ACRO

v1.2   2013/02/09

Typeset Acronyms

Clemens NIEDERBERGER

https://bitbucket.org/cgnieder/acro/
contact@mychemistry.eu

English documentation

ACRO not only allows you to create acronyms in a simple way but also lets you add them to different classes of acronyms. Lists can be created of separate classes wherever you want the list to appear.
ACRO also provides an option `single` which ignores acronyms that are used only once in the whole document.
As an experimental feature ACRO also offers the option `sort` which automatically sorts the list created by `\printacronyms`.

Contents

1. Licence and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the LaTeX Project Public License, version 1.3 or later (`http://www.latex-project.org/lppl.txt`). The package has the status "maintained."

ACRO loads and needs the following packages: expl3,[1] xparse, xtemplate, l3keys2e[2] and translations.[3]

2. About

> Acronyms and initialisms are abbreviations formed from the initial components in a phrase or a word. These components may be individual letters (as in CEO) or parts of words (as in Benelux and Ameslan). There is no universal agreement on the precise definition of the various terms nor on written usage.   [*Wik12a*]

After WIKIPEDIA told us what acronyms are and we won't confuse them with units or other kinds of abbreviations – why would we need another package for them? There are several already: acronym[4] [Oet12], acromake[5] [Tho95], acroterm[6] [Vos10], the abbreviations package abbrevs[7] [Swi01] (the current version 1.4 has a bug,[8] though), the nomenclature package nomencl[9] [Vey+05], and of course the mighty glossaries[10] [Tal12]. So there is really no *need* for a new package.

On the other hand acronym, the best of the acronym specific packages, has one or two shortcomings and sometimes using glossaries seems a bit of an overkill (or simply inconvenient as one has to run `makeglossaries`, `makeindex` or `xindy`, then[11]). So ACRO stands somewhere in between (but closer to acronym).

The main reason for the existance of ACRO is a question on {TeX}.SE[12] which intrigued me and in consequence led to ACRO and it's option `single`.

ACRO has many similarities with the acronym package. In fact, quite some macros have the same name and meaning.[13]

Please take a minute to think and decide which package will suit your needs best. Are you planning to add a glossary to your book? You should probably go with glossaries, then. Are you planning to add a nomenclature? You may want to choose nomencl and so on...

---

[1] CTAN: l3kernel   [2] All three part of the l3packages bundle   [3] Part of the exsheets bundle.
[4] CTAN: acronym   [5] CTAN: acromake   [6] CTAN: acroterm   [7] CTAN: abbrevs   [8] see `http://tex.stackexchange.com/q/59840/5049` for solutions.   [9] CTAN: nomencl   [10] CTAN: glossaries   [11] Rumour has it there is going to be a version that can be used without running an external program [12] `http://tex.stackexchange.com/q/59449/5049`   [13] *Not* in the sense of `\meaning`!

3. Updating from Version 0.*

Version 1.0 introduces a major syntax change. In order to keep documents using a prior version compilable ACRO introduces the option `version = 0`. It provides the acronym defining commands with the old syntax and also all the additional commands like `\DeclareAcronymFormat` that have been dropped in version 1.0. Also the old syntax of `\printacronyms` is restored. Versions 0.* also set labels when an acronym was used the first time, version 1.0 doesn't do that any more. `version = 0` sets them as expected.

Note that `version = 0` *does not and will not* support new features of any kind.

With version 1.0 acronyms are declared with a key/value syntax similar to the way in package glossaries glossary entries are defined. This was made for a good reason: in the beginning the old `\DeclareAcronym` already had lots of arguments that were hard to remember. Every new feature needed to introduce further commands so that in version 0.6 there have been *four* different commands already. All this isn't necessary any more and there is only one command with only two arguments.

4. Basics

4.1. Creating New Acronyms

Changed in version 1.0

Acronyms are created with the command `\DeclareAcronym` that can only be used in the preamble.

`\DeclareAcronym{<id>}{<list of keys>}`

This command understands a number of keys which are listed below. Some of them are not described immediately but at appropriate places in the documentation.

`short = <text>`
the short form of the acronym.

`long = <text>`
the long form of the acronym.

`short-plural = <text>`   Default: s
the plural ending appended to the short form.

`long-plural = <text>`   Default: s
the plural ending appended to the long form.

`long-plural-form = <text>`
plural long form of the acronym; replaces the long form when used instead of appending the plural ending.

