An Acronym Environment for LaTeX 2ε*

Tobias Oetiker

2004/06/08

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 v1.12, last revised 2004/06/08.

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.

\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}
```

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{$\mathrm{H_2O}$}{H2O}) ...]
        {The \acf{H2O} ...}
```

3 An example file

```
⟨*acrotest⟩
\documentclass{article}
\usepackage[printonlyused]{acronym}
\begin{document}

\section{Intro}
In the early nineties, \acs{GSM} was deployed in many European
countries. \ac{GSM} offered for the first time international
roaming for mobile subscribers. The \acs{GSM}'s use of \ac{TDMA} as
its communication standard was debated at length. And every now
and then there are big discussion whether \ac{CDMA} should have
been chosen over \ac{TDMA}.

\section{Furthermore}
\acresetall
The reader could have forgotten all the nice acronyms, so we repeat the
meaning again.

If you want to know more about \acf{GSM}, \acf{TDMA}, \acf{CDMA}
and other acronyms, just read a book about mobile communication. Just
to mention it: There is another \ac{UA}, just for testing purposes!

\subsection{Some chemistry and physics}
\label{Chem}
\ac{NAD+} is a major electron acceptor in the oxidation
of fuel molecules. The reactive part of \ac{NAD+} is its nictinamide
ring, a pyridine derivate.

One mol consists of \acs{NA} atoms or molecules. There is a relation
between the constant of Boltzmann and the \acl{NA}:
\begin{equation}
  k = R/\acs{NA}
\end{equation}

\section{Acronyms}
\begin{acronym}[TDMA]
 \acro{CDMA}{Code Division Multiple Access}
 \acro{GSM}{Global System for Mobile communication}
 \acro{NA}[\ensuremath{N_{\mathrm A}}]
      {Number of Avogadro\acroextra{ (see \S\ref{Chem})}}
 \acro{NAD+}[NAD\textsuperscript{+}]{Nicotinamide Adenine Dinucleotide}
 \acro{NUA}{Not Used Acronym}
 \acro{TDMA}{Time Division Multiple Access}
 \acro{UA}{Used Acronym}
\end{acronym}

\end{document}
⟨/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}[2004/11/03
52                           v1.16
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}
```

Now we process the options.

```
66 \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.

```
\ifAC@smaller
  \RequirePackage{relsize}
  \newcommand*{\acsfont}[1]{\textsmaller{#1}}
\else
  \newcommand*{\acsfont}[1]{#1}
\fi
\newcommand*{\acffont}[1]{#1}
\newcommand*{\acfsfont}[1]{#1}
```

4.4 Hyperlinks and PDF support

\AC@hyperlink
\AC@hypertarget

Define dummy hyperlink commands

```
\def\AC@hyperlink#1#2{#2}
\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.

```
\ifAC@nohyperlinks
\else
   \AtBeginDocument{%
      \@ifpackageloaded{hyperref}
         {\let\AC@hyperlink=\hyperlink
          \newcommand*\AC@raisedhypertarget[2]{%
             \Hy@raisedlink{\hypertarget{#1}{}}#2}%
          \let\AC@hypertarget=\AC@raisedhypertarget}{}}
\fi
```

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

```
\AtBeginDocument{%
   \providecommand\texorpdfstring[2]{#1}%
   \providecommand\pdfstringdefDisableCommands[1]{}%
}
```

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

```
\newtoks\clearlist
```

\AC@addtoclearlist Adds acronyms to the clear list

```
\newcommand*\AC@addtoclearlist[1]{%
  \global\clearlist\expandafter{\the\clearlist\AC@reset{#1}}%
}
```

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

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

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

```
\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.

```
\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.

```
\newcommand*{\AC@logged}[1]{%
   \@bsphack
   \protected@write\@auxout{}{\string\acronymused{#1}}%
   \@esphack}
```

Keep it out of bookmarks.

```
\AtBeginDocument{%
   \pdfstringdefDisableCommands{%
      \let\AC@logged\@gobble
   }%
}
```

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

```
\newcommand*{\acronymused}[1]{%
   \expandafter\ifx\csname acused@#1\endcsname\AC@used
      \relax
   \else
       \global\expandafter\let\csname acused@#1\endcsname\AC@used
       \global\let\AC@populated\AC@used
   \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*`>`.

```
\newcommand*\newacro[1]{%
  \@ifnextchar[{\AC@newacro{#1}}{\AC@newacro{#1}[\AC@temp]}}
\newcommand\AC@newacro{}
\def\AC@newacro#1[#2]#3{%
   \def\AC@temp{#1}%
   \expandafter\gdef\csname fn@#1\endcsname{{#2}{#3}}%
   }
```

`\acrodef` `\AC@acrodef`

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

```
\newcommand*\acrodef[1]{%
  \@ifnextchar[{\AC@acrodef{#1}}{\AC@acrodef{#1}[\AC@temp]}}
\newcommand\AC@acrodef{}
\def\AC@acrodef#1[#2]#3{%
   \def\AC@temp{#1}%
   \@bsphack
   \protected@write\@auxout{}{\string\newacro{#1}[#2]{#3}}%
   \@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.

```
\def\bflabel#1{{\textbf{\textsf{#1}}\hfill}}
\newenvironment{AC@deflist}[1]%
        {\raggedright\begin{list}{}%
                {\settowidth{\labelwidth}{\textbf{\textsf{#1}}}%
                \setlength{\leftmargin}{\labelwidth}%
                \addtolength{\leftmargin}{\labelsep}%
                \renewcommand{\makelabel}{\bflabel}}}%
        {\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.

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

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

