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\[⟨linebreak penalty⟩\]ˈ⟨acronym⟩

To enter an acronym inside the text, use the \ac 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.

When an acronym is being used, for the first time (with the footnote option not specified), next to the end of the line, a line break between the full name of the acronym and the acronym in brackets can be encountered. The optional variable represents the penalty level of breaking the line at that place, taking integer values between 0 and 4. A higher number corresponds to a higher penalty.

\Ac

Works in the same way as \ac, but starts the long form with an upper case
letter. Use case: when the acronym is used for the first time, at the beginning of a sentence.

\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[⟨linebreak penalty⟩]{⟨acronym⟩}

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

When an full acronym is being used next to the end of the line, a line break between the full name of the acronym and the acronym in brackets can be encountered. The optional variable represents the penalty level of breaking the line at that place, taking integer values between 0 and 4. A higher number corresponds to a higher penalty.

\Acf Works in the same way as \acf, but starts the long form with an upper case letter.

\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⟩}

\Acl Works in the same way as \ac, but makes the short and/or long forms into plurals.

\Acp Works in the same way as \acp, but starts the long form with an upper case letter.

\acfp Works in the same way as \acf, but makes the short and long forms into plurals.

\Acfp Works in the same way as \acfp, but starts the long form with an upper case letter.

\acsp Works in the same way as \acs, but makes the short form into a plural.

\aclp Works in the same way as \aclp, but starts with an upper case letter.

\acfi Works in the same way as \acf, but prints the Full Name acronym (\acl) in italics and the abbreviated form (\acs) in upshaped form.

\Acfi Works in the same way as \acfi, but starts the long form with an upper case letter.

\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.
\aclu Works in the same way as `\aclu`, but starts with an upper case letter.

   Example: `\acl{lox}/\acl{lh2} \acsu{lox}/\acsu{lh2}`

\iac Works in the same way as the `\ac` command but prefixes it with an appropriate

   indefinite article.

\iacu Works in the same way as the `\ac` command but prefixes it with an appropriate

   upper case indefinite article.

\...* 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{\...}, \Ac{\...}, \acs{\...}, \acl{\...}, \acf{\...}, \Acf{\...}, \acp{\...}, \Acp{\...}, \acsu{\...}, \acsu{\...}, \iac{\...} \Iac{\...} \acfs{\...} \Acf{\...}`

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 and upper case 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:

   \begin{verbatim}
   \renewcommand*{\acsfont}[1]{\textsmaller{#1}}
   \end{verbatim}

2.3 Defining Acronyms

Acronyms can either defined from an environment specifically introduced for that

   purpose or by direct definitions.

   The `acro` 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.

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

   \begin{verbatim}
   \acro{(acronym)}{(short name)}{(full name)}
   \end{verbatim}

   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, \ldots), then this code can be given in the optional argument \textit{(short name)}. The first argument \textit{(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\textsubscript{2}O” and \acl{H2O} prints “water”.

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_{\text{\textit{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 \texttt{description} environment. If you pass an optional parameter to the \acro 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 \textit{HBCI} is the longest acronym used, the list should start with

\begin{acronym}[HBCI]

The short form of each acronym in the list is formatted using \texttt{\aclabelfont}, which typesets its arguments in bold font by default. It can be redefined to produce bold sans-serif labels, for example, with

\renewcommand*{\aclabelfont}[1]{\textbf{\textsf{\acsfont{#1}}}}

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 \texttt{printonlyused} (used at least once) or \texttt{printonlyreused} (use more than once):

\usepackage[printonlyused]{acronym}

In \texttt{printonly(re)used}-mode you can add to each acronym the the page number where it was first used by additionally specifying the option \texttt{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

\acrodef
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 indefinite articles

Sometimes the indefinite article of an acronym differs between its short form and its long form, for example “a Federal Bureau of Investigation (FBI) agent” and “an FBI agent”. To deal with this, the package provides the following three commands

\acroindefinite\acrodefindefinite\acroindefinite

that allow one to define indefinite articles. The \acroindefinite command is meant to be used in the acronym environment. The difference among the latter two is that \acrodefindefinite puts the acronym definition in the .aux file, so that the acronym exception is available at the next run from start-up.