`short-indefinite = <text>`   Default: a

Introduced in version 1.2

indefinite article for the short form.

3

`long-indefinite = <text>`   Default: a
  indefinite article for the long form.

`long-pre = <text>`
  <text> is prepended to the long form in the text but not in the list of acronyms.

`long-post = <text>`
  <text> is appended to the long form in the text but not in the list of acronyms.

`alt = <text>`
  alternative short form.

`alt-indefinite = <text>`   Default: a
  indefinite article for the alternative form.

`extra = <text>`
  extra information to be added in the list of acronyms.

`sort = <text>`
  if used the acronym will be sorted according to this key instead of its ID.

`class = <text>`
  the class the acronym belongs to.

`cite = [<prenote>][<postnote>]{citation keys}`
  a citation that is printed to the acronym according to an option explained later.

`short-format = <TEX code>`
  the format used for the short form of the acronym.

`long-format = <TEX code>`
  the format used for the long form of the acronym.

`first-long-format = <TEX code>`
  the format used for the first long form of the acronym as set with \ac, \acf or \acflike and
  their uppercase, plural and indefinite forms.

`pdfstring = <text>/<plural ending>`
  used as PDF string replacement in bookmarks when used together with the hyperref package.
  The appended plural ending is optional. If you leave it (*and* the /) the default ending is used.

`accsupp = <text>`
  sets the ActualText key as presented by the accsupp package for the acronym.

`index-sort = <text>`
  If you use the package option index every occurrence of an acronym is recorded to the index
  and sorted by its ID or (if set) by the value of the sort key. This key allows to set an individual
  sorting option for the index. See section 5.6 for details.

`index = <text>`

This key allows to overwrite the automatic index entry with an arbitrary one. See section 5.6
for details.

`index-cmd = <text>`

This key let's you set an individual index creating command for this acronym. It should be a
command that takes one mandatory argument. See section 5.6 for details.

In its simplest form an acronym needs a short and a long form. Please note that both keys
must be set and that the short key *must* always be the *first* key that is set.

```
1  % preamble:
2  \DeclareAcronym{test}{
3    short = ST ,
4    long  = Some Test
5  }
```

This creates the acronym "ST" with the ID "test" and the long form "Some Test."
The format key allows you to choose a specific format for the short form of an acronym:

```
1  % preamble:
2  \DeclareAcronym{ot}{
3    short        = ot ,
4    long         = Other Test ,
5    short-format = \scshape
6  }
```

The short form now looks like this: OT.
The cite key needs a bit explaining. It expects arguments like the standard \cite com-
mand, i.e., two optional arguments setting the `<prenote>` and `<postnote>` and one mandatory
argument setting the citation key.

```
1   % preamble:
2   \DeclareAcronym{ny}{
3     short        = NY ,
4     short-plural = ,
5     long         = New York ,
6     long-plural  = ,
7     cite         = {NewYork}
8   }
9   % bib file for use with biber/biblatex:
10  @online{NewYork,
11    author  = {Wikipedia},
12    title   = {New York City},
13    urldate = {2012-09-27},
14    url     = {http://en.wikipedia.org/wiki/New_York_City},
```

```
15    year    = {2012}
16  }
```

The first appearance now looks as follows[14]: New York (NY) [Wik12b].

4.2. Using the Acronyms – the Commands

Acronyms are used with with one of the following commands:

`\ac*{<id>}`
> basic command; the first output is different from subsequent ones.

`\Ac*{<id>}`
> same as `\ac` but capitalizes the first letter of the long form.

`\acs*{<id>}`
> **s**hort form; the actual acronym.

`\acl*{<id>}`
> **l**ong form; the meaning of the acronym.

`\Acl*{<id>}`
> same as `\acl` but capitalizes first letter.

`\aca*{<id>}`
> **a**lternative short form as specified in the `alt` key of `\DeclareAcronym`; if it hasn't been specified this is identical to `\acs`.

`\acf*{<id>}`
> first form; output like the first time `\ac` is output.

`\Acf*{<id>}`
> same as `\acf` but capitalizes first letter of the long form.

`\acp*{<id>}`
> **p**lural form of `\ac`;

`\Acp*{<id>}`
> same as `\acp` but capitalizes first letter of the long form.

`\acsp*{<id>}`
> plural form of `\acs`;

`\aclp*{<id>}`
> plural form of `\acl`;

`\Aclp*{<id>}`
> same as `\aclp` but capitalizes first letter.