```
\newenvironment{acronym}[1][1]{%
   \providecommand*{\acro}{\AC@acro}%
   \long\def\acroextra##1{##1}%
   \ifx1#1
      \global\expandafter\let\csname ac@des@mark\endcsname\AC@used
      \begin{description}%
   \else
      \begin{AC@deflist}{#1}%
   \fi%
  }%
  {%
   \ifx\AC@populated\AC@used\else
      \item[]\relax
   \fi
   \expandafter\ifx\csname ac@des@mark\endcsname\AC@used
      \end{description}%
   \else
      \end{AC@deflist}%
   \fi}
```

\AC@acro
\AC@@acro

```
\newcommand*\AC@acro[1]{%
  \@ifnextchar[{\AC@@acro{#1}}{\AC@@acro{#1}[\AC@temp]}}
\newcommand\AC@@acro{}
\def\AC@@acro#1[#2]#3{%
 \def\AC@temp{#1}%
 \ifAC@printonlyused
    \expandafter\ifx\csname acused@#1\endcsname\AC@used%
       \item[\protect\AC@hypertarget{#1}{\acsfont{#2}}] #3%
    \fi%
 \else%
    \item[\protect\AC@hypertarget{#1}{\acsfont{#2}}] #3%
 \fi
 \begingroup
    \def\acroextra##1{}%
    \@bsphack
    \protected@write\@auxout{}%
       {\string\newacro{#1}[\string\AC@hyperlink{#1}{#2}]{#3}}%
    \@esphack
  \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*>.

```
\newcommand*\AC@get[3]{%
   \ifx#1\relax
      \PackageWarning{acronym}{Acronym `#3' is not defined}%
      \textbf{#3!}%
   \else
      \expandafter#2#1\null
   \fi}
```

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

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

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

```
\newcommand*{\acs}[1]{%
   \texorpdfstring{\protect\@acs{#1}}{#1}}
\newcommand*{\@acs}[1]{%
   \acsfont{\AC@acs{#1}}%
%% having a footnote on acs sort of defetes the purpose
%%   \ifAC@footnote
%%      \footnote{\AC@acl{#1}{}}%
%%   \fi
   \AC@logged{#1}}
```

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

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

\acf \@acf The user macro \acf always prints the full name with the acronym. The format depends on \acffont and \acfsfont, 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 \acfsfont.

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

```
   \texorpdfstring{\protect\@acf{#1}}{\AC@acl{#1} (#1)}%
   }
\newcommand*{\@acf}[1]{%
    \ifAC@footnote
       \acsfont{\AC@acs{#1}}%
       \footnote{\AC@acl{#1}{}}%
    \else
       \acffont{%
          \AC@acl{#1}%
          \nolinebreak[3] %
          \acfsfont{(\acsfont{\AC@acs{#1}})}%
        }%
    \fi
    \expandafter\ifx\csname ac@#1\endcsname\AC@used
        \relax%
    \else
        \global\expandafter\let\csname ac@#1\endcsname\AC@used
        \AC@addtoclearlist{#1}%
    \fi
    \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.

```
\newcommand{\ac}{\protect\@ac}%
\newcommand{\@ac}[1]{%
   \expandafter\ifx\csname ac@#1\endcsname\AC@used
      \acs{#1}%
   \else
      \acf{#1}%
   \fi}
```

`\acsp` `\@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'.

```
\newcommand*{\acsp}[1]{%
   \texorpdfstring{\protect\@acsp{#1}}{#1s}}
\newcommand*{\@acsp}[1]{%
   \acsfont{\AC@acs{#1}}s%
   \AC@logged{#1}}
```

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

```
\newcommand*{\aclp}{\protect\@aclp}
\newcommand*{\@aclp}[1]{%
   \AC@acl{#1}s%
   \AC@logged{#1}}
```

\acfp
\@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.

```
\newcommand*{\acfp}[1]{%
   \texorpdfstring{\protect\@acfp{#1}}{\AC@acl{#1}s (#1s)}%
   }
\newcommand*{\@acfp}[1]{%
   \ifAC@footnote
      \acsfont{\AC@acs{#1}}s%
      \footnote{\AC@acl{#1}s{}}%
   \else
      \acffont{%
         \AC@acl{#1}s%
         \nolinebreak[3] %
         \acfsfont{(\acsfont{\AC@acs{#1}}s)}%
         }%
   \fi
   \expandafter\ifx\csname ac@#1\endcsname\AC@used
      \relax%
   \else
       \global\expandafter\let\csname ac@#1\endcsname\AC@used
          \AC@addtoclearlist{#1}%
   \fi
   \AC@logged{#1}}
```

\acp
\@acp

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

```
\newcommand{\acp}{\protect\@acp}
\newcommand{\@acp}[1]{%
   \expandafter\ifx\csname ac@#1\endcsname\AC@used
      \acsp{#1}%
   \else
      \acfp{#1}%
   \fi}

\endinput
```

⟨/acronym⟩

That's it.