When using \iac and \Iac without first defining an article, the default article is “a”.

### 2.3.2 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 Parlaments”, 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)”.

\newacro{IC}{Integrated Circuit}
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

\acroplural{}, \newacroplural{}, \acrodefplural{}

that allow one to define plural exceptions. The \acroplural{} command is meant to be used in the \acro{} 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 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 \shortname{} for pdf\LaTeX\ with \hyperref{}.

• When the long form of the acronym is used in sectional headers (for pdf\LaTeX\ 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 \textorpdfstring{} defined by \hyperref{}, for example:

\acro{Nx}{\ensuremath{N_{\chi}}}{\textorpdfstring{$\chi$}{X}-factor}

which will then give
• For acronyms in sectional headers, the file must be PDF\LaTeX ed 3 times before the bookmarks are correct.

• Acronyms in sectional headers together with the \texttt{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:

\begin{verbatim}
\chapter[The water \texttt{\textorpdfstring{$\mathrm{H}_2\mathrm{O}$}{H2O}) ...]{The \texttt{\textorpdfstring{$\mathrm{H}_2\mathrm{O}$}{H2O}) ...}]
\end{verbatim}
\section{An example file}

\documentclass{article}
\usepackage{colorlinks}
\usepackage{hyperref}
\usepackage{printonlyused,withpage}{acronym}
\begin{document}
\section{Intro}
In the early nineties, \acs{GSM} was deployed in many European countries. \acs{GSM} offered for the first time international roaming for mobile subscribers. The \acs{GSM}'s use of \acs{TDMA} as its communication standard was debated at length. And every now and then there are big discussion whether \acs{CDMA} should have been chosen over \acs{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 \acs{GSM}, \acs{TDMA}, \acs{CDMA} and other acronyms, just read a book about mobile communication. Just to mention it: There is another \ac{UA}, just for testing purposes!

\begin{figure}[h]
\caption{A float also admits references like \acs{GSM} or \acs{CDMA}.}
\end{figure}

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

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

\Acp{LFVP} are processes in which the lepton number of the initial and final states are different. An example for \iac{LFVP} is neutrinoless double beta decay.

\subsection{Some testing fundamentals}
When testing \acs{IC}, one typically wants to identify functional
blocks to be tested separately. The latter are commonly indicated as \ac{BUT}. To test a \ac{BUT} requires defining a testing strategy.

\section{Acronyms}
\begin{acronym}
\acro{CDMA}{Code Division Multiple Access}
\acro{GSM}{Global System for Mobile communication}
\acro{NA}{Number of Avogadro (see \S\ref{Chem})}
\acro{NAD+}{Nicotinamide Adenine Dinucleotide}
\acro{LFVP}{lepton flavor violating process}
\acro{NUA}{Not Used Acronym}
\acro{TDMA}{Time Division Multiple Access}
\acro{UA}{Used Acronym}
\acro{LOX}{Liquid Oxygen}
\acro{LH_2}{Liquid Hydrogen}
\acro{IC}{Integrated Circuit}
\acro{BUT}{Block Under Test}
\end{acronym}
4 The implementation

4.1 Identification

First we test that we got the right format and name the package.
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\ProvidesPackage{acronym}[2020/03/13 v1.46]
 Support for acronyms (Tobias Oetiker)
\RequirePackage{suffix,xstring}

4.2 Options

\IfAC@footnote \newif\ifAC@footnote \AC@footnotefalse
\IfAC@footnote\DeclareOption{footnote}{\AC@footnotetrue}
\IfAC@nohyperlinks \newif\ifAC@nohyperlinks \AC@nohyperlinksfalse
\IfAC@nohyperlinks\DeclareOption{nohyperlinks}{\AC@nohyperlinkstrue}
\IfAC@noacroprefix \newif\ifAC@noacroprefix \AC@noacroprefixfalse
\IfAC@noacroprefix\DeclareOption{noacroprefix}{\AC@noacroprefixtrue}
\IfAC@printonlyused \newif\ifAC@printonlyused \AC@printonlyusedfalse
\IfAC@printonlyused\DeclareOption{printonlyused}{\AC@printonlyusedtrue}
\IfAC@printonlyreused \newif\ifAC@printonlyreused \AC@printonlyreusedfalse
\IfAC@printonlyreused\DeclareOption{printonlyreused}{\AC@printonlyreusedtrue}