---

[14] The appearance of the citation of course depends on the citation style you're using.

`\acap*{<id>}`
  plural form of `\aca`;

`\acfp*{<id>}`
  plural form of `\acf`;

`\Acfp*{<id>}`
  same as `\acfp` but capitalizes first letter of the long form.

If an acronym is used the first time with `\ac` its output is different from subsequent uses. To be clear on this: the first time! If the acronym has been used with *any* of the output commands before it is *not* the first time any more.

If you use the starred variant an acronym will not be marked as used. This proves useful if an acronym is typeset in a section title, for example, since then the appearance in the table of contents won't mark it as used.

```
1  % preamble:
2  % \DeclareAcronym{cd}{
3  %    short        = cd ,
4  %    long         = Compact Disc ,
5  %    short-format = \scshape
6  % }
7  first time: \ac{cd} \\
8  second time: \ac{cd} \\
9  short: \acs{cd} \\
10 alternative: \aca{cd} \\
11 first again: \acf{cd} \\
12 long: \acl{cd} \\
13 short plural: \acsp{cd} \\
14 long plural: \aclp{cd}
```

first time: Compact Disc (CD)
second time: CD
short: CD
alternative: CD
first again: Compact Disc (CD)
long: Compact Disc
short plural: CDS
long plural: Compact Discs

4.3. Plural Forms

If an acronym is defined in the standard way ACRO uses an 's' that's appended to both the short and the long form when one of the plural commands is used. However, that is not always the best solution. For one thing not all acronyms may have a plural form. Second, the plural form especially of the long forms may be formed differently. And third, other languages can have other plural endings.

    For these reasons `\DeclareAcronym` can get the following keys:

`short-plural = <text>`

`long-plural = <text>`

`long-plural-form = <text>`

These keys are optional. If they're not used, the default setting is 's'. If you use `long-plural-form` the long form will be replaced by the specified plural form when necessary.

    Suppose we define the following acronyms:

```
1  \DeclareAcronym{cd}{
2    short       = cd ,
3    long        = Compact Disc ,
4    short-format = \scshape
5  }
6  \DeclareAcronym{ny}{
7    short       = NY ,
8    short-plural = ,
9    long        = New York ,
10   long-plural  =
11 }
12 \DeclareAcronym{sw}{
13   short       = SW ,
14   long        = Sammelwerk ,
15   long-plural = e
16 }
17 \DeclareAcronym{mp}{
18   short           = MP ,
19   long            = Member of Parliament ,
20   long-plural-form = Members of Parliament
21 }
```

These acronyms now have the following plural appearances:

```
1  \acsp{cd}, \aclp{cd} \\          CDs, Compact Discs
2  \acsp{ny}, \aclp{ny} \\          NY, New York
3  \acsp{sw}, \aclp{sw} \\          SWs, Sammelwerke
4  \acsp{mp}, \aclp{mp}             MPs, Members of Parliament
```

4.4. Alternative Short Forms

For some acronyms it might be useful to have alternative forms. For this `\DeclareAcronym` has another key:

```
alt = <text>
```

```
1  % preamble:
2  % \DeclareAcronym{jpg}{
3  %   short = JPEG ,
4  %   alt   = JPG ,
5  %   long  = Joint Photographic Experts Group
6  % }
7  default: \acs{jpg} \\
8  alt.: \aca{jpg}
```

8

> default: JPEG
> alt.: JPG

The alternative form uses the same plural ending as the default short form and is formatted in the same way.

4.5. Extra Information for the List Entry

Of course you can print a list of acronyms where their meaning is explained. Sometimes it can be useful to add additional information there. This is done with another key to `\DeclareAcronym`:

`extra = <text>`

These information will only be displayed in the list. See section 6 for the impact of the following example.

```
1  % preamble:
2  % \DeclareAcronym{nato}{
3  %   short       = nato ,
4  %   long        = North Atlantic Treaty Organization ,
5  %   extra       = \textit{deutsch}: Organisation des Nordatlantikvertrags
      ,
6  %   short-format = \scshape
7  % }
8  The \ac{nato} is an intergovernmental military alliance based on the
9  North Atlantic Treaty which was signed on 4~April 1949. \ac{nato}
10 headquarters are in Brussels, Belgium, one of the 28 member states
11 across North America and Europe, the newest of which, Albania and
12 Croatia, joined in April 2009.
```

> The North Atlantic Treaty Organization (NATO) is an intergovernmental military alliance based on the North Atlantic Treaty which was signed on 4 April 1949. NATO headquarters are in Brussels, Belgium, one of the 28 member states across North America and Europe, the newest of which, Albania and Croatia, joined in April 2009.

