

ACRO

v3.0alpha-2 2020/04/09

Typeset Acronyms and other Abbreviations

Clemens NIEDERBERGER

<https://github.com/cgnieder/acro/>

contact@mychemistry.eu

ACRO allows you to define and manage acronyms and abbreviations. It can also be used for glossaries or nomenclatures.

Table of Contents

I. Get started with ACRO	3	6. Alternative short forms	10
1. Licence and requirements	3	7. The first or full appearance	11
2. ACRO for the impatient	3	8. Single appearances of an acronym	12
3. Setting options	4	9. Plural forms and other endings	13
3.1. Load-time options	4	9.1. The plural ending and the plural form	13
3.2. Setup command	4	9.2. Other endings	15
II. Comprehensive description of creation and usage of acronyms	5	10. Indefinite forms	16
4. Declaring acronyms and other abbreviations	5	11. Foreign language acronyms	16
4.1. Basic properties	5	12. Uppercasing	19
4.2. Properties related to plural and indefinite forms	6	13. Printing the list	19
4.3. Properties related to formatting	7	13.1. The main command and its options	19
4.4. Properties related to the created PDF file	8	13.2. Several lists using classes . . .	19
4.5. Further properties	8	13.3. Local lists	19
5. Using acronyms	9	14. Trailing tokens	19
		15. Citing and indexing	19
		15.1. Citing	19

15.2. Indexing	19	19.2. Defining new templates	19
16. Using or resetting acronyms	19	19.3. New acronym templates . . .	19
17. Localisation	19	19.4. New list templates	19
18. Bookmarks and accessibility		19.5. New heading templates	19
support	19	20. Own acronym commands	19
18.1. hyperref support	19		
18.2. PDF comments	19	IV. Appendix	21
18.3. Accessibility support	19	A. Examples	21
III. Extending ACRO	19	B. Acronyms	21
19. Templates	19	C. Index	21
19.1. Pre-defined templates	19		



Hi and thanks that you are testing v3.0 of ACRO before it is released to CTAN. If you want to test the new version use `\usepackage[version=3]{acro}`. With `version=2` or no option at all you get the old version of `acro`. Using `\usepackage[version=3,deprecation]{acro}` is supposed to give as much meaningful warnings and errors as possible.

Part I.

Get started with **ACRO**

1. Licence and requirements

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

2. **ACRO** for the impatient

Acronyms are defined in the preamble via the command

`\DeclareAcronym{<id>}{<properties>}`

where `<id>` is a unique string to identify the acronym and `<properties>` is a key/value list of properties.

In the document acronyms are used with these commands:

`\ac{<id>}` `\Ac{<id>}`

`\ac` prints the acronym `<id>`, the first time with full description and every subsequent use only the abbreviated form. `\Ac` does the same but uppercases the first letter – this may be needed at the beginning of a sentence.

`\acs{<id>}` `\Acs{<id>}`

`\acs` prints the short form of the acronym `<id>`. `\Acs` does the same but uppercases the first letter.

`\acl{<id>}` `\Acl{<id>}`

`\acl` prints the long form of the acronym `<id>`. `\Acl` does the same but uppercases the first letter.

`\acf{<id>}` `\Acf{<id>}`

`\acf` prints the full form of the acronym `<id>`. `\Acf` does the same but uppercases the first letter.

Let’s say you defined CD as follows:

```
1 \DeclareAcronym{cd}{
2   short = CD ,
3   long  = compact disc
4 }
```

Then the usage is

1 <code>\begin{tabular}{ll}</code>		
2 first & <code>\ac{cd}</code> \\\	first	compact disc (CD)
3 second & <code>\ac{cd}</code> \\\	second	CD
4 long & <code>\acl{cd}</code> \\\	long	compact disc
5 short & <code>\acs{cd}</code> \\\	short	CD
6 full & <code>\acf{cd}</code> \\\	full	compact disc (CD)
7 <code>\end{tabular}</code>		

3. Setting options

3.1. Load-time options

ACRO knows only a few set of load-time options which can be used as argument to `\usepackage`. To be more precise it knows only one such option:

upgrade

When this option is used **ACRO** tries to give as much helpful and meaningful warning or error messages when a deprecated or removed command or setup is used. This is especially useful if you are upgrading from version 2.

3.2. Setup command

All options of **ACRO** that have *not* been mentioned in section 3.1 have to be set up either with this command

`\acsetup{<options>}`

or as option to other commands. If this is possible then it is described when the corresponding commands are explained. Options usually follow a key/value syntax like and are always described in the following way:

option

An option without a value. Those options are very rare if there are any.

`option = {<value>}`

Initial: preset

An option where a value can be given. The pre-set value is given to the right.

`option = choiceA|choiceB|choiceC`

Initial: choiceB

An option with a determined set of choices. The underlined value is chosen if the option is given without value.