\ifAC@footnote The option footnote leads to a redefinition of \acf, \Acf, \acfp, and \Acfp, making the full name appear as a footnote.
\ifAC@footnote\newif\ifAC@footnote \AC@footnotefalse
\ifAC@footnote\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.
\ifAC@nohyperlinks\newif\ifAC@nohyperlinks \AC@nohyperlinksfalse
\ifAC@nohyperlinks\DeclareOption{nohyperlinks}{\AC@nohyperlinkstrue}
\ifAC@noacroprefix With the noacroprefix option the acronym commands are not prefixed. This reproduces the old behavior of version 1.43, but can cause collisions between user-defined acronyms and commands of this package.
\ifAC@noacroprefix\newif\ifAC@noacroprefix \AC@noacroprefixfalse
\ifAC@noacroprefix\DeclareOption{noacroprefix}{\AC@noacroprefixtrue}
\ifAC@printonlyused We need a marker which is set if the option printonlyused was used.
\ifAC@printonlyused\newif\ifAC@printonlyused \AC@printonlyusedfalse
\ifAC@printonlyused\DeclareOption{printonlyused}{\AC@printonlyusedtrue}
\ifAC@printonlyreused With the printonlyreused option, only those acronyms are included in the list of acronyms that have been used more than once, i.e. at least twice.
\ifAC@printonlyreused\newif\ifAC@printonlyreused \AC@printonlyreusedfalse
\ifAC@printonlyreused\DeclareOption{printonlyreused}{\AC@printonlyreusedtrue}
\ifAC@withpage  A marker which tells us to print page numbers.
\newif\ifAC@withpage
\AC@withpagefalse
\DeclareOption{withpage}{\AC@withpagetrue}
\ifAC@smaller
The option \texttt{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 \texttt{textsmaller} provided by the \texttt{relsize} package. As
\texttt{\RequirePackage} cannot be used inside \texttt{\DeclareOption}, we need a boolean vari-
able.
\newif\ifAC@smaller
\AC@smallerfalse
\DeclareOption{smaller}{\AC@smallertrue}
\ifAC@dua
The option \texttt{dua} stands for “don’t use acronyms”. It leads to a redefinition of \ac,
\Ac, \acp, and \Acp, making the full name appear all the time and suppressing
all acronyms but the explicitity requested by \acf, \Acf, \acfp or \Acfp.
\newif\ifAC@dua
\AC@duafalse
\DeclareOption{dua}{\AC@duatrue}
\ifAC@nolist
The option \texttt{nolist} stands for “don’t write the list of acronyms”.
\newif\ifAC@nolist
\AC@nolistfalse
\DeclareOption{nolist}{\AC@nolisttrue\AC@nohyperlinkstrue}
\ifAC@nolinebreak
The option \texttt{nolinebreak} dictates whether to forbid, by defalt, a line break between
the full name and the short name, when they are presented together.
\newif\ifAC@nolinebreak
\AC@nolinebreakfalse
\DeclareOption{nolinebreak}{\AC@nolinebreaktrue}

Now we process the options.
\ProcessOptions\relax

4.3  Setup macros
\acsfont
\acffont
\acfsfont
The appearance of the output of the commands \texttt{\acs} and \texttt{\acf} is partially con-
trolled by \acsfont, \acffont, and \acfsfont. By default, they do nothing
except when the \texttt{smaller} option is loaded.