5. Additional Commands and Possibilities

5.1. Indefinite Forms

Introduced in version 1.2

Unlike many other languages[15] in English the indefinite article is not determined by the grammatical case, gender or number but by the pronounciation of the following word. This means that the short and the long form of an acronym can have different indefinite articles. For these cases ACRO offers the keys `short-indefinite`, `alt-indefinite` and `long-indefinite` whose default is a. For every lowercase singular command two alternatives exist, preceded by i and I, respectively, which output the lowercase and uppercase version of the corresponding indefinite article.

---

[15] Let's better say: unlike the other languages where I know at least the basics.

```
1  % preamble:
2  % \DeclareAcronym{ufo}{
3  %   short         = UFO ,
4  %   long          = unidentified flying object ,
5  %   long-indefinite = an
6  % }
7  \Iac{ufo}; \iacs{ufo}; \iacl{ufo}
```

An unidentified flying object (UFO); a UFO; an unidentified flying object

5.2. Simulating the First Appearance

Users told me[16] that there are cases when it might be useful to have the the acronym typeset according to the `first-style` but with another text than the long form. For such cases ACRO offers the following commands.

`\acflike*{<id>}{<instead of long form>}`

`\acfplike*{<id>}{<instead of long form>}`

```
1  \acsetup{first-style=footnote}
2  \acflike{ny}{the big apple}
```
NY[a]
_____
[a] the big apple [Wik12b]

The plural ending in `\acfplike` is only appended to the short form. It makes no sense to append it to the text that is inserted manually anyway. Note that whatever text you're inserting might be gobbled depending on the `first-style` you're using.

5.3. Using Classes

The acronyms of ACRO can be divided into different classes. This doesn't change the output but allows different acronym lists, see section 6. For this `\DeclareAcronym` has an additional key:

`class = <text>`

This might be useful if you can and want to divide your acronyms into different types, technical and grammatical ones, say, that shall be listed in different lists.

_____

[16] Well – one, to be precise ;)

```
1  % preamble:
2  % \DeclareAcronym{la}{
3  %   short       = LA ,
4  %   short-plural = ,
5  %   long        = Los Angeles ,
6  %   long-plural  = ,
7  %   class       = city
8  % }
9  % \DeclareAcronym{ny}{
10 %   short       = NY ,
11 %   short-plural = ,
12 %   long        = New York ,
13 %   long-plural  = ,
14 %   class       = city ,
15 %   cite        = NewYork
16 % }
17 \acl{la} (\acs{la}) \\
18 \acl{ny} (\acs{ny})
```
Los Angeles (LA)
New York (NY)

5.4. Reset or Mark as Used

If you want for some reason to fool ACRO into thinking that an acronym is used for the first
time you can call one of these commands:

`\acreset{<comma separated list of ids>}`

`\acresetall`

```
1  \acreset{ny}\ac{ny}                    New York (NY) [Wik12b]
```

Beware that both commands act *globally*! There are also commands that effectively do the
opposite of `\acreset`, i.e., mark acronyms as used:

`\acuse{<comma separated list of ids>}`

`\acuseall`

5.5. \ac and Friends in PDF Bookmarks

ACRO's commands usually are not expandable which means they'd leave unallowed tokens in
PDF bookmarks. hyperref offers `\texorpdfstring` to circumvent that issue manually but that
isn't really a nice solution. What's the point of having macros to get output for you if you have
to specify it manually after all?

That is why ACRO offers a preliminary solution for this. In a bookmark every `\ac` like
command falls back to a simple text string typesetting what `\acs` would do (or `\acsp` for plural
forms). These text strings both can accessed manually and can be modified to an output reserved
for PDF bookmarks.

`\acpdfstring{<id>}`
access the text string used in PDF bookmarks.

`\acpdfstringplural{<id>}`
access the plural form of the text string used in PDF bookmarks.

`pdfstring = <pdfstring>/<plural ending>`
Key for `\DeclareAcronym` to declare a custom text string for PDF bookmarks. The plural ending can be set optionally.

`accsupp = <text>`

Key for `\DeclareAcronym` to set the `ActualText` property of `\BeginAccSupp` (see accsupp's documentation for details) to be used for an acronym. It only has an effect when the package option `accsupp` is used, too.