`option = true|false`

A boolean option.

module/option

An option at a deeper level belonging to the module **module**.

All of the above is probably clear from an example (using real options):

```

1 \acsetup{
2   make-links = true , % boolean
3   index ,           % boolean
4   format = \emph ,   % standard
5   list / local ,     % boolean option of the list module
6   list/display = all % choice option of the list module
7 }

```

Part II.

Comprehensive description of creation and usage of acronyms

4. Declaring acronyms and other abbreviations

All acronyms have to be declared in the preamble with the following command in order to be used in the document. Any usage of an acronym which has not been declared leads to an error message.

`\DeclareAcronym{⟨id⟩}{⟨list of properties⟩}`

The basic command for declaring an acronym where `⟨id⟩` is a unique string identifying the acronym. Per default behaviour this is case sensitive which means `id` is different from `ID`, for example. There is an option `case-sensitive` to change this.

This command understands a number of properties which are listed in the following sections. This is a comprehensive overview over the existing properties. Most of these properties are explained in more detail in later sections of this manual.

! In its simplest form an acronym needs a short and a long form. Please note that both properties *must* be set.

4.1. Basic properties

`short = {⟨text⟩}` (required)

The short form of the acronym. *This property is required:* an acronym must have a short form.

Maybe you mostly have simple acronyms where the `ID` and short form are the same. In that case you can use

`use-id-as-short = true|false` Initial: `false`

to use the `ID` of the acronym as short form. For more complicated cases this would still allow you to set the short form.

4. Declaring acronyms and other abbreviations

long = { $\langle text \rangle$ } (required)
The long form of the acronym. *This property is required:* an acronym must have a description.

alt = { $\langle text \rangle$ } (initially empty)
Alternative short form.

extra = { $\langle text \rangle$ } (initially empty)
Extra information to be added in the list of acronyms.

foreign = { $\langle long\ form\ in\ foreign\ language \rangle$ } (initially empty)
Can be useful when dealing with acronyms in foreign languages, see section 11 on page 16 for details.

post = { $\langle text \rangle$ } (initially empty)
 $\langle text \rangle$ is appended to the acronym in the text but not in the list of acronyms.

single = { $\langle text \rangle$ } if unused then equal to **long**
If provided $\langle text \rangle$ will be used instead of the long form if the acronym is only used a single time and the option **single** has been set, see section 8 on page 12.

sort = { $\langle text \rangle$ } if unused then equal to **short**
If used the acronym will be sorted according to this property instead of its ID.

class = { $\langle csv\ list \rangle$ } (initially empty)
The class(es) the acronym belongs to.

cite = [$\langle prenote \rangle$][$\langle postnote \rangle$]{ $\langle citation\ keys \rangle$ } (initially empty)
A citation that is printed to the acronym according to an option explained later.

index = { $\langle text \rangle$ }
Default This property allows to overwrite the automatic index entry with an arbitrary one. See section 15.2 on page 19 for details.

4.2. Properties related to plural and indefinite forms

short-plural = { $\langle text \rangle$ } Initial: s
The plural ending appended to the short form.

short-plural-form = { $\langle text \rangle$ } (initially empty)
The plural short form of the acronym; replaces the short form when used instead of appending the plural ending.

long-plural = { $\langle text \rangle$ } Initial: s
The plural ending appended to the long form.

long-plural-form = { $\langle text \rangle$ } (initially empty)
Plural long form of the acronym; replaces the long form when used instead of appending the plural ending.

`alt-plural` = $\{\langle text \rangle\}$ Initial: s
The plural ending appended to the alternative form.

`alt-plural-form` = $\{\langle text \rangle\}$ (initially empty)
The plural alternative form of the acronym; replaces the alternative form when used instead of appending the plural ending.

`foreign-plural` = $\{\langle text \rangle\}$ Initial: s
The plural ending appended to the foreign form.

`foreign-plural-form` = $\{\langle text \rangle\}$ (initially empty)
Plural foreign form of the acronym; replaces the foreign form when used instead of appending the plural ending.

`short-indefinite` = $\{\langle text \rangle\}$ Initial: a
Indefinite article for the short form.

`long-indefinite` = $\{\langle text \rangle\}$ Initial: a
Indefinite article for the long form.

`alt-indefinite` = $\{\langle text \rangle\}$ Initial: a
Indefinite article for the alternative form.

4.3. Properties related to formatting

`format` = $\{\langle \textit{\texttt{TeX code}} \rangle\}$ (initially empty)
The format used for both short and long form of the acronym.

`short-format` = $\{\langle \textit{\texttt{TeX code}} \rangle\}$ if unused then equal to `format`
The format used for the short form of the acronym.

