

An Acronym Environment for L^AT_EX 2_ε*

Tobias Oetiker

August 17, 2000

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.

`\acf` If later in the text again the Full Name of the acronym should be printed, use the command

```
\acf{\acronym}
```

to access the acronym. It stands for “acronym full” 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 used throughout your document. It is possible to add a longer description to each acronym definition.

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

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

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 `\acl{H2O}` prints “water”.

The `acronym` environment uses the `description` environment and the whole `\acro` definition acts like:

*This file has version v1.6 last revised 2000/05/21

`\item[acronym]full name explanation`

Or, if the optional argument *short name* is given, it looks like:

`\item[short name]full name explanation`

\acrodef If you want to define acronyms which do not appear in the **acronym** environment, you can use the command:

`\acrodef{acronym}[short name]{full name}`

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. And the acronym definitions can be anywhere in the text.

3 An example file

```
1 (*acrotest)
2 \documentclass{article}
3 \usepackage{acronym}
4 \begin{document}
5 \section{Intro}
6 In the early nineties, \acs{GSM} was deployed in many European
7 countries. \ac{GSM} offered for the first time international
8 roaming for mobile subscribers. The \acs{GSM}'s use of \ac{TDMA} as
9 its communication standard was debated at length. And every now
10 and then there are big discussion whether \ac{CDMA} should have
11 been chosen over \ac{TDMA}.
12
13 If you want to know more about \acf{GSM}, \acf{TDMA}, \acf{CDMA}
14 and \ac{oa}, just read a book about mobile communication.
15
16 \subsection{Some chemistry and physics}
17 \ac{NAD+} is a major electron acceptor in the oxidation of fuel
18 molecules. The reactive part of \ac{NAD+} is its nictinamide ring,
19 a pyridine derivate.
20
21 One mol consists of \acs{NA} atoms or molecules. There is a relation
22 between the constant of Boltzmann and the \acl{NA}:
23 \begin{equation}
24   k = R/\acs{NA}
25 \end{equation}
26
27 \section{Acronyms}
28 \begin{acronym}
29   \acro{GSM}{Global System for Mobile communication}.
30   \acs{GSM} is the new standard for digital cellular
31   communication in Europe.
32   \acro{TDMA}{Time Division Multiple Access}.
33   Some would say, that this is not as good as \ac{CDMA}.
34   \acro{CDMA}{Code Division Multiple Access}. The spread spectrum
35   modulation used in the Qualcomm system.
36   \acrodef{oa}{other acronyms}
37   \acro{NAD+}[NAD\textsuperscript{+}]{Nicotinamide Adenine Dinucleotide}.
38   \acro{NA}[\ensuremath{N_{\mathrm{A}}}]{Number of Avogadro}:
39   $\acs{NA} = 6.022045\cdot 10^{23}\,\mathrm{mol}^{-1}$
40 \end{acronym}
41 \end{document}
42 \end{acrotest}
```

4 The implementation

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

```
43 (*acronym)
44 \NeedsTeXFormat{LaTeX2e}[1994/06/01]
45 \ProvidesPackage{acronym}
46 [\filedate\space\fileversion\space Support for acronyms]
```

4.1 Defining acronyms

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

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

There are two possibilities to scan the optional argument: It can be done by each command, or it can be done by a help command, that uses the information of the first argument to provide the optional argument, if it is not given. So the next command always have an optional argument. Here the second method is used, because it keeps the definition of the acronym defining commands simpler.

`\AC@dblargafter` The internal help macro `\AC@dblargafter` implements the second method:

```
\AC@dblargafter{first}\cmd[opt]... calls \cmd{first}[opt]...
and
\AC@dblargafter{first}\cmd... gives \cmd{first}[first]...
47 \newcommand{\AC@dblargafter}[2]{%
48 \ifnextchar[{\#2{\#1}}{\AC@@dblargafter{\#1}{\#2}}}%
49 }
50 \newcommand\AC@@dblargafter{}
51 \def\AC@@dblargafter#1#2{\#2{\#1}{\#1}}
```

`\newacro` The internal macro `\newacro` stores the *full name* and eventually the *short name* of the acronym in the command `\fn@<acronym>`. The *short name* is only saved, if it differs from the *acronym*.

```
52 \newcommand*{\newacro}[1]{\AC@dblargafter{\#1}\AC@newacro}
53 \newcommand\AC@newacro{}
54 \def\AC@newacro#1[#2]#3{%
55 \begingroup
56 \def\reserved@a{\#1}%
57 \def\reserved@b{\#2}%
58 \ifx\reserved@a\reserved@b
59 \expandafter\gdef\csname fn@#1\endcsname{\#3}%
60 \else
61 \expandafter\gdef\csname fn@#1\endcsname{\#3}{\#2}%
62 \fi
63 \endgroup
64 }
```

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

```
65 \newcommand*{\acrodef}[1]{\AC@dblargafter{\#1}\AC@acrodef}
66 \newcommand\AC@acrodef{}
67 \def\AC@acrodef#1[#2]#3{%
```

```

68 \newacro{#1}[{#2}]{#3}%
69 \if@filesw
70   \begingroup
71     \def\reserved@a{#1}%
72     \def\reserved@b{#2}%
73     \ifx\reserved@a\reserved@b
74       \protected@write\@auxout{}{\protect\newacro{#1}{#3}}%
75     \else
76       \protected@write\@auxout{}{\protect\newacro{#1}[{#2}]{#3}}%
77     \fi
78   \endgroup
79 \fi
80 }