For example the PDF acronym used in the title for this section is defined as follows:

```
1  \DeclareAcronym{pdf}
2  {
3    short    = pdf ,
4    long     = Portable Document Format ,
5    format   = \scshape ,
6    pdfstring = PDF ,
7    accsupp  = PDF
8  }
```

This also demonstrates the `accsupp` key. For this to work you need to use the *package option* `accsupp`, too, which will load the package accsupp. Then the key `accsupp` will set the `ActualText` property of `\BeginAccSupp`. Please refer to accsupp's documentation for details. To see its effect copy PDF and paste it into a text file. You should get uppercase letters instead of lowercase ones.

5.6. Adding Acronyms to the Index

ACRO has the package option `index`. If it is used an index entry will be recorded every time an *unstarred* acronym command is used. The index entry will be `<id>@<short>`, `<sort>@<short>` if the `sort` key has been set, `<index-sort>@<short>`if the `index-sort` has been set, or `<index>` if the key `index` has been set for the specific acronym. The short version appearing there are formatted according to the chosen format, of course.

This document demonstrates the feature. You can find every acronym that has been declared in the index. In order to allow flexibility the indexing command can be chosen both globally via package option and individually for every acronym. This would allow to add acronyms to a specific index if more than one index is used, for example with help of the imakeidx[17] package.

I'm not yet convinced this is a feature many people if anyone needs and if they do if it is flexible enough. If you have any thoughts on this I'd appreciate an email.

---

[17] CTAN: imakeidx

6. Printing the List

Printing the whole list of acronyms is easy: just place `\printacronyms` wherever you want the list to be.

`\printacronyms[<options>]`

The commands takes a few options, namely the following ones:

`include-classes = <list of classes>`
takes a comma-separated list of the classes of acronyms that should be in the list.

`exclude-classes = <list of classes>`
takes a comma-separated list of the classes of acronyms that should *not* be in the list.

`name = <name of the list>`
sets the name for the list.

`header = <sectioning command without leading backslash>`   Default: section*
sets the sectioning command for the header of the list.

```
1  \acsetup{extra-style=comma}
2  \printacronyms[exclude-classes=city]
3
4  \printacronyms[include-classes=city,name={City Acronyms}]
```

Acronyms

CD Compact Disc

ID identification string

JPEG Joint Photographic Experts Group

MP Member of Parliament

NATO North Atlantic Treaty Organization, *deutsch*: Organisation des Nordatlantikvertrags

PDF Portable Document Format

SW Sammelwerk

ST Some Test

UFO unidentified flying object

City Acronyms

LA Los Angeles

NY New York

You can see that the default layout is a `description` list with a `section*` title. Both can be changed, see section 7.

The command `\printacronyms` needs two LaTeX runs. This is a precaution to avoid error messages with a possibly empty list. But since almost all documents need at least two runs and often are compiled much more often than that, this fact shouldn't cause too much inconvenience.

7. Options and Customization

7.1. General Options

There are a few options which change the general behaviour of ACRO. Underlined values are used if no value is given.

`version = 0|1`   Default: `1`
> Provide backwards compatibility for documents set with ACRO in a version prior to v1.0.

`single = true|false`   Default: `false`
> If set to `true` an acronym that's used only once '(with `\ac`) in a document will only print the long form and will not be printed in the list.

`hyperref = true|false`   Default: `false`
> If set to `true` the short forms of the acronyms will be linked to their list entry.

`record-pages = true|false`   Default: `true`
> Since ACRO can handle arabic, roman and Roman page numbers but *not* any other kind of numbering this option enables to turn the page number recording of for these cases as it would lead to errors else. This affects the whole document and can only be set in the preamble! It means you cannot have page numbers in the list of acronyms in this case.

`only-used = true|false`   Default: `true`
> This option is `true` as default. It means that only acronyms that are actually used in the document are printed in the list. If `false`, all acronyms defined with `\DeclareAcronym` will be written to the list.