`long-format` = $\{\langle \textit{\texttt{TeX code}} \rangle\}$ if unused then equal to `format`
The format used for the long form of the acronym.

`alt-format` = $\{\langle \textit{\texttt{TeX code}} \rangle\}$ if unused then equal to `short-format`
The format used for the alternative form of the acronym. If this is not given the short format will be used.

`extra-format` = $\{\langle \textit{\texttt{TeX code}} \rangle\}$ (initially empty)
The format used for the additional information of the acronym.

`foreign-format` = $\{\langle \textit{\texttt{TeX code}} \rangle\}$ (initially empty)
The format used for the foreign form of the acronym.

`single-format` = $\{\langle \textit{\texttt{TeX code}} \rangle\}$ if unused then equal to `long-format`
The format used for the acronym if the acronym is only used a single time.

`first-style` = long-short|short-long|short|long|footnote (initially empty)
The style of the first appearance of the acronym, see also section 7 on page 11.

`single-style` = long-short | short-long | short | long | footnote (initially empty)
The style of a single appearance of the acronym, see also section 8 on page 12.

4.4. Properties related to the created PDF file

`pdfstring` = { $\langle pdfstring \rangle$ } if unused then equal to `short`
Used as PDF string replacement in bookmarks when used together with the hyperref [OR19] or the bookmark package [Obe19].

`pdfcomment` = { $\langle text \rangle$ }
Sets a tooltip description for an acronym. For actually getting tooltips you also need an appropriate setting of the options `pdfcomment/cmd` and `pdfcomment/use`, see also section 18.2 on page 19.

`short-acc` = { $\langle text \rangle$ } if unused then equal to `short`
Sets the ActualText property as presented by the accsupp package for the short form of the acronym.

`long-acc` = { $\langle text \rangle$ } if unused then equal to `long`
Sets the ActualText property as presented by the accsupp package for the long form of the acronym.

`alt-acc` = { $\langle text \rangle$ } if unused then equal to `alt`
Sets the ActualText property as presented by the accsupp package for the alternative short form of the acronym.

`foreign-acc` = { $\langle text \rangle$ } if unused then equal to `foreign`
Sets the ActualText property as presented by the accsupp package for the foreign form of the acronym.

`extra-acc` = { $\langle text \rangle$ } if unused then equal to `extra`
Sets the ActualText property as presented by the accsupp package for the extra information of the acronym.

`single-acc` = { $\langle text \rangle$ } if unused then equal to `long-acc`
Sets the ActualText property as presented by the accsupp package for a single appearance of the acronym.

4.5. Further properties

`list` = { $\langle text \rangle$ } if unused then equal to `long`
If specified this will be written in the list as description instead of the long form if the corresponding list template supports it.

`foreign-babel` = { $\langle language \rangle$ } (initially empty)
The babel [Bra19] or polyglossia [Cha19] language of the foreign form. This language is used to wrap the entry with `\foreignlanguage{ $\langle language \rangle$ }` if either babel or polyglossia is loaded. You'll need to take care that the corresponding language is loaded by babel or polyglossia.

5. Using acronyms

`foreign-locale` = { $\langle language \rangle$ } (initially empty)

The language name that is output when the option `locale/display` is used. If this property is not set then the appropriate value might be derived from `foreign-babel`. See section 11 on page 16 for details.

`index-sort` = { $\langle text \rangle$ } if unused then equal to `sort`

If you use the 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` property. This property allows to set an individual sorting option for the index. See section 15.2 on page 19 for details

5. Using acronyms

There are a number of commands to use acronyms with. Their names always follow the same pattern which should make their usage intuitive immediately.

All of these commands have a starred form which means “don’t count this as usage”. All of these commands also have an optional argument that allows to set options for that usage only.

`\acrocommand*[\langle options \rangle]{\langle id \rangle}`

This is the general syntax of all of the commands listed below. The star and the optional argument is left away for the sake of readability.

`\ac{\langle id \rangle}` `\Ac{\langle id \rangle}` `\acp{\langle id \rangle}` `\Acp{\langle id \rangle}` `\iac{\langle id \rangle}` `\Iac{\langle id \rangle}`

`\ac` prints the acronym $\langle id \rangle$, the first time with full description and every subsequent use only the abbreviated form. `\Ac` does the same but uppercases the first letter – this may be needed at the beginning of a sentence. The commands `\acp` and `\Acp`, resp., print the corresponding plural forms. The commands `\iac` and `\Iac`, resp., print indefinite forms.

`\acs{\langle id \rangle}` `\Acs{\langle id \rangle}` `\acsp{\langle id \rangle}` `\Acsp{\langle id \rangle}` `\iacs{\langle id \rangle}` `\Iacs{\langle id \rangle}`

`\acs` prints the short form of the acronym $\langle id \rangle$. `\Acs` does the same but uppercases the first letter. The commands `\acsp` and `\Acsp`, resp., print the corresponding plural forms. The commands `\iacs` and `\Iacs`, resp., print indefinite forms.