```

acronym I like to have a list of all acronyms I used in my document. Therefore you can define your acronyms inside the `acronym` environment. Not only stating the name of the acronym, but optionally also giving an explanation on it.

```

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

```

\acro Acronyms can be defined with the user command `\acro` in this `acronym` environment:

```

81 \newenvironment{acronym}{%
82   \begin{description}
83     \providecommand*\acro}[1]{\AC@dblargafter{##1}\AC@acro}%
84 }{%
85   \end{description}}%
86 }
87 \newcommand\AC@acro{}
88 \def\AC@acro#1[#2]#3{%
89   \acrodef{#1}[{#2}]{#3}%
90   \item[{#2}] #3%
91 }

```

4.2 Using acronyms

\AC@check If the acronym is undefined, the internal macro `\AC@check` warns the user and provides a default for the full name. The acronym is printed fat with an exclamation mark at the end.

```

92 \newcommand*\AC@check}[1]{%
93   \expandafter\ifx\csname fn@#1\endcsname\relax
94     \PackageWarning{acronym}{Acronym ‘#1’ is not defined}%
95     \expandafter\gdef\csname fn@#1\endcsname
96       {\textbf{#1!}}}%
97   \fi
98 }

```

\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
\acfsfont
99 \newcommand*\acsfont}[1]{#1}
100 \newcommand*\acffont}[1]{#1}
101 \newcommand*\acfsfont}[1]{#1}

```

`\acs` The user macro `\acs` prints the short form of the acronym using the font specified by `\acsfont`. This is the acronym itself or the *short name*, if the optional argument is given in the definition of the acronym. This *short name* is available as second argument of the data in `\fn@<acronym>`. The help macro `\AC@getsecond` extracts this second argument out of a list of two or three arguments. The output goes through `\acsfont`.

```
102 \newcommand*{\acs}[1]{%
103   \AC@check{#1}%
104   \acsfont{%
105     \expandafter\expandafter\expandafter\AC@getsecond
106     \csname fn@#1\endcsname{#1}\@nil
107   }%
108 }
109 \newcommand\AC@getsecond{}
110 \long\def\AC@getsecond#1#2#3\@nil{#2}
```

`\acl` The user macro `\acl` prints the full name of the acronym. It uses the L^AT_EX macro `\@car` to extract the first argument of the data, that are stored in the command `\fn@<acronym>`.

```
111 \newcommand*{\acl}[1]{%
112   \AC@check{#1}%
113   \expandafter\expandafter\expandafter\@car
114   \csname fn@#1\endcsname\@nil
115 }
```

`\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.

```
116 \newcommand*{\acf}[1]{%
117   \acffont{%
118     \acl{#1}%
119     \nolinebreak[3] %
120     \acsfont{(\acs{#1})}%
121   }%
122 }
```

`\ac` The first time an acronym is accessed its Full Name (FN) is printed. The next time just (FN). This is done by `\gdefining` the `\ac@FN` to be `\@empty` after its first use.

```
123 \newcommand{\ac}[1]{%
124   \expandafter\ifx\csname ac@#1\endcsname\relax
125     \acf{#1}%
126     \global\expandafter\let\csname ac@#1\endcsname\@empty
127   \else
128     \acs{#1}%
129   \fi
130 }
```

`\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’. This macro is a wrapper around `\acs`.

```
131 \newcommand*{\acsp}[1]{%
132   \acs{#1}s%
133 }
```

`\ac1p` The user macro `\ac1p` prints the plural full name of the acronym. It is a wrapper around `\ac1`.

```
134 \newcommand*{\ac1p}[1]{%
135   \ac1{#1}s%
136 }%
```

`\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.

```
137 \newcommand*{\acfp}[1]{%
138   \acffont{%
139     \ac1p{#1}%
140     \nolinebreak[3] %
141     \acfsfont{(\acsp{#1})}%
142   }%
143 }
```

`\acp` The first time an acronym is accessed Full Names (FNs) is printed. The next time just (FNs). This is done by `\gdefining` the `\ac@FN` to be `\@empty` after its first use.

```
144 \newcommand{\acp}[1]{%
145   \expandafter\ifx\csname ac@#1\endcsname\relax
146     \acfp{#1}%
147     \global\expandafter\let\csname ac@#1\endcsname\@empty
148   \else
149     \acsp{#1}%
150   \fi
151 }
```

4.3 Options

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

```
152 \DeclareOption{footnote}{%
153   \renewcommand*{\acf}[1]{%
154     \acs{#1}%
155     \footnote{\ac1{#1}{}}%
156   }%
157   \renewcommand*{\acfp}[1]{%
158     \acsp{#1}%
159     \footnote{\ac1p{#1}{}}%
160   }%
161 }
```

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 `resize` package. As `\RequirePackage` cannot be used inside `\DeclareOption`, we need a boolean variable.

```
162 \newif\if@ac@smaller
163 \@ac@smallerfalse
164 \DeclareOption{smaller}{\@ac@smallertrue}
```

Now we process the options and care for the `smaller` option.

```
165 \ProcessOptions\relax
166 \if@ac@smaller
167   \RequirePackage{reysize}
168   \renewcommand*{\acsfont}[1]{\textsmaller{#1}}
169 \fi
```

That's it.

```
170 \endinput
171 \</acronym>
```