`mark-as-used = first|any`   Default: `any`

> This option determines wether an acronym is mark as used when the *first* form is used the first time (with `\ac`, `\acf` or `\acflike` and their uppercase, plural and indefinite forms) or when any of the `\ac`-like commands is used.

macros = <u>true</u>|false   Default: `false`

If set to `true` this option will create a macro `\<id>` for each acronym as a shortcut for `\ac{<id>}`. Already existing macros will *not* be overwritten.

xspace = <u>true</u>|false   Default: `false`

If set to `true` this option will append `\xspace` from the xspace[18] package to the commands created with the `macros` option.

strict = <u>true</u>|false   Default: `false`

If set to `true` and the option `macros = true` is in effect then already existing macros will be overwritten.

sort = <u>true</u>|false   Default: `true`

If set to `true` the acronym list will be sorted automatically. The entries are sorted by their ID ignoring upper and lower case. This option needs the experimental package l3sort (from the l3experimental[19] bundle) and can only be set in the preamble.

cite = <u>all</u>|first|none   Default: `first`

This option decides whether citations that are added via `\DeclareAcronymCitation` are added to each first, every or no appearance of an acronym.

cite-cmd = <cmd>   Default: `\cite`

This option determines which command is used for the citation. Each citation command that takes the cite key as argument is valid, for example biblatex's[20] `\footcite`.

cite-space = <code>   Default: `\nobreakspace`

Depending on the citation command in use a space should be inserted before the citation or maybe not (e.g. `\footcite`…). This option allows you to set this. Actually it can be used to place arbitrary code right before the citation.

index = <u>true</u>|false   Default: `false`

If set to `true` an index entry will be recorded every time an *unstarred* acronym command is used for the corresponding acronym.

index-cmd = <cmd>   Default: `\index`

Chooses the index command that is used when option `index` has been set to `true`.

accsupp = <u>true</u>|false   Default: `false`

Activates the access support as provided by the accsupp package.

uc-cmd = <cmd>   Default: `\acro_first_upper_case:n`

The command that is used to capitalize the first word in the `\Ac` and the like commands. You can change it to another one like for example `\makefirstuc`[21] or `\MakeTextUppercase`.[22]

All options of this and the following sections can be set up either as package options or via the setup command:

---

[18] CTAN: xspace   [19] CTAN: l3experimental   [20] CTAN: biblatex   [21] from the mfirstuc package   [22] from the textcase package

`\acsetup{<options>}`

set up ACRO anywhere in the document. Or separate package loading from setup.

```
1  % with \acsetup{macros}
2  we could have used these before: \nato, \ny
```

we could have used these before: NATO, NY

7.2. Options Regarding Acronyms

The options described in this section all influence the layout of one of the possible output forms of the acronyms.

`short-format = <format>`   (initially empty)

Sets a format for all short forms. For example `short-format = \scshape` would print all short forms in small caps.

`long-format = <format>`   (initially empty)

The same for the long forms.

`first-long-format = <format>`   (initially empty)

The format for the long form on first usage (with `\ac`, `\acf` or `\acflike` and their uppercase, plural and indefinite forms).

`list-short-format = <format>`   (initially empty)

An extra format for the short entries in the list. If not used this is the same as `short-format`. Please be aware that a call of `short-format` after this one will overwrite it again.

`list-long-format = <format>`   (initially empty)

An extra format for the long entries in the list. If not used this is the same as `long-format`. Please be aware that a call of `long-format` after this one will overwrite it again.

`extra-format = <format>`   (initially empty)

The same for the extra information.

`first-style = default|plain|empty|square|short|reversed|plain-reversed|`
`footnote|sidenote`   Default: `default`

The basic style of the first appearance of an acronym. The value `sidenote` needs the command `\sidenote` to be defined for example by the sidenotes[23] package.

`extra-style = default|plain|comma|paren|bracket`   Default: `default`

Defines the way the extra information is printed in the list.

`plural-ending = <tokenlist>`   Default: `s`

With this option the default plural ending can be set.

---

[23] CTAN: sidenotes

```
1  % (Keep in mind that we're in
2  % a minipage here!)
3  \acsetup{first-style=empty}
4  \acf{ny} \\                              NY
5  \acsetup{first-style=footnote}           NYᵃ
6  \acf{ny} \\                              New York [NY] [Wik12b]
7  \acsetup{first-style=square}             NY [Wik12b]
8  \acf{ny} \\                              NY (New York) [Wik12b]
9  \acsetup{first-style=short}              New York – NY [Wik12b]
10 \acf{ny} \\                              NY – New York [Wik12b]
11 \acsetup{first-style=reversed}
12 \acf{ny} \\                              ─────────────
13 \acsetup{first-style=plain}              ᵃ New York [Wik12b]
14 \acf{ny} \\
15 \acsetup{first-style=plain-reversed}
16 \acf{ny}
```