`\acl{\langle id \rangle}` `\Acl{\langle id \rangle}` `\aclp{\langle id \rangle}` `\Aclp{\langle id \rangle}` `\iacl{\langle id \rangle}` `\Iacl{\langle id \rangle}`

`\acl` prints the long form of the acronym $\langle id \rangle$. `\Acl` does the same but uppercases the first letter. The commands `\aclp` and `\Aclp`, resp., print the corresponding plural forms. The commands `\iacl` and `\Iacl`, resp., print indefinite forms.

`\aca{\langle id \rangle}` `\Aca{\langle id \rangle}` `\acap{\langle id \rangle}` `\Acap{\langle id \rangle}` `\iaca{\langle id \rangle}` `\Iaca{\langle id \rangle}`

`\aca` prints the alternative short form of the acronym $\langle id \rangle$. `\Aca` does the same but uppercases the first letter. The commands `\acap` and `\Acap`, resp., print the corresponding plural forms. The commands `\iaca` and `\Iaca`, resp., print indefinite forms.

`\acf{\langle id \rangle}` `\Acf{\langle id \rangle}` `\acfp{\langle id \rangle}` `\Acfp{\langle id \rangle}` `\iacf{\langle id \rangle}` `\Iacf{\langle id \rangle}`

`\acf` prints the full form of the acronym $\langle id \rangle$. `\Acf` does the same but uppercases the first letter. The commands `\acfp` and `\Acfp`, resp., print the corresponding plural forms. The commands `\iacf` and `\Iacf`, resp., print indefinite forms.

6. Alternative short forms

The usage should be clear. Let's assume you have defined an acronym UFO like this:

```
1 \DeclareAcronym{ufo}{  
2   short = UFO ,  
3   long = unidentified flying object ,  
4   foreign = unbekanntes Flugobjekt ,  
5   foreign-plural-form = unbekannte Flugobjekte ,  
6   foreign-babel = ngerman ,  
7   long-indefinite = an  
8 }
```

Then typical outputs look like this:

```
1 \ac{ufo} \\  
2 \iac{ufo} \\  
3 \iacl{ufo} \\  
4 \Iacf{ufo} \\  
5 \acfp{ufo}
```

unidentified flying object (unbekanntes Flugobjekt, UFO)
a UFO
an unidentified flying object
an Unidentified flying object (unbekanntes Flugobjekt, UFO)
unidentified flying objects (unbekannte Flugobjekte, UFOs)

6. Alternative short forms

Sometimes expressions have two different short forms. An example might be JPEG which also often is JPG. This is what the property `alt` is there for.

`alt = {⟨text⟩}`

Alternative short form.

Let's define JPEG:

```
1 \DeclareAcronym{jpg}{  
2   short = JPEG ,  
3   sort  = jpeg ,  
4   alt   = JPG ,  
5   long  = Joint Photographic Experts Group  
6 }
```

And let's see how to use it:

7. The first or full appearance

```
1 \ac{jpg} \\  
2 \ac{jpg} \\  
3 \aca{jpg}
```

Joint Photographic Experts Group (JPEG or JPG)
JPEG
JPG

As you can see the full form shows both short forms of the acronym. This could be changed by altering the template for the full form, see section 19 on page 19 and section 7. The alternative form is also printed in the list of acronyms, see section A on page 21. This can also be changed by altering the template for the list, again see section 19.

7. The first or full appearance

If an acronym is used for the first time with `\ac` (after any number of usages with the starred forms of the usage commands listed in section 5 on page 9) or if an acronym is used `\acf`, then the first or full appearance of the acronym is printed.¹

The first or full appearance of an acronym is determined by this option:

`first-style` = long-short|short-long|short|long|footnote Initial: long-short

The style of the first appearance of the acronym. This options sets the appearance for all acronyms. Available options in reality are the names of all defined templates of the type acronym. All pre-defined templates can be found in section 19.1 on page 19.

It might be desirable to set the first appearance of an acronym individually. This is possible by setting the corresponding property:

`first-style` = long-short|short-long|short|long|footnote (initially empty)

The style of the first appearance of the acronym.

Let's again look at an example:

```
1 \acf[first-style=long-short]{cd} \\  
2 \acf[first-style=short-long]{cd} \\  
3 \acf[first-style=footnote]{cd} \\  
4 \acf[first-style=long]{cd} \\  
5 \acf[first-style=short]{cd}
```

compact disc (CD)
CD (compact disc)
CD^a
compact disc
CD

a. compact disc

This also demonstrates the use of the optional argument.