The option \texttt{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 \texttt{textsmaller} provided by the \texttt{relsize} package.
\ifAC@smaller
\\RequirePackage{relsize}
\newcommand*{\acsfont}{\textsmaller{#1}}
\else

When the option \texttt{nolinebreak} is specified, the default penalty for a line break is being set to the maximum. Otherwise, the default penalty is one level below the maximum, meaning that most of the times, by default, the line will not get broken.

\begin{verbatim}
\AC@linebreakpenalty
\newcommand*{\acsfont}[1]{#1}
\newcommand*{\acffont}[1]{#1}
\newcommand*{\acsfsfont}[1]{#1}
\end{verbatim}

\subsection{Hyperlinks and PDF support}

Define dummy hyperlink commands

\begin{verbatim}
\AC@hyperlink
\AC@hyperref
\AC@hypertarget
\AC@phantomsection
\end{verbatim}

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

\begin{verbatim}
\AC@raisedhypertarget
\AtBeginDocument{\if\AC@nohyperlinks}
\AtBeginDocument{\@ifpackageloaded{hyperref}{
\let\AC@hyperlink=\hyperlink
\let\AC@hyperref=\hyperref
\newcommand*{\AC@raisedhypertarget}[2]{\Hy@raisedlink{\hypertarget{#1}{}}#2}
\let\AC@hypertarget=\AC@raisedhypertarget
\def\AC@phantomsection{\Hy@GlobalStepCount\Hy@linkcounter\edef\@currentHref{section*\the\Hy@linkcounter}\hyper@anchorstart{\@currentHref}\hyper@anchorend}}}
\fi}
\end{verbatim}

\begin{verbatim}
\AC@pageref
\usepackage{hyperref}
\AtBeginDocument{%}
\if\packageloaded{hyperref}{
\let\AC@pageref=\@pagerefstar%
\end{verbatim}

Use \texttt{\pageref*} instead of \texttt{\pageref} when the hyperref package is used.

\begin{verbatim}
\AC@pageref
\AtBeginDocument{%}
\if\packageloaded{hyperref}{%
\let\AC@pageref=\@pagerefstar%
\end{verbatim}

\end{verbatim}
The `hyperref` package defines \texttt{\texttt{\texttt{\texttt{\texttt{pdfstringdefDisableCommands}}} and \texttt{\texttt{\texttt{\texttt{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]{}%
  \pdfstringdefDisableCommands{%
    \csname AC@starredfalse\endcsname
    \csname AC@footnotefalse\endcsname
    \let\AC@hyperlink\@secondoftwo
    \let\acsfont\relax
    \let\acffont\relax
    \let\acfsfont\relax
    \let\acused\relax
    \let\null\relax
    \def\AChy@call#1#2{%
      \ifx*#1\@empty
        \expandafter#2%
      \else
        #2{#1}%
      \fi
    }%
    \def\acs#1{\AChy@call{#1}\AC@acs}%
    \def\acl#1{\AChy@call{#1}\@acl}%
    \def\Acl#1{\AChy@call{#1}\@Acl}%
    \def\acf#1{\AChy@call{#1}\AC@acf}%
    \def\Acf#1{\AChy@call{#1}\AC@Acf}%
    \def\ac#1{\AChy@call{#1}\@ac}%
    \def\Ac#1{\AChy@call{#1}\@Ac}%
    \def\acsp#1{\AChy@call{#1}\@acsp}%
    \def\aclp#1{\AChy@call{#1}\@aclp}%
    \def\Aclp#1{\AChy@call{#1}\@Aclp}%
    \def\acfp#1{\AChy@call{#1}\AC@acfp}%
    \def\Acfp#1{\AChy@call{#1}\AC@Acf}%
    \def\acp#1{\AChy@call{#1}\@acp}%
    \def\Acp#1{\AChy@call{#1}\@Acp}%
    \def\acfi#1{\AChy@call{#1}\AC@acf}%
    \def\Acfi#1{\AChy@call{#1}\AC@Acf}%
  }%

\let\acsu\acs
\let\aclu\acl
\let\Aclu\Acl
\def\AChy@acf#1{\AC@acl{#1} (\AC@acs{#1})}%
\def\AChy@Acf#1{\AC@Acl{#1} (\AC@acs{#1})}%
\def\AChy@acfp#1{\AC@aclp{#1} (\AC@acsp{#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.