7.3. Options Regarding the List

page-ref = none|plain|comma|paren   Default: none
> If this option is set to a value other than none the page numbers of the an acronym appeared on are printed in the list. Please note that this is an experimental feature and might fail in quite a number of cases. If you notice anything please send me an email!

page-name = <page name>   Default: p.\@\,
> The "name" of the page label. This is automatically translated to the active language. However for the time being there are many translations missing, yet. Please notify me if you find your language missing.

page-name = <page name plural>   Default: pp.\@\,

Introduced in version 1.0
> The "name" of the page label when there are more than one page. This is automatically translated to the active language. However for the time being there are many translations missing, yet. Please notify me if you find your language missing.

page-ranges = <u>true</u>|false   Default: true

Introduced in version 1.0
> If set to false ranges will be set by appending f or ff, respectively.

next-page = <text>   Default: \,f.\@

Introduced in version 1.0
> Appended to a page number when page-ranges is set to false and the range is only 2 pages long. This is automatically translated to the active language. However, for the time being there are many translations missing, yet. Please notify me if you find your language missing.

next-pages = <text>   Default: \,ff.\@

Introduced in version 1.0
> Appended to a page number when page-ranges is set to false and the range is more than 2 pages long. This is automatically translated to the active language. However, for the time being there are many translations missing, yet. Please notify me if you find your language missing.

`list-type = table|<list>`   Default: `description`
> This option let's you choose how the list is printed. `<list>` can be any valid list like `itemize` or `description`.

`list-style = list|tabular|longtable|extra-tabular|extra-longtable|`
`extra-tabular-rev|extra-longtable-rev`   Default: `list`
> If you choose `list-type = table` you have to specify which kind of table should be used. If you choose `longtable`, `extra-longtable` or `extra-longtable-rev` you have to load longtable[24] in your preamble. The values `extra-longtable` and `extra-longtable-rev` put the extra information in a column of its own.

`list-header = chapter|chapter*|section|section*|subsection|subsection*|`
`addchap|addsec`   Default: `section*`
> The header type of the list. The last two only work with a KOMA-Script class that also defines the appropriate command.

`list-name = <list name>`   Default: `Acronyms`
> The name of the list. This is what's written in the list-header. This is automatically translated to the active language. However, for the time being there are many translations missing, yet. Please notify me if you find your language missing.

`list-table-width`   Default: `.7\linewidth`
> This has only an effect if you chose `list-type = table`. The second column (or the third if you choose one of the `-rev` styles) of the table is a `p` column whose width can be specified with this option.

`list-caps = `<u>`true`</u>`|false`   Default: `false`
> Print the first letters of the long form capitalized.

  If you for example have loaded enumitem[25] you can define a custom list for the acronym list:

```
% preamble:
% \usepackage{enumitem}
\newlist{acronyms}{description}{1}
\newcommand*\addcolon[1]{#1:}
\setlist[acronyms]{labelwidth=3em,leftmargin=3.5em,noitemsep,
  itemindent=0pt,font=\addcolon}
\acsetup{list-type=acronyms,hyperref=false,extra-style=comma}
\printacronyms
```

City Acronyms

CD: Compact Disc

---

[24] CTAN: longtable   [25] CTAN: enumitem

ID:	identification string
JPEG:	Joint Photographic Experts Group
LA:	Los Angeles
MP:	Member of Parliament
NATO:	North Atlantic Treaty Organization, *deutsch*: Organisation des Nordatlantikvertrags
NY:	New York
PDF:	Portable Document Format
SW:	Sammelwerk
ST:	Some Test
UFO:	unidentified flying object

8. About Page Ranges

If you enable the `page-ref` option ACRO adds page numbers to the list of acronyms. In version 0.*
it would add a page reference for an acronym in the list of acronyms that used `\pageref` to
refer to the first appearance of an acronym. This is retained using `version = 0`. Version 1.0
uses a different approach that doesn't use a label but instead will list *all* pages an acronym
appeared on. With hyperref the pages are referenced using `\hyperpage`.

There are some options that control how this list will be typeset, e.g., `page-ranges`, `next-pages`
or the option `page-ref` itself. It is important to mention that the page list will always take two
compilation runs until changes in the options or the actual page numbers affect it. This is due
to the fact that the updated sequence is first written to the aux file and only read in during the
next run.