An example of an abbreviation that should have long as first appearance might be “etc.”, defined like this

1. This usually requires at least two compilations.

8. Single appearances of an acronym

```
1 \DeclareAcronym{etc}{  
2   short = etc\acdot ,  
3   long = et cetera ,  
4   format = \textit ,  
5   first-style = long  
6 }
```

and output like this:

```
1 \ac{etc}, \ac{etc} \ac{etc}.           et cetera, etc. etc.
```

The command `\acdot` is explained in section 14 on page 19. Basically it checks if a dot follows and outputs a dot if not.

8. Single appearances of an acronym

If an acronym is used only once (not counting usages with the starred forms of the usage commands listed in section 5 on page 9), then the single appearance of the acronym is printed.²

The single appearance of an acronym is determined by this option:

`single = true|false|⟨number⟩` Initial: false

This option determines whether a single appearance of an acronym counts as *usage*. It might be desirable in such cases that an acronym is simply printed as long form and not added to the list of acronym. This is what this option does. With `⟨number⟩` the minimal number of usages can be given that needs to be exceeded. `single = {1}` is the same as `single = {true}`.

`single-style = long-short|short-long|short|long|footnote` Initial: long

The style of the single appearance of an acronym. Can be used to determine how a single appearance is printed if the option `single` has been set. This option sets the appearance for all acronyms. Available options in reality are the names of all defined templates of the type acronym. All pre-defined templates can be found in section 19.1 on page 19.

If you like you can also set the single appearance of an acronym individually:

`single = {⟨text⟩}` if unused then equal to `long`

If provided `⟨text⟩` will be used instead of the long form if the acronym is only used a single time and the option `single` has been set.

`single-format = {⟨TEX code⟩}` if unused then equal to `long-format`

The format used for the acronym if the acronym is only used a single time.

`single-style = long-short|short-long|short|long|footnote` (initially empty)

The style of the single appearance of the acronym.

2. This usually requires at least two compilations.

9. Plural forms and other endings

Let's again look at an example. The acronym PNG is defined as follows:

```
1 \DeclareAcronym{png}{  
2   short = PNG ,  
3   long  = Portable Network Graphics ,  
4   first-style = short-long ,  
5   single-style = short  
6 }
```

And it is used only once in this manual³:

```
1 \ac{png}                                PNG
```

Please be aware that `\acf` would still print the full form, of course.

9. Plural forms and other endings

9.1. The plural ending and the plural form

Not in all languages plural forms are as easy as always appending an “s”. Not even English. Sometimes there's other endings instead.⁴ This is why `ACRO` has quite a number of different properties related to plural forms or endings:

`short-plural` = $\{\langle text \rangle\}$ Initial: s

The plural ending appended to the short form.

`short-plural-form` = $\{\langle text \rangle\}$ (initially empty)

The plural short form of the acronym; replaces the short form when used instead of appending the plural ending.

`long-plural` = $\{\langle text \rangle\}$ Initial: s

The plural ending appended to the long form.

`long-plural-form` = $\{\langle text \rangle\}$ (initially empty)

Plural long form of the acronym; replaces the long form when used instead of appending the plural ending.

`alt-plural` = $\{\langle text \rangle\}$ Initial: s

The plural ending appended to the alternative form.

`alt-plural-form` = $\{\langle text \rangle\}$ (initially empty)

The plural alternative form of the acronym; replaces the alternative form when used instead of appending the plural ending.

3. You will find it in the list of acronyms in section A nonetheless as this document does `list/display = {all}`.

4. German is full of such examples.

9. Plural forms and other endings

`foreign-plural = {⟨text⟩}` Initial: s
The plural ending appended to the foreign form.

`foreign-plural-form = {⟨text⟩}` (initially empty)
Plural foreign form of the acronym; replaces the foreign form when used instead of appending the plural ending.

There are two options which allow to change the default values for the whole document:

`short-plural-ending = {⟨text⟩}` Initial: s
Defines the plural ending for the short forms to be ⟨text⟩.

`long-plural-ending = {⟨text⟩}` Initial: s
Defines the plural ending for the long forms to be ⟨text⟩.

Now let's see two simple examples demonstrating the two different kinds of plural settings:

```
1 \DeclareAcronym{sw}{  
2   short = SW ,  
3   long = Sammelwerk ,  
4   long-plural = e  
5 }  
6 \DeclareAcronym{MP}{  
7   short = MP ,  
8   long = Member of Parliament ,  
9   long-plural-form = Members of Parliament  
10 }
```

The first one has another plural ending than the usual “s”. The second one has a different plural form altogether because appending an “s” would give a wrong form:

1 \acfp{sw} \par	Sammelwerke (SWs)
2 \acfp{MP}	Members of Parliament (MPs)

9.2. Other endings

There are other such concepts which is why **ACRO** generalizes the concept of endings.