\Ac@clearlist
\newtoks\Ac@clearlist
\Ac@addto\Ac@clearlist Adds acronyms to the clear list
\newcommand*{\Ac@addto\Ac@clearlist}[1]{% \global\Ac@clearlist\expandafter{\the\Ac@clearlist\Ac@reset{#1}}%}
\acresetall \Ac@reset This macro resets the \Ac@FN - tag of each acronym, therefore \ac will use Full Name (FN) next time it is called
\newcommand*{\acresetall}{\the\Ac@clearlist\Ac@clearlist={}}\Ac@reset
\Ac@reset
\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 printonly(re)used option, because the acronym list are then empty, resulting in a "missing item" error.
\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.
This methodology is needed when the list of acronyms is in the front matter of the document.
\newcommand*{\Ac@logged}[1]{% \Acronymused{#1}% mark it as used in the current run too \@bsphack \protected@write\@auxout{}{\string\Acronymused{#1}}% \@esphack}
\Ac@logged
\Ac@logged
\Acronymused 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@once\endcsname\AC@used% 
  \expandafter\ifx\csname acused@#1@twice\endcsname\AC@used% 
  \relax% 
  \else% 
  \global\expandafter\let\csname acused@#1@twice\endcsname\AC@used% 
  \global\let\AC@populated\AC@used% 
  \fi% 
  \else% 
  \global\expandafter\let\csname acused@#1@once\endcsname\AC@used% 
  \ifAC@printonlyreused% 
  \relax% 
  \else% 
  \global\let\AC@populated\AC@used% 
  \fi% 
  \fi% 
} 

\@firstupper Internal commands for making a first letter upper case.
\newcommand{\@firstupper}[1]{% 
  \StrLeft{#1}{1}[\firstletter]% 
  \StrGobbleLeft{#1}{1}[\remainder]% 
  \MakeUppercase\firstletter\remainder% 
} 

\AC@prefix Returns the prefix used to build the defined acronym commands as long as the noacroprefix option is disabled. Otherwise the output is empty, so the old behaviour from version \texttt{1.43} is reproduced.
\ifAC@noacroprefix 
  \newcommand*{\AC@prefix{}} 
  \else 
  \newcommand*{\AC@prefix{acronyms@}} 
  \fi

4.6 Defining acronyms

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

\acro{\texttt{(acronym)}}{\texttt{(short name)}}{\texttt{(full name)}}

The mechanism used in this package is to make the optional \texttt{(short name)} identical to the \texttt{(acronym)} when it is empty (no optional argument), thereby only the second (optional) argument is stored together with the \texttt{(full name)}.
\acrodef The user command \acrodef calls \newacro and writes it into the .aux file.

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

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.

\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.

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

\newenvironment{acronym}[1][1]{% 
  \providecommand*{\acro}{\AC@acro}% 
  \providecommand*{\acroplural}{\AC@acroplural}% 
  \providecommand*{\acroindefinite}{\AC@acroindefinite}% 
  \long\def\acroextra##1{##1}% 
  \@tempa{1}\def\@tempb{#1}% 
  \ifx\@tempa\@tempb% 
    \global\expandafter\let\csname AC@des@mark\endcsname\AC@used% 
    \ifAC@nolist% 
      \else% 
      \begin{description}% 
    \else% 
      \begin{AC@deflist}{#1}% 
    \fi% 
  \else% 
    \begin{description}% 
  \fi% 
  \fi% 
  \expandafter\ifx\csname AC@des@mark\endcsname\AC@used% 
  \ifAC@nolist% 
    \else% 
    \end{description}% 
  \fi% 
  \else% 
  \end{AC@deflist}% 
  \fi% 
}\AC@acro