There is another important fact: ACRO can only handle arabic and lowercase and uppercase
roman page numbering. If you happen to choose a different numbering style you either have to
use the package option `record-pages = false` which means you will have to pass on page
numbers in the list of acronyms, or you need to make sure that you only use the *starred* versions
of the `\ac` commands on these pages so acronyms don't count as used there!

9. hyperref Support

The option `hyperref = true` adds internal links from all short (or alternative) forms to their
respective list entries. Of course this only works if you have loaded the hyperref package in your
preamble. You should use this option with care: if you don't use `\printacronyms` anywhere
this option will result in loads of hyperref warnings. Also printing several lists can result in
warnings if don't clearly separate the lists into different classes. If an acronym appears in more
than one list there will also be more than one hypertarget for this acronym.

Using hyperref will also add `\hyperpage` to the page numbers in the list (provided they are
displayed in the style chosen). Like with an index the references will thus not point to the
acronyms directly but to the page they're on.

A. Commands provided in version 0.*

Prior to version 1.0 the following commands were available. They are still provided if you set the option `version = 0`. They will be mentioned here shortly but they won't be explained any more. If you'd like a more thorough description ask me for the documentation to version 0.6a.

`\DeclareAcronym*{<id>}{<short>,<pl>}[<alt.>]{<long>,<pl>}{<extra>}[<class>]`
` % can be used only in preamble`

`\DeclareAcronymFormat{<id>}{<format>}` ` % can be used only in preamble`

`\DeclareAcronymCitation{<id>}[<pre>][<post>]{<cite keys>}`
` % can be used only in preamble`

`\DeclareAcronymPDFString{<id>}{<pdf entry>,<plural ending>}`
` % can be used only in preamble`

B. All Acronyms Used in this Documentation

```
1  \acsetup{
2    list-type   = table,
3    list-style  = longtable,
4    list-header = subsection*,
5    extra-style = comma,
6    page-ref    = comma
7  }
8  \printacronyms[name=All Acronyms]
```

All Acronyms

CD	Compact Disc, pp. 7, 8
ID	identification string, pp. 4, 5, 15
JPEG	Joint Photographic Experts Group, p. 8
LA	Los Angeles, p. 10
MP	Member of Parliament, p. 8
NATO	North Atlantic Treaty Organization, *deutsch*: Organisation des Nordatlantikvertrags, pp. 9, 16
NY	New York, pp. 6, 8, 10, 11, 16
PDF	Portable Document Format, pp. 4, 11, 12
SW	Sammelwerk, p. 8
ST	Some Test, p. 5
UFO	unidentified flying object, p. 9

```
1  \acsetup{
2    list-type   = table,
3    list-style  = longtable,
4    list-header = subsection*,
5    page-ref    = comma
6  }
7  \printacronyms[include-classes=city,name=City Acronyms]
```

City Acronyms

LA   Los Angeles, p. 10
NY   New York, pp. 6, 8, 10, 11, 16

References

[Oet12]   Tobias Oetiker. acronym – An Acronym Environment for LaTeX 2ε. Version 1.38,
           Oct. 29, 2012. URL: http://www.ctan.org/pkg/acronym.

[Swi01]   Matt Swift. abbrevs. Version 1.4, Sept. 8, 2001. URL: http://www.ctan.org/pkg/
           abbrevs.

[Tal12]   Nicola L.C. Talbot. glossaries. Version 3.04, Nov. 18, 2012. URL: http://www.ctan.
           org/pkg/glossaries.

[Tho95]   Paul A. Thompson. acromake. Version na, July 16, 1995. URL: http://www.ctan.
           org/pkg/acromake.

[Vey+05]   Boris Veytsman, Bernd Schandl, Lee Netherton, and C.V. Radhakrishnan. nomencl
           – A Package to Create a Nomenclature. Version 4.2, Sept. 22, 2005. URL: http:
           //www.ctan.org/pkg/nomencl.

[Vos10]   Jakob Voss. acroterm. Version 0.1, Nov. 19, 2010. URL: http://www.ctan.org/
           pkg/acroterm.

[Wik12a]   Wikipedia. *Acronym and initialism*. 2012. URL: http://en.wikipedia.org/wiki/
           Acronyms (visited on 06/21/2012).

[Wik12b]   Wikipedia. *New York City*. 2012. URL: http://en.wikipedia.org/wiki/New_
           York_City (visited on 09/27/2012).

Index

Section titles are indicated **bold**, packages sans serif, commands \brown and options yellow.