`\DeclareAcroEnding{⟨name⟩}{⟨short default⟩}{⟨long default⟩}`

This command can be used to define properties and options analogous to the plural endings which have been defined this way:

```
1 \DeclareAcroEnding{plural}{s}{s}
```

In general `\DeclareAcroEnding{⟨foo⟩}{⟨x⟩}{⟨y⟩}` defines these options

<code>short-⟨foo⟩-ending = {⟨value⟩}</code>	Initial: ⟨x⟩
<code>long-⟨foo⟩-ending = {⟨value⟩}</code>	Initial: ⟨y⟩
and these properties	
<code>short-⟨foo⟩ = {⟨value⟩}</code>	Initial: ⟨x⟩
<code>short-⟨foo⟩-form = {⟨value⟩}</code>	(initially empty)
<code>alt-⟨foo⟩ = {⟨value⟩}</code>	Initial: ⟨x⟩
<code>alt-⟨foo⟩-form = {⟨value⟩}</code>	(initially empty)
<code>long-⟨foo⟩ = {⟨value⟩}</code>	Initial: ⟨y⟩
<code>long-⟨foo⟩-form = {⟨value⟩}</code>	(initially empty)
<code>foreign-⟨foo⟩ = {⟨value⟩}</code>	Initial: ⟨y⟩
<code>foreign-⟨foo⟩-form = {⟨value⟩}</code>	(initially empty)
<code>single-⟨foo⟩ = {⟨value⟩}</code>	Initial: ⟨y⟩
<code>single-⟨foo⟩-form = {⟨value⟩}</code>	(initially empty)
<code>extra-⟨foo⟩ = {⟨value⟩}</code>	Initial: ⟨y⟩
<code>extra-⟨foo⟩-form = {⟨value⟩}</code>	(initially empty)

In addition another command is defined which is meant to be used in template definitions.

`\acro⟨foo⟩`

This command tells the template that the ending ⟨foo⟩ should be used.

Section 20 on page 19 has an example of how this can be used to define a possessive ending and commands that make use of them like this:

```
\acfg{MP}
```

Member's of Parliament (MP's)

10. Indefinite forms

Indefinite forms can be a problem if the short and the long form of acronyms have different indefinite articles.⁵

5. This may very well be a language specific issue.

<pre>1 \acreset{ufo}% 2 a \ac{ufo} \par 3 an \ac{ufo}</pre>	a unidentified flying object (unbekanntes Flugobjekt, UFO) an UFO
---	---

And what good would it be to use a package like **ACRO** if you have to keep track of of and second uses, anyway? This is why UFO should be defined like we did on page 10. We then can just use the dedicated commands and let them decide for us:

<pre>1 \acreset{ufo}% 2 \iac{ufo} \par 3 \iac{ufo}</pre>	an unidentified flying object (unbekanntes Flugobjekt, UFO) a UFO
--	---

The commands which also output the indefinite article all start with an “i” and have all been described in section 5 on page 9 already: `\iac`, `\Iac`, `\iacs`, `\Iacs`, `\iacl`, `\Iacl`, `\iaca`, `\Iaca`, `\iacf`, and `\Iacf`.

11. Foreign language acronyms

Sometimes and in some fields more often than in others abbreviations are used that are derived from another language. **ACRO** provides a number of properties for such cases:

foreign = { \langle long form in foreign language \rangle } (initially empty)

Can be useful when dealing with acronyms in foreign languages, see section 11 for details.

foreign-plural = { \langle text \rangle } Initial: s

The plural ending appended to the foreign form.

foreign-plural-form = { \langle text \rangle } (initially empty)

Plural foreign form of the acronym; replaces the foreign form when used instead of appending the plural ending.

foreign-format = { \langle TeX code \rangle } (initially empty)

The format used for the foreign form of the acronym.

foreign-babel = { \langle language \rangle } (initially empty)

The babel or polyglossia language of the foreign form. This language is used to wrap the entry with `\foreignlanguage{ \langle language \rangle }` if either babel or polyglossia is loaded. You’ll need to take care that the corresponding language is loaded by babel or polyglossia.

foreign-locale = { \langle language \rangle } (initially empty)

The language name that is output when the option `locale/display` is used. If this property is not set then the appropriate value might be derived from **foreign-babel**.

There are also some options:

11. Foreign language acronyms

`display = true|false` `locale/` Initial: false

This options determines whether the language of the foreign form is printed or not when the full form of the acronym is printed.

`list/locale/display = true|false` Initial: false

The same but for the list of acronyms.

`locale/` `format = {<code>}` Initial: `\em\text_titlecase_first:n`

Determines how said language is formatted when printed. The last command in `<code>` may take a mandatory argument.