\AC@acro

\acro

\AC@acro
4.6.1 Nonstandard indefinite articles

\newacroindefinite Sets up a non standard indefinite article for a given acronym.
\begin{verbatim}
\newcommand*{\newacroindefinite}[3]{\expandafter\gdef\csname fn@#1@IS\endcsname{#2}\
\expandafter\gdef\csname fn@#1@IL\endcsname{#3} }
\end{verbatim}
\acrodefindefinite Same as above, storing content in aux file.
\begin{verbatim}
\newcommand*{\acrodefindefinite}[3]{\@bsphack\protected@write\@auxout{}{\string\newacroindefinite{#1}{#2}{#3}}\@esphack }
\end{verbatim}
\AC@acroindefinite Internal command to set up an indefinite article in the acronym environment.
\begin{verbatim}
\newcommand{\AC@acroindefinite}[3]{\@bsphack\protected@write\@auxout{}{\string\newacroindefinite{#1}{\string\AC@hyperlink{#1}{#2}}{#3}}\@esphack }
\end{verbatim}

4.6.2 Non standard or foreign plural forms

\newacroplural Sets up a non standard plural form for a given acronym.
\begin{verbatim}
\newcommand*{\newacroplural}[1]{\@ifnextchar[\AC@newacroplurali{#1}]{\AC@newacropluralii{#1}}
\end{verbatim}
\begin{verbatim}
\newcommand{\AC@newacroplurali}{
\def\AC@newacroplurali#1[#2]#3{\expandafter\gdef\csname fn@#1@PS\endcsname{#2}\
\expandafter\gdef\csname fn@#1@PL\endcsname{#3} }
\end{verbatim}
\begin{verbatim}
\newcommand{\AC@newacropluralii}[2]{\expandafter\gdef\csname fn@#1@PL\endcsname{#2} }
\end{verbatim}
\acrodefplural Same as above, storing content in aux file.
\begin{verbatim}
\newcommand*{\acrodefplural}[1]{\@ifnextchar[\AC@acrodefplurali{#1}]{\AC@acrodefpluralii{#1}}
\end{verbatim}
\begin{verbatim}
\newcommand{\AC@acrodefplurali}{
\def\AC@acrodefplurali#1[#2]#3{\@bsphack\protected@write\@auxout{}{\string\newacroplural{#1}[#2]{#3}}\@esphack }
\end{verbatim}
\begin{verbatim}
\newcommand{\AC@acrodefpluralii}{
\@ifnextchar[\AC@acrodefpluralii{#1}]{\AC@acrodefpluralii{#1}}
\end{verbatim}
\end{verbatim}
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.

\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 \ACS@acs and long forms \ACS@acl of the acronym saved in
\fn@<acronym>.

\ACS@acs

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

\acs

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

\acs

\acs

\acs

\acs

\acs

\acs

\acs

\acs

\acs

\acs

\acs
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)

\newcommand*{\@verridelabel}[1]{% 
  \@bsphack
  \protected@write\@auxout{{\string\AC@undonewlabel{#1}}}% 
  \label{#1}% 
  \AC@overriddenmessage rs{#1}% 
  \@esphack
}\}
\newcommand*{\AC@undonewlabel}{\AC@undonewlabel rs}%
\newcommand*{\AC@undonewlabel}[3]{% 
  \@ifundefined{#1@#3}{% 
    \global\expandafter\let\csname#2@#3\endcsname\@nnil
  }{% 
    \global\expandafter\let\csname#1@#3\endcsname\relax
  }% 
}\newcommand*{\AC@overriddenmessage}[3]{% 
  \expandafter\ifx\csname#2@#3\endcsname\@nnil
  \expandafter\@firstoftwo
  \else
    \@ifundefined{#1@#3}{% 
      \@ifundefined{#2@#3}{\expandafter\@firstoftwo}{}% 
    }{% 
      \expandafter\@secondoftwo
    }% 
  \fi
  \PackageInfo{acronym}{Label '#3' newly defined as it shall be overridden although it is yet undefined}% 
  \global\expandafter\let\csname#2@#3\endcsname\empty
}%
\PackageInfo{acronym}{Label ‘#3’ overridden}%
@ifundefined{#2@#3}{%
        \global\expandafter\let\csname#2@#3\endcsname\empty}{}%
        \expandafter\g@addto@macro\csname#2@#3\endcsname{i}%
    }%
}%
\newcommand*{\AC@testdef}[3]{%
    \@ifundefined{s@#2}
        \@secondoftwo\@firstofone
        {%
            \expandafter\ifx\csname s@#2\endcsname\empty
            \expandafter\@firstofone
            \else
            \expandafter\xdef\csname s@#2\endcsname{%
            \expandafter\expandafter
            \expandafter\expandafter
            \expandafter\@gobble
            \csname s@#2\endcsname
            \expandafter\@gobble
            }%
            \fi
        }%
        \@testdef{#1}{#2}{#3}%
    }%
}%
\AtBeginDocument{\immediate\write\@auxout{\string\AC@reset@newl@bel}}
\newcommand*{\AC@reset@newl@bel}{%
    \ifx\@newl@bel\@testdef
        \let\@newl@bel\AC@testdef
        \let\AC@undonewlabel\@gobble
    \fi
}%
\newcommand*{\AC@placelabel}[1]{%
    \expandafter\ifx\csname AC@\AC@prefix#1\endcsname\AC@used
    \else
    {\AC@phantomsection\@verridelabel{acro:#1}}%
    \ifAC@starred\else%
    \global\expandafter\let\csname AC@\AC@prefix#1\endcsname\AC@used
    \fi%
    \AC@addtoAC@clearlist{#1}%
    \fi
}%
\acf
\acfa
\@acf
\Acf
\Acfa
\@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.
\@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.
\@Acf
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. If the option \footnote is not specified, the optional variable determines the penalty for
The first time an acronym is accessed, its Full Name (FN) is printed. The next time just (FN) is used. When the *footnote* option is used, the short form (FN) is always used. The optional variable is being passed to \acf, in case it is used.
\iac \@iac \@iaci \Iac \@Iac

