfp{#1}%
276     \fi
277   \fi
278 }
279 \endinput
280 </acronym>

```

That's it.