Let's say you are writing a German document and are using the abbreviation ECU for Steuergerät which stems from the English "Electronic Control Unit". Then you can define it as follows:

```
1 \DeclareAcronym{ecu}{
2   short   = ECU ,
3   long    = Steuergerät ,
4   foreign = Electronic Control Unit ,
5   foreign-babel = english ,
6   foreign-locale = englisch
7 }
```

Now the abbreviation is introduced so that everyone understands the confusion:

```
1 \ac{ecu} \par
2 \acsetup{locale/display,locale/format=\emph}
3 \acf{ecu}
```

Steuergerät (Electronic Control Unit, ECU)
Steuergerät (*englisch*: Electronic Control Unit, ECU)

The property `foreign-babel` is used for ensuring correct hyphenation as long as you use `babel` or `polyglossia` and load the corresponding language, too. If you are writing your document in English then `ACRO` is able to deduce the language used for the "locale" field by itself:

```
1 \DeclareAcronym{eg}{
2   short = e.g.\acdot ,
3   long  = for example ,
4   foreign = exempli gratia ,
5   foreign-babel = latin ,
6   short-format = \textit ,
7   foreign-format = \textit
8 }
```

11. Foreign language acronyms

```
1 \acsetup{locale/display,first-style=short-long}  
2 \acf{eg}
```

e.g. (Latin: *exempli gratia*: for example)

12. Uppercasing

13. Printing the list

13.1. The main command and its options

13.2. Several lists using classes

13.3. Local lists

14. Trailing tokens

15. Citing and indexing

15.1. Citing

15.2. Indexing

16. Using or resetting acronyms

17. Localisation

18. Bookmarks and accessibility support

18.1. hyperref support

18.2. PDF comments

18.3. Accessibility support

Part III.

Extending ACRO

19. Templates

19.1. Pre-defined templates

19.2. Defining new templates

19.3. New acronym templates

19.4. New list templates

19.5. New heading templates

20. Own acronym commands

20. Own acronym commands

```
1 \DeclareAcroEnding{possessive}{'s}{'s}
2
3 \NewAcroCommand\acg{m}{\acropossessive\AcroUseTemplate{first}{#1}}
4 \NewAcroCommand\acsg{m}{\acropossessive\AcroUseTemplate{short}{#1}}
5 \NewAcroCommand\aclg{m}{\acropossessive\AcroUseTemplate{long}{#1}}
6 \NewAcroCommand\acfg{m}{%
7   \acrofull
8   \acropossessive
9   \AcroUseTemplate{first}{#1}%
10 }
11 \NewAcroCommand\iacsg{m}{%
12   \acroindefinite
13   \acropossessive
14   \AcroUseTemplate{short}{#1}%
15 }
```

Part IV.

Appendix

A. Examples

B. Acronyms

Below all abbreviations are listed which have been defined for the manual.

CD compact disc	3, 11
CTAN Comprehensive T _E X Archive Network	2
e.g. for example (Latin: <i>exempli gratia</i>)	18
ECU Steuergerät (<i>Englisch</i> : Electronic Control Unit)	17
etc. <i>et cetera</i>	12
ID identification string	5f., 9
JPEG/JPG Joint Photographic Experts Group	11
LA Los Angeles	
LPPL L ^A T _E X Project Public License	
MP Member of Parliament	14f.
NATO Organisation des Nordatlantikvertrags (<i>Englisch</i> : North Atlantic Treaty Organization)	
NY New York	
PDF Portable Document Format	8
PNG Portable Network Graphics	13
T_EX.sx T _E X StackExchange	
UFO unidentified flying object (<i>German</i> : unbekanntes Flugobjekt)	10, 15f.

C. References

- [Bra19] Johannes BRAAMS. *babel*. Version 3.33. July 19, 2019.
URL: <https://www.ctan.org/pkg/babel/>.
- [Cha19] François CHARETTE. *polyglossia*. Version 1.44. Apr. 4, 2019.
URL: <https://www.ctan.org/pkg/polyglossia/>.
- [Obe19] Heiko OBERDIEK. *bookmark*. Version 1.28. Dec. 3, 2019.
URL: <https://www.ctan.org/pkg/bookmark/>.
- [OR19] Heiko OBERDIEK and Sebastian RAHTZ. *hyperref*. Version 6.88i. Sept. 12, 2019.
URL: <https://www.ctan.org/pkg/hyperref/>.