Indefinite article correct expansion. The optional variable is being passed to \ac.

\acsp \acspa \@acsp

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

\@acsp
The user macro `\aclp` prints the plural full name of the acronym. 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`. If the option `footnote` is not specified, the optional variable determines the penalty for a line break.
The first time an acronym is accessed Full Names (FNs) is printed. The next time just (FNs). The optional variable is being passed to \acfp, in case it is used.

\acp
\@acp
\Acp
\@Acp

The Full Name is printed in italics and the abbreviated is printed in upshape. The optional variable determines the penalty for a line break.

\acfi
\acfia
\Acfi
\Acfia
\newcommand{\acfi}[2][\AC@linebreakpenalty]{% \	exorpdfstring{\protect\@acfi[#1]{#2}}{{\AC@Acl{#2}} (#2)}}
\newcommand*{\@acfi}[2][\AC@linebreakpenalty]{% \acffont{% \AC@placelabel{#2}{\itshape\AC@acl{#2}}% \nolinebreak[#1] \% \acfsfont{\acsfont{\AC@acs{#2}}}}% \ifAC@starred\else\AC@logged{#2}\fi}
\newcommand{\acfi}[2][\AC@linebreakpenalty]{% \acffont{% \AC@placelabel{#2}{\itshape\AC@acl{#2}}% \nolinebreak[#1] \% \acfsfont{\acsfont{\AC@acs{#2}}}}% \ifAC@starred\else\AC@logged{#2}\fi}
\acused Marks the acronym as used. Don’t confuse this with \acronymused!
\newcommand{\acused}[1][1]{% \global\expandafter\let\csname AC@\AC@prefix#1\endcsname\AC@used% \AC@addtoAC@clearlist{#1}}
\acsu Print the short form of the acronym and mark it as used.
\newcommand*{\acsu}{\AC@starredfalse\protect\acsua}\WithSuffix\newcommand\acsu*{\AC@starredtrue\protect\acsua} \newcommand{\acsua}[1]{% \ifAC@starred\acs*{#1}\else\acs{#1}\fi\acused{#1}}
\aclu Print the long form of the acronym and mark it as used.
\newcommand*{\aclu}{\AC@starredfalse\protect\aclua}\WithSuffix\newcommand\aclu*{\AC@starredtrue\protect\aclua} \newcommand{\aclua}[1]{% \ifAC@starred\acl*{#1}\else\acl{#1}\fi\acused{#1}}
\Aclu\Aclua\endinput

That’s it.