D. Index

A		
<code>\Ac</code>	3, 9	
<code>\ac</code>	3 f., 9–13, 16 f.	
<code>\Aca</code>	9	
<code>\aca</code>	9, 11	
<code>\Acap</code>	9	
<code>\acap</code>	9	
<code>accsupp</code> (package)	8	
<code>\acdot</code>	12, 17	
<code>\Acf</code>	3, 9	
<code>\acf</code>	3 f., 9, 11, 13, 17 f.	
<code>\acfg</code>	15, 20	
<code>\Acfp</code>	9	
<code>\acfp</code>	9 f., 14	
<code>\acg</code>	20	
<code>\Acl</code>	3, 9	
<code>\acl</code>	3 f., 9	
<code>\aclg</code>	20	
<code>\Aclp</code>	9	
<code>\aclp</code>	9	
<code>\Acp</code>	9	
<code>\acp</code>	9	
<code>\acreset</code>	16	
<code>\acrofull</code>	20	
<code>\acroindefinite</code>	20	
<code>\acropossessive</code>	20	
<code>\AcroUseTemplate</code>	20	
<code>\Acs</code>	3, 9	
<code>\acs</code>	3 f., 9	
<code>\acsetup</code>	4 f., 17 f.	
<code>\acsg</code>	20	
<code>\Acsp</code>	9	
<code>\acsp</code>	9	
<code>alt</code> (property)	6, 8, 10	
<code>alt-acc</code> (property)	8	
<code>alt-format</code> (property)	7	
<code>alt-indefinite</code> (property)	7	
<code>alt-plural</code> (property)	7, 13	
<code>alt-plural-form</code> (property) ..	7, 13	
B		
<code>babel</code>	8	
<code>babel</code> (package)	8, 16 f.	
BEZOS, Javier	8	
<code>bookmark</code>	8	
<code>bookmark</code> (package)	8	
BRAAMS, Johannes	8	
C		
<code>case-sensitive</code>	5	
CD	4, 11	
CHARETTE, François	8	
<code>cite</code> (property)	6	
<code>class</code> (property)	6	
CTAN	2	
D		
<code>\DeclareAcroEnding</code>	14, 20	
<code>\DeclareAcronym</code> ..	3, 5, 10, 12 ff., 17	
<code>display</code>	9, 16 f.	
E		
<code>e.g.</code>	18	
ECU	17	
<code>etc.</code>	12	
<code>extra</code> (property)	6, 8	
<code>extra-acc</code> (property)	8	
<code>extra-format</code> (property)	7	
F		
<code>first-style</code>	11	
G		
H		
<code>hyperref</code>	8	
<code>hyperref</code> (package)	8	
I		
<code>\Iac</code>	9, 16	
<code>\iac</code>	9 f., 16	
<code>\Iaca</code>	9, 16	
<code>\iaca</code>	9, 16	
<code>\Iacf</code>	9 f., 16	
<code>\iacf</code>	9, 16	
<code>\Iacl</code>	9, 16	
<code>\iacl</code>	9 f., 16	
<code>\Iacs</code>	9, 16	
<code>\iacs</code>	9, 16	
<code>\iacsg</code>	20	
ID	5 f., 9	
<code>index</code>	9	
<code>index</code> (property)	6	
<code>index-sort</code> (property)	9	
J		
JPEG	11	

Index

L		
<code>list</code> (property)	8	
<code>list/display</code>	13	
<code>locale</code>	9, 16	
<code>long</code> (property)	6, 8, 12	
<code>long-acc</code> (property)	8	
<code>long-format</code> (property)	7, 12	
<code>long-indefinite</code> (property)	7	
<code>long-plural</code> (property)	6, 13	
<code>long-plural-ending</code>	14	
<code>long-plural-form</code> (property)	6, 13	
<code>LPPL</code>	3	
M		
<code>MP</code>	14 f.	
N		
<code>\NewAcroCommand</code>	20	
O		
OBERDIEK, Heiko	8	
P		
<code>PDF</code>	8	
<code>pdfcomment</code> (property)	8	
<code>pdfcomment/cmd</code>	8	
<code>pdfcomment/use</code>	8	
<code>pdfstring</code> (property)	8	
<code>PNG</code>	13	
<code>polyglossia</code>	8	
<code>polyglossia</code> (package)	8, 16 f.	
<code>post</code> (property)	6	
R		
RAHTZ, Sebastian	8	
REUTENAUER, Arthur	8	
S		
<code>short</code> (property)	5 f., 8	
<code>short-acc</code> (property)	8	
<code>short-format</code> (property)	7	
<code>short-indefinite</code> (property)	7	
<code>short-plural</code> (property)	6, 13	
<code>short-plural-ending</code>	14	
<code>short-plural-form</code> (property)	6, 13	
<code>single</code>	6, 12	
<code>single</code> (property)	6, 12	
<code>single-acc</code> (property)	8	
<code>single-format</code> (property)	7, 12	
<code>single-style</code>	8, 12	
<code>single-style</code> (property)	8, 12	
<code>sort</code> (property)	6, 9	
<code>SW</code>	14	
U		
<code>UFO</code>	10, 16	
<code>upgrade</code>	4	
<code>use-id-as-short</